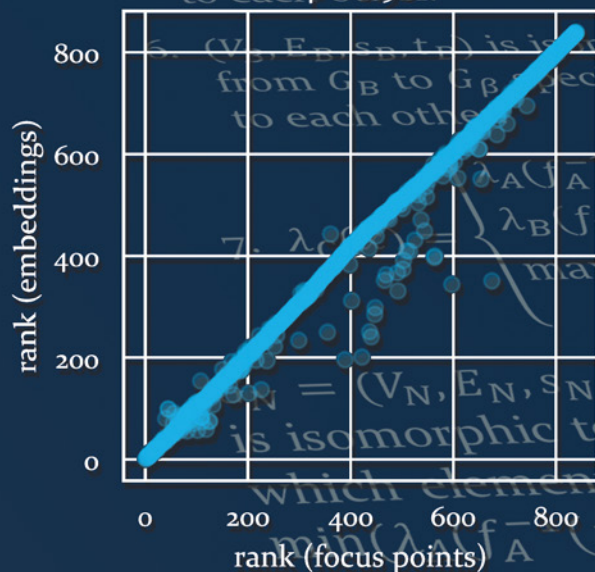


$G_C = (V_C, E_C, s_C, t_C, \langle \Sigma, \leq \rangle, \lambda_C)$ is a linguistic structure in which linguistic structure $G_A = (V_A, E_A, s_A, t_A, \langle \Sigma, \leq \rangle, \lambda_A)$ cooccurs with linguistic structure $G_B = (V_B, E_B, s_B, t_B, \langle \Sigma, \leq \rangle, \lambda_B)$ such that the following constraints hold:

1. $G_\alpha = (V_\alpha, E_\alpha, s_\alpha, t_\alpha)$ is a connected subgraph of (V_C, E_C, s_C, t_C) .
2. $G_\beta = (V_\beta, E_\beta, s_\beta, t_\beta)$ is a connected subgraph of (V_C, E_C, s_C, t_C) .
3. $V_\alpha \cup V_\beta = V_C$ and $E_\alpha \cup E_\beta = E_C$.
4. $G_\ell = (V_\ell, E_\ell, s_\ell, t_\ell)$ is a connected subgraph of G_C where $V_\ell = V_\alpha$, $E_\ell = E_\alpha \cap E_\beta$ and s_ℓ and t_ℓ are the restrictions of s_C and t_C to E_ℓ .
5. (V_A, E_A, s_A, t_A) is isomorphic to G_α and the isomorphism f_A from G_A to G_α specifies which elements of the two graphs to each other.

$\rho = 0.98$



FAU Forschungen, Reihe A, Geisteswissenschaften 12

Thomas Proisl

The cooccurrence of linguistic structures

The cooccurrence of linguistic structures

FAU Forschungen, Reihe A

Geisteswissenschaften

Band 12

Herausgeber der Reihe:

Wissenschaftlicher Beirat der FAU University Press

Thomas Proisl

The cooccurrence of linguistic structures

Erlangen
FAU University Press
2019

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind
im Internet über <http://dnb.d-nb.de> abrufbar.

Bitte zitieren als

Proisl, Thomas. 2019. *The cooccurrence of linguistic structures*.
FAU Forschungen, Reihe A Geisteswissenschaften Band 12.
Erlangen: FAU University Press. DOI: 10.25593/978-3-96147-201-7.

Das Werk, einschließlich seiner Teile, ist urheberrechtlich geschützt.
Die Rechte an allen Inhalten liegen bei ihren jeweiligen Autoren.
Sie sind nutzbar unter der Creative Commons Lizenz BY.

Der vollständige Inhalt des Buchs ist als PDF über den OPUS Server
der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:
<https://opus4.kobv.de/opus4-fau/home>

Verlag und Auslieferung:
FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck: docupoint GmbH

ISBN: 978-3-96147-200-0 (Druckausgabe)
eISBN: 978-3-96147-201-7 (Online-Ausgabe)
ISSN: 2199-014X
DOI: 10.25593/978-3-96147-201-7

The cooccurrence of linguistic structures

der Philosophischen Fakultät und Fachbereich Theologie
der Friedrich-Alexander-Universität
Erlangen-Nürnberg
zur
Erlangung des Doktorgrades Dr. phil.

vorgelegt von
Thomas Proisl
aus Heilbronn

Als Dissertation genehmigt
von der Philosophischen Fakultät und Fachbereich Theologie
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: 2018-12-14

Vorsitzender des Promotionsorgans: Prof. Dr. Thomas Demmelhuber

Gutachter: Prof. Dr. Stefan Evert
Prof. Dr. Thomas Herbst
Prof. Dr. Ulrich Heid

To Susi, Sophie and Hugo.

Abstract

The study of cooccurrences has had a profound impact on our view of language. It contributed greatly to the insight that semi-preconstructed phrases and item-specific knowledge are central to how language works. Cooccurrences of various items have been studied under a variety of names, e. g. collocation, colligation or collostruction, and are important for identifying multiword units, extracting valency patterns and generally investigating patterns of language use. While there are well-understood and fully worked out statistical models for the analysis of cooccurrences of pairs of words, no such model exists for cooccurrences of larger linguistic structures. This situation is remedied by the current work.

In Chapter 2, we give an overview of approaches that analyze cooccurrences of linguistic items ranging from words to syntactic structures. This includes positional and relational cooccurrences of words, different variants of collostructional analysis, hybrid n-grams involving both lexical and grammatical items, certain approaches to multiword collocations and certain approaches to the extraction of valency patterns. Where appropriate, we discuss the limitations of the respective approaches and show that none of them allows for the analysis of cooccurrences of arbitrary linguistic structures.

The predominant approach to the statistical analysis of cooccurrences – that is also the basis for the model presented in the current work – is based on the analysis of 2×2 contingency tables that represent a cross-classification of the units of analysis. In Chapter 3, we describe that approach and discuss the criticism that has been leveled against it.

In Chapter 4, we develop the generalized cooccurrence model, an explicit formal model for the statistical analysis of cooccurrences of arbitrary linguistic structures. To this end, we first introduce the necessary prerequisites for formally representing linguistic structures. We opt for a representation of linguistic structures as labeled connected directed graphs where the labels are organized as a partially ordered set. That choice allows for a high degree of flexibility in the way linguistic information can be modeled. Based on the concept of subgraph isomorphism and on the partial order of the labels, we formally define the subsumption relation on linguistic structures. We call

the resulting mapping from a more general linguistic structure to a more specific one that is subsumed by it an embedding.

We demonstrate that determining the frequency of a linguistic structure in a corpus is a non-trivial task and formally define four different counting methods. Additionally, we provide a generalized perspective on the counting methods, showing that all four can be formally represented as partitions of the set of embeddings. When cooccurrences of linguistic structures are counted and cross-classified, two further problems can arise: The problem of inflated frequencies, where, due to the effects of combination and permutation, the same tokens are counted multiple times, and the inconsistency problem, where a unit of analysis cannot be consistently classified into one of the four cells of a 2×2 contingency table. The different counting methods are affected by those problems to different degrees.

The main contribution of the current work is the generalized cooccurrence model that is developed and formalized in Section 4.5 and that allows for the analysis of cooccurrences of two arbitrary linguistic structures. The general idea is that if two linguistic structures cooccur, then they overlap in at least one vertex, i. e. the cooccurrence of the two is itself a connected linguistic structure. The instances of that overlapping part serve as the units of analysis and are cross-classified in a contingency table. The formalization takes the generalized counting methods into account and suggests workarounds for coping with the inconsistency problem.

We show that existing methods for the analysis of two-word cooccurrences and for collostructional analysis are merely special cases of the generalized cooccurrence model. We also show that some seeming limitations of the generalized cooccurrence model can be overcome by introducing additional vertices and edges to the graph structure. This increases the range of possible analyses and allows, for example, to explicitly search for monotransitive uses of a verb, i. e. instances without an outgoing indirect object relation.

The evaluation in Chapter 5 has a double focus. In the theoretical part, we analyze how popular association measures react to inflated frequencies and find that they fall into three classes with quite different behaviors. In the empirical part, we conduct experiments involving a large number of queries on authentic language data and find that inflated frequencies are not a major problem when working with authentic language data and that the overall impact of the different counting methods is limited.

In Chapter 6, we present a series of case studies that range from reconstructions of traditional collocational and collostructional analyses to cooc-

currences of larger structures, and that highlight the possibilities and the flexibility of the generalized cooccurrence model.

Based on a highly flexible graph-based representation of linguistic structures, the generalized cooccurrence model is a complete formal model for analyzing cooccurrences of arbitrary linguistic structures. Specifically, it formally defines the cooccurrence of two linguistic structures, several methods for counting linguistic structures and the cross-classification of the units of analysis for populating a 2×2 contingency table to which a wide range of association measures can be applied.

Zusammenfassung

Die Kookkurrenz sprachlicher Strukturen

Die Kookkurrenzforschung hat unser Bild von Sprache maßgeblich beeinflusst. Sie hat wesentlich zur Erkenntnis beigetragen, dass „Halbfertigprodukte der Sprache“ (Hausmann, 1984: 398) und einzelwortspezifisches Wissen zentrale Elemente der Funktionsweise von Sprache sind. Die Kookkurrenz von sprachlichen Einheiten wurde mit verschiedenen Ansätzen und unter verschiedenen Bezeichnungen erforscht, etwa als Kollokation, Kolligation oder Kollostruktion. Kookkurrenzen spielen eine wichtige Rolle bei der Identifikation von Mehrworteinheiten, der Extraktion von Valenzmustern und allgemein bei der Untersuchung von sprachlichen Verwendungsmustern. Während es für die Analyse von Zweiwortkookkurrenzen wohlverstandene und vollständig ausgearbeitete statistische Modelle gibt, fehlen solche Modelle für Kookkurrenzen von größeren sprachlichen Strukturen. Diese Lücke wird durch die vorliegende Arbeit geschlossen.

In Kapitel 2 wird ein Überblick über verschiedene Ansätze zur Untersuchung von Kookkurrenzen sprachlicher Strukturen, von Wörtern bis hin zu syntaktischen Strukturen, gegeben. Dazu gehören etwa positionelle und relationale Kookkurrenzen von Wörtern, verschiedene Varianten von kollostruktioneller Analyse, hybride N-Gramme, die lexikalisches und grammatisches Material kombinieren, bestimmte Ansätze zur Extraktion von Mehrwortkollokationen und bestimmte Ansätze zur Extraktion von Valenzmustern. Die Schwachstellen und Beschränkungen der jeweiligen Ansätze werden diskutiert und es wird gezeigt, dass keiner der Ansätze die Analyse von Kookkurrenzen beliebiger sprachlicher Strukturen erlaubt.

Der vorherrschende Ansatz zur statistischen Analyse von Kookkurrenzen, auf dem auch diese Arbeit aufbaut, basiert auf der Analyse von 2×2 -Kontingenztafeln, die eine Kreuzklassifikation der Analyseeinheiten darstellen. In Kapitel 3 wird dieser Ansatz beschrieben und die Kritik, die an ihm geübt wurde, diskutiert.

In Kapitel 4 wird das verallgemeinerte Kookkurrenzmodell entwickelt, ein explizites formales Modell für die statistische Analyse von Kookkurrenzen beliebiger sprachlicher Strukturen. Zu diesem Zweck werden zunächst

die notwendigen Voraussetzungen für eine formale Repräsentation sprachlicher Strukturen eingeführt. Sprachliche Strukturen werden als benannte zusammenhängende gerichtete Graphen repräsentiert, deren Namen in einer halbgeordneten Menge organisiert sind. Dieses Repräsentationsformat ermöglicht eine sehr flexible Modellierung der linguistischen Informationen. Basierend auf dem Begriff der Teilgraph-Isomorphie und der Halbordnung der Namen wird die Subsumptionsrelation auf sprachlichen Strukturen formal definiert. Die Abbildung von einer allgemeineren auf eine speziellere, von dieser subsumierten sprachlichen Struktur wird hier als Einbettung bezeichnet.

Es wird gezeigt, dass die Ermittlung der Frequenz einer sprachlichen Struktur in einem Korpus eine nicht-triviale Aufgabe ist. Vier verschiedene Zählweisen werden formal definiert; außerdem wird eine verallgemeinerte Sichtweise auf die Zählweisen entwickelt, die alle vier auf Partitionierungen der Menge der Einbettungen zurückführt. Beim Zählen und bei der Kreuzklassifikation von Kookkurrenzen sprachlicher Strukturen können zwei weitere Probleme auftauchen: Zum einen kann es durch Kombinations- und Permutationseffekte vorkommen, dass manche Wörter mehrfach gezählt werden und es zu aufgeblähten Frequenzen kommt, zum anderen kann es geschehen, dass eine Analyseeinheit nicht konsistent einem der vier Felder der 2×2 -Kontingenztafel zugewiesen werden kann und es zu Inkonsistenzen kommt. Die verschiedenen Zählweisen sind von diesen Problemen unterschiedlich stark betroffen.

Der Hauptbeitrag dieser Arbeit ist die Entwicklung und Formalisierung des verallgemeinerten Kookkurrenzmodells in Abschnitt 4.5, mit dem die Kookkurrenz zweier beliebiger sprachlicher Strukturen analysiert werden kann. Die Grundidee ist, dass sich zwei sprachliche Strukturen in mindestens einem Knoten überlappen wenn sie kookkurrieren, d. h. dass die Kookkurrenz der beiden ebenfalls eine zusammenhängende sprachliche Struktur ist. Die Instanzen dieses überlappenden Teils fungieren als Analyseeinheiten und werden in die vier Felder der Kontingenztafel kreuzklassifiziert. Die Formalisierung des Modells berücksichtigt die verallgemeinerten Zählweisen und schlägt Umgehungslösungen für das Problem der Inkonsistenzen vor.

Es wird gezeigt, dass die bestehenden Methoden zur Analyse von Zweiwortkookkurrenzen und zur kollostruktionellen Analyse lediglich Spezialfälle des verallgemeinerten Kookkurrenzmodells sind. Es wird außerdem gezeigt, dass einige scheinbare Einschränkungen des verallgemeinerten Kookkurrenzmodells durch das Hinzufügen von zusätzlichen Knoten und Kanten zur Graphstruktur überwunden werden können. Dadurch vergrößert sich die

Bandbreite möglicher Analysen und es wird beispielsweise möglich, explizit nach monotransitiven Verwendungen eines Verbs zu suchen, d. h. nach Vorkommen ohne ausgehende indirekte Objektsrelation.

Die Evaluation in Kapitel 5 hat zwei Schwerpunkte. Im theoretischen Teil wird untersucht, welchen Einfluss aufgeblähte Frequenzen auf populäre Assoziationsmaße haben. Die untersuchten Assoziationsmaße fallen dabei in drei Klassen mit ganz unterschiedlichem Verhalten. Im empirischen Teil werden Experimente mit einer großen Anzahl von Abfragen auf authentischen Sprachdaten durchgeführt. Dabei zeigt sich, dass aufgeblähte Frequenzen bei der Arbeit mit authentischen Sprachdaten kein größeres Problem darstellen und dass die Wahl der Zählweise die Ergebnisse nur begrenzt beeinflusst.

In Kapitel 6 wird eine Reihe von Fallstudien vorgestellt, die von Rekonstruktionen traditioneller kollokationeller und kollostruktioneller Analysen bis hin zu Kookkurrenzen größerer sprachlicher Strukturen reicht. Dabei werden insbesondere die Flexibilität und die vielfältigen Möglichkeiten, die das verallgemeinerte Kookkurrenzmodell bietet, betont.

Aufbauend auf einer hochflexiblen graphbasierten Repräsentation sprachlicher Strukturen ist das verallgemeinerte Kookkurrenzmodell ein vollständiges formales Modell für die Analyse von Kookkurrenzen beliebiger sprachlicher Strukturen. Insbesondere gibt es formale Definitionen für die Kookkurrenz zweier sprachlicher Strukturen, für verschiedene Zählweisen von sprachlichen Strukturen und für die Kreuzklassifikation der Analyseeinheiten in einer 2×2 -Kontingenztafel, auf die eine Vielfalt von Assoziationsmaßen angewandt werden kann.

Acknowledgements

I am very grateful to my supervisor, Stefan Evert, for his guidance and the tremendously useful feedback he gave in our discussions of methodological issues. I am also very grateful to Thomas Herbst, who sparked my interest in the “less tidy and perhaps slightly messy” aspects of language and who reviewed this thesis. Special thanks also go to the remaining members of the examining committee, Ulrich Heid, who was an external reviewer, and Friedrich Michael Dimpel, who was an examiner for my oral defense.

Thanks to my long-term colleagues Besim Kabashi and Peter Uhrig for the many fruitful discussions we have had and for all the pleasantly distracting side projects we have done together.

Thanks also to my (former) colleagues (in alphabetical order) Gabriella Lapesa, Natalie Dykes, Paul Greiner, Philipp Heinrich and Sebastian Wankerl for many interesting discussions and for a great atmosphere.

Last, but certainly not least, heartfelt thanks to my parents for their support and to my wife Susi for her patience and her love.

Contents

Abstract	v
Zusammenfassung	ix
Acknowledgements	xiii
1 Introduction	1
1.1 Cooccurrences	1
1.2 Corpus-based analysis of cooccurrences	5
1.2.1 Cooccurrences of words	6
1.2.2 Collostructional analysis	7
1.3 Goals and limitations	8
1.3.1 Goals	8
1.3.2 Limitations	9
2 Related work	11
2.1 Cooccurrences of pairs of words	11
2.1.1 Positional cooccurrences	12
2.1.2 Relational cooccurrences (including covarying collexeme analysis)	16
2.2 Cooccurrences involving grammatical structures	19
2.2.1 Simple collexeme analysis and distinctive collexeme analysis	19
2.2.2 N-gram-based approaches	23
2.2.3 Multiword collocations based on cooccurrences of syntactic structures	26
2.2.4 Valency pattern extraction	28
2.3 Summary	32
3 Statistical analysis of cooccurrence data	35
3.1 Inferential statistics	35
3.1.1 Statistical models	36
3.1.2 Hypothesis testing	37
3.2 Contingency tables	38

3.3	Association measures	40
3.3.1	Exact tests	41
3.3.2	Approximations to Fisher's exact test	42
3.3.3	Approximations to the binomial test	43
3.3.4	Measures of effect size	44
3.4	Criticism	45
3.4.1	Criticism of statistical hypothesis testing	45
3.4.2	Adequacy of the randomness assumption in corpus linguistics	48
3.4.3	Beyond contingency tables	49
3.5	Summary	52
4	A generalized model for the cooccurrence of linguistic structures	53
4.1	Representing linguistic structures	53
4.1.1	Relations and functions	56
4.1.2	Directed graphs	58
4.1.3	Subgraph isomorphism	60
4.1.4	Partial orders	63
4.1.5	Labeled directed graphs	66
4.1.6	Subsumption	68
4.1.7	Linguistic structures	71
4.2	Counting linguistic structures	73
4.2.1	Counting embeddings	77
4.2.2	Counting subgraphs	77
4.2.3	Counting graphs (sentences)	78
4.2.4	Counting focus points	78
4.2.5	A generalized perspective on counting methods . . .	82
4.3	A preliminary model based on relational cooccurrences . . .	83
4.3.1	Relational cooccurrences revisited	83
4.3.2	Covarying collexeme analysis revisited	85
4.3.3	Generalization to arbitrary linguistic structures . . .	87
4.4	Counting and cross-classifying cooccurrences	90
4.4.1	Counting embeddings	91
4.4.2	Counting subgraphs	94
4.4.3	Counting focus points	101
4.4.4	Counting graphs (sentences)	103
4.4.5	Adequacy of the counting methods	106

4.5	The generalized cooccurrence model	107
4.5.1	Simple collexeme analysis revisited	107
4.5.2	Generalization to arbitrary linguistic structures	109
4.5.3	Formalization	110
4.6	Special cases	116
4.6.1	Collocation as a special case	116
4.6.2	Collostruction as a special case	119
4.7	Seeming limitations	120
4.7.1	Solution: Additional vertices and edges	121
4.8	Summary	124
5	Evaluation of the methodology	125
5.1	Association measures and sample size	125
5.1.1	Thought experiment 1	126
5.1.2	Thought experiment 2	128
5.1.3	Overview of results	130
5.2	Experiments with frequent subgraphs	131
5.2.1	Extraction of frequent subgraphs	132
5.2.2	Query creation	134
5.2.3	Evaluation based on completely lexicalized subgraphs	134
5.2.4	Evaluation based on partly unlexicalized subgraphs	138
5.2.5	Evaluation based on mostly unlexicalized subgraphs	153
5.3	Summary	161
6	Case studies	163
6.1	Cooccurring words	163
6.1.1	Relational cooccurrences based on all relations	163
6.1.2	Relational cooccurrences based on verbal relations	165
6.1.3	Alternative ways of formulating the queries	168
6.2	Words and relations	171
6.2.1	Cooccurrences of words and relations	172
6.2.2	Cooccurrences of words and relation-word pairs	174
6.2.3	Word sketches (cooccurrences of words by relation)	174
6.3	Cooccurring structures	177
6.3.1	Simple collexeme analysis	177
6.3.2	Filtering relational cooccurrences	181
6.3.3	Cooccurrences of larger structures	183
6.4	Summary	189

7	Conclusion and future work	191
7.1	Conclusion	191
7.2	Future work	193
A	Notes on the implementation	197
A.1	Installation	197
A.2	Usage	197
A.2.1	Input formats	197
A.2.2	Convert a corpus into an SQLite3 database	200
A.2.3	Association between two linguistic structures	200
A.2.4	Simple collexeme analysis	202
A.2.5	Relational cooccurrences and covarying collexeme analysis	203
A.2.6	Associated larger structures	205
	Bibliography	209

List of Figures

4.1	Analysis of example sentence 3 (Tesnière, 2015: 343)	54
4.2	Analysis of example sentence 4 (Hudson, 2010: 178)	55
4.3	SD analysis of example sentence 5	55
4.4	Directed graph with loop and parallel edges	59
4.5	Two isomorphic graphs (Petersen graph)	64
4.6	A couple of graphs isomorphic to subgraphs of the Petersen graph	64
4.7	More easily recognizable drawings of subgraphs of the Petersen graph	65
4.8	A couple of graphs not isomorphic to subgraphs of the Petersen graph	65
4.9	Hasse diagram of a part of the Stanford Dependencies hierarchy	67
4.10	Hasse diagram of the subset ordering of subsets of $\{a, b, c\}$	67
4.11	An example graph and its poset of labels	70
4.12	A couple of graphs subsuming the graph in Figure 4.11a	70
4.13	A couple of graphs not subsuming the graph in Figure 4.11a	71
4.14	Hasse diagram of Σ	72
4.15	Words-and-dependencies representation of example sentence 6	72
4.16	Representation of example sentence 6 with annotation as graph structure	74
4.17	Representation of example sentence 6 with annotation as complex labels	75
4.18	Example Σ for demonstration purposes	75
4.19	G_1 and G_2 occur in G_3	76
4.20	What is the frequency of H_1 in H_2 ?	76
4.21	What is the frequency of G_1 in G_2 ?	79
4.22	Examples of focus point vertices	81
4.23	Query graphs for relational cooccurrences	84
4.24	G_A and G_B cooccur in G_C	85
4.25	Query graphs for covarying collexeme analysis	86

4.26	Cooccurrence of two arbitrary linguistic structures within another arbitrary linguistic structure	88
4.27	Hasse diagram of the linguistic structures	90
4.28	SD analysis of a part of example sentence 7	91
4.29	Examples of query and sentence graphs	94
4.30	Number of embeddings of G_{N_k} in G_{S_n}	95
4.31	G_S is subsumed by G_A and G_B but not by G_C	96
4.32	G_S is subsumed by G_A and G_B but not by G_C	97
4.33	Number of subgraphs for G_{N_k} and G_{S_n}	100
4.34	The focus point vertex r in G_S is part of G_A and G_B but not of G_C	101
4.35	G_S is subsumed by G_A and G_B but not by G_C	104
4.36	Query graphs for simple collexeme analysis	108
4.37	Query graphs for the generalized model	110
4.38	G_S matches G_A and G_B but not G_C	110
4.39	Hasse diagram of the linguistic structures in the generalized cooccurrence model	112
4.40	Inconsistencies even when counting embeddings	114
4.41	Query graphs for relational cooccurrences	117
4.42	Query graphs for segment-based cooccurrences	118
4.43	Query graphs for simple collexeme analysis	119
4.44	Query graphs for covarying collexeme analysis	120
4.45	Representation of example sentence 6 with added edges for indirect relations	122
4.46	Representation of an example sentence with added edges for linear order	122
4.47	Representation of an example sentence with added “negated” edges for absent relations	123
5.1	The vertices of the frequent subgraph have been randomly assigned to V_i , $V_{\alpha'}$ and $V_{\beta'}$	135
5.2	Query graphs for the structure “could not be more different” as a cooccurrence of “not be more different” and “could be different”.	135
5.3	Inconsistencies for completely lexicalized queries	137
5.4	Degree of inflation for completely lexicalized queries (number of embeddings per focus point)	137
5.5	Scatterplots and Spearman correlation coefficients for different counting methods based on the Dice coefficient (completely lexicalized queries)	139

5.6	Scatterplots and Spearman correlation coefficients for different counting methods based on <i>t</i> -score (completely lexicalized queries)	140
5.7	Scatterplots and Spearman correlation coefficients for different counting methods based on log-likelihood (completely lexicalized queries)	141
5.8	Distribution of association scores for different counting methods (completely lexicalized queries)	142
5.9	Query graphs for the structure “could not VERB more ADJECTIVE” as a cooccurrence of “not VERB more ADJECTIVE” and “could VERB ADJECTIVE”.	143
5.10	Inconsistencies for partly unlexicalized queries	144
5.11	Query graphs for the “VERBed that he VERBed” structure yielding inconsistent results	145
5.12	Query graphs for the “the NOUN of the NOUN” structure yielding inconsistent results	146
5.13	Degree of inflation for partly unlexicalized queries (number of embeddings per focus point)	147
5.14	Scatterplots and Spearman correlation coefficients for different counting methods based on the Dice coefficient (partly unlexicalized queries)	148
5.15	Scatterplots and Spearman correlation coefficients for different counting methods based on <i>t</i> -score (partly unlexicalized queries)	149
5.16	Scatterplots and Spearman correlation coefficients for different counting methods based on log-likelihood (partly unlexicalized queries)	150
5.17	Query graphs for the “it (BE) DETERMINER short NOUN” structure	151
5.18	Distribution of association scores for different counting methods (partly unlexicalized queries)	152
5.19	Query graphs for the structure “MODAL not VERB more ADJECTIVE” as a cooccurrence of “not VERB more ADJECTIVE” and “MODAL VERB ADJECTIVE”.	154
5.20	Inconsistencies for mostly unlexicalized queries	155
5.21	Degree of inflation for mostly unlexicalized queries (number of embeddings per focus point)	156

5.22	Scatterplots and Spearman correlation coefficients for different counting methods based on the Dice coefficient (mostly unlexicalized queries)	157
5.23	Scatterplots and Spearman correlation coefficients for different counting methods based on <i>t</i> -score (mostly unlexicalized queries)	158
5.24	Scatterplots and Spearman correlation coefficients for different counting methods based on log-likelihood (mostly unlexicalized queries)	159
5.25	Query graphs for the “is ADJECTIVE to VERB” structure	160
5.26	Distribution of association scores for different counting methods (mostly unlexicalized queries)	161
6.1	Query graphs for relational cooccurrences based on all relations	164
6.2	Query graphs for relational cooccurrences based on verbal relations	166
6.3	Query graphs for node-based cooccurrences	168
6.4	Query graphs for collocate-based cooccurrences	170
6.5	Query graphs for cooccurrences of words and relations	173
6.6	Query graphs for cooccurrences of words and word-relation pairs	175
6.7	Query graphs for cooccurrences within a specific type of relation	176
6.8	Query graphs for simple collexeme analysis	182
6.9	Query graphs for adjectival modifiers of <i>bank</i> as subject	184
6.10	Query graphs for structures associated with <i>give damn</i>	186
6.11	Query graphs for the first approach	187
6.12	Query graphs for the second approach	187
A.1	Larger structures most strongly associated with monotransitive <i>give</i>	207

List of Tables

1.1	Types of cooccurrences	4
1.2	2 × 2 contingency table	7
2.1	Contingency table for <i>accident waiting to happen</i> taken from Stefanowitsch and Gries (2003: 219)	22
3.1	2 × 2 contingency table	38
3.2	2 × 2 contingency table of expected values	40
4.1	Counting methods and their problems	107
5.1	Behavior of association measures in the scenarios discussed in Sections 5.1.1 and 5.1.2	131
5.2	Distribution of extracted subgraphs	134
6.1	Dependents and governors of the verb <i>give</i>	165
6.2	Dependents and governors of the verb <i>give</i>	167
6.3	Dependents and governors for node-based cooccurrences	169
6.4	Dependents and governors for collocate-based cooccurrences	171
6.5	Outgoing and incoming dependency relations of the verb <i>give</i>	173
6.6	Dependents and governors of the verb <i>give</i> and their dependency relations	175
6.7	Word sketch for some of the outgoing dependency relations of the verb <i>give</i>	178
6.8	Word sketch for some of the incoming dependency relations of the verb <i>give</i>	180
6.9	Syntactic structures associated with the verb <i>give</i>	183
6.10	Adjectival modifiers of <i>bank</i> as nsubj (left) and dobj (right)	185
6.11	Structures associated with <i>give damn</i>	188
6.12	Adverb-verb-direct object triples obtained by both approaches	189
6.13	Adverb-verb-direct object triples for the noun <i>head</i>	190

1 Introduction

1.1 Cooccurrences

In linguistics, the study of cooccurrences has not only a fairly long tradition – it has also greatly contributed to a major paradigm shift in the subject.

One type of cooccurrence, the cooccurrence of words, has been studied extensively under the name of collocation. John Rupert Firth, whose famous dictum “You shall know a word by the company it keeps!” (Firth, 1957: 11) is probably one of the most quoted sentences in corpus linguistics, is usually credited with the systematic introduction of the concept into linguistics (cf. Bartsch, 2004: 30). He illustrates it with examples like “don’t be such an ass”, “They are milking the cows” (Firth, 1957: 11–12) or “dark night” Firth (1957: 196). The closest he comes to a definition (“Exactly what Firth meant by collocability is never made clear”, Lyons (1977: 612) complains) are probably the following two quotes:

The habitual collocations in which words under study appear are quite simply the mere word accompaniment, the other word-material in which they are most commonly or most characteristically embedded.

Firth (1957: 11–12)

The collocation of a word or a ‘piece’ is not to be regarded as mere juxtaposition, it is an order of *mutual expectancy*. The words are mutually expectant and mutually prehended.

Firth (1957: 12)

Today, the term collocation commonly means two slightly different things. 1) In the tradition of Firth, there is what Herbst (1996) calls the “statistically oriented approach” to collocation where collocation usually means that (usually two) words “co-occur more often than their respective frequencies and the length of the text in which they appear would predict.” (Jones and Sinclair, 1974: 19) In this approach, Firth’s mutual expectancy is translated into statistical association measures. Evert (2009) calls this kind of collocation “empirical collocations”. 11) The second approach to collocation, Herbst (1996) calls it the “significance oriented approach”, originates in lexicography and has a much tighter definition of what collocation is. Here, the term collocation is

usually used for combinations of a semantically autonomous word (called *Basis* by Hausmann, 1984), e. g. *tea*, and a another word that is semantically dependent on it (called *Kollokator*), e. g. *make*. This kind of word combination, that Evert (2009) calls “lexical collocation”, is important in lexicography and language learning, since only *make tea* is idiomatic English, whereas similar combinations like *cook tea* are not. It is important to note that both notions of collocation are based on the cooccurrence of two words and are as such “applications” of cooccurrences that use different additional criteria (statistical or semantic) to characterize some cooccurrences as collocations.¹ An excellent overview of the history of the term and the status of collocation in mainstream linguistics and lexicography as well as of defining criteria is given by Bartsch (2004: 27–78).²

Another concept that Firth introduced into linguistics is that of colligation, i. e. the interrelation “of word and sentence classes or of similar categories” (Firth, 1957: 13). Today, the term is usually used to describe “the co-occurrence of grammatical choices” or “grammatical phenomena” (Sinclair, 2004: 41, 142) or “the co-occurrence of words with grammatical choices” (Sinclair, 2004: 174).³ In his own examples, e. g. the cooccurrence of *naked eye* with a leading preposition or the cooccurrence of *true feelings* with a possessive, Sinclair (2004: 32, 35) mainly illustrates the latter variant. At its heart, colligation is just another “application” of cooccurrences. Instead of with cooccurrences of words, it deals with cooccurrences of words with word classes or syntactic functions. Similar to the notion of empirical collocation, colligation usually involves some kind of statistical analysis of the cooccurrence data.

A variety of other types of cooccurrences is analyzed under the cover term collostructional analysis that was introduced by Stefanowitsch and Gries (2003). Collostructional analysis deals with cooccurrences that involve

¹ As Evert (2009: 1213) points out, the term collocation has also been used in computational linguistics for what is nowadays usually called “multiword expressions”, i. e. “for any lexicalised word combination that has idiosyncratic semantic or syntactic properties and may therefore require special treatment in a machine-readable dictionary or natural language processing system.”

There has also been what Herbst (1996) calls a “text oriented approach” to collocations by Halliday and Hasan (1976). However, their definition of collocation as “a cover term for the cohesion that results from the co-occurrence of lexical items that are in some way or other typically associated with one another, because they tend to occur in similar environments” (Halliday and Hasan, 1976: 287) has not played a role in the literature (cf. Bahns, 1996: 26) and has later been discarded by Hasan (1984) herself.

² For the earlier history of the term see also Mitchell (1975: 134).

³ There are also very broad notions of colligation like that of Hoey (2004: 174) where textual colligation comprises the preference of a lexical item for lexical chains, theme-rheme relations, semantic relations, genres, as well as positional preferences within a text.

constructions in the sense of Construction Grammar, specifically the Goldberg (1995, 2006) variant:

C is a CONSTRUCTION iff_{def} C is a form-meaning pair $\langle F_i, S_i \rangle$ such that some aspect of F_i or some aspect of S_i is not strictly predictable from C 's component parts or from other previously established constructions.

(Goldberg, 1995: 4)

The three variants of collostructional analysis proposed so far are: I) Simple collexeme analysis, where the association (attraction or repulsion) between a construction and the words in a particular slot of this construction is analyzed, e.g. the association between the ditransitive construction and the verbs in this construction (Stefanowitsch and Gries, 2003: 227–230); II) distinctive collexeme analysis, which analyzes the association between a word and a pair of alternating constructions this word occurs in, e.g. between the verb *give* and the ditransitive and the *to*-dative (Gries and Stefanowitsch, 2004b: 104–107); III) covarying collexeme analysis, where the association between a pair of words occurring in two different slots of a construction is analyzed, e.g. between the two verbs in the *into*-causative (as in “force someone into making a decision”; Gries and Stefanowitsch, 2004a). In all three variants, association strength is determined by a statistical test. We can say that collostructional analysis is an extension of the empirical notion of collocation that is concerned with other types of cooccurrences, i.e. words with constructions or words with words within constructions.

Table 1.1 gives a brief overview of the types of cooccurrence introduced so far and the names under which they have been studied in corpus linguistics. Note, however, that in some cases the terminological distinction is not as clear-cut as the table might suggest. The cooccurrence of a word with a construction is usually investigated under the name of collostruction (simple collexeme analysis) but if the construction is a simple grammatical relation, we could also call it a colligation. Similarly, the cooccurrence of a word with word within a construction is the domain of a type of collostructional analysis (covarying collexeme analysis) but if the construction is a simple grammatical relation, the phenomenon is also studied under the name of collocation.

The impact that the study of cooccurrences has had on the field of linguistics can only be understood against the background of generative grammar. For much of the second half of the twentieth century, linguistic research was dominated by generative grammar, a highly formalized approach to grammar that seeks “maximally abstract and general representations of grammatical form and meaning” (Croft and Cruse, 2004: 1). Typical for this approach is the distinction between the so-called core of a language that comprises all

type of cooccurrence	name in linguistics	example
word with word	collocation	<i>dark night, make tea</i>
word with word class	colligation	PREP DET <i>naked eye</i> , POSS <i>true feelings</i>
word with grammatical relation	colligation	<i>attention</i> as direct object
word with construction	collostruction	<i>give</i> + ditransitive, <i>trick</i> + into-causative
word with alternating constructions	collostruction	<i>give</i> + ditransitive vs. <i>give</i> + to-dative
word with word within construction	collostruction	<i>force into making</i> , <i>fool into thinking</i>

Table 1.1: Types of cooccurrences

the regular and well-behaved aspects that are supposed to make up most of a language and the so-called periphery that is made up of “phenomena that result from historical accident, dialect mixture, personal idiosyncrasies and the like.” (Chomsky, 2015: 17) Of course, the focus of generative grammar has almost exclusively been on the core. In this approach, language is represented as a set of rather general rules and a lexicon that more or less only provides fillers for the slots that are opened by operating the rules (also, any “exceptions” will have to be stored in the lexicon).

This view of language has been challenged (among others) by corpus linguists studying cooccurrences. The extent to which specific words, grammatical choices and whole constructions interrelate with each other led Sinclair (1991: 110) to the formulation of the idiom principle, according to which “a language user has available to him or her a large number of semi-preconstructed phrases that constitute single choices, even though they might appear to be analysable into segments.” Why did generative grammarians overlook these interrelations, one might ask? One contributing factor is given by (Hanks, 2010: 143): “It is a surprising but well-established fact that human beings are unable to recall to the conscious mind the details of their unconscious linguistic behaviour.” This means that cooccurrences had been hiding in plain sight, inaccessible by introspection, only to be revealed by the analysis of large amounts of authentic language data, i. e. corpus linguistics. The insight gained by the corpus linguistic study of phraseology is probably best summarized by the following quote:

[I]n the construction of a grammar, more is needed than a system of general grammatical rules and a lexicon of fixed words and phrases. Those linguistic

processes that are thought of as irregular cannot be accounted for by constructing lists of exceptions: the realm of idiomaticity in a language includes a great deal that is productive, highly structured, and worthy of serious grammatical investigation.

Fillmore et al. (1988: 534)

What is so special about the conclusion that corpus linguists drew from the analysis of cooccurrences, namely that a lot of linguistic knowledge has to be item-specific and that the “slightly messy” (Herbst, 2007: 17) periphery is at least as important as the clean and regular core, is that a lot of other branches of linguistics independently came to the same conclusion! Syntacticians from a valency-theoretic background, foreign language linguists, usage-based cognitive linguists and linguists working on usage-based approaches to language learning all came to the same conclusion: Unpredictable item-specific properties are a central part of language and if we want to arrive at a better understanding of how language works, it is these properties that we have to study. Or, as Michael Stubbs puts it:

[W]hen scholars set out from different starting points within different traditions, use data of different kinds and independent arguments, but nevertheless arrive at similar conclusions, then the conclusions are worth studying closely, because the convergence of views is *prima facie* evidence that they are well founded.

Stubbs (2009: 27)

1.2 Corpus-based analysis of cooccurrences

The standard approach to the analysis of cooccurrences that is predominant in the literature on empirical collocations and collocations is based on the statistical analysis of cooccurrence data with the help of contingency tables. This approach is also the focus of the current work. However, it is worth mentioning that there are also other approaches to the analysis of cooccurrences, especially in the literature on multiword expressions, that complement a conventional statistical analysis with other automatic methods for determining, for example, the compositionality (Katz and Giesbrecht, 2006; Kiela and Clark, 2013; Yazdani et al., 2015), non-modifiability (Nissim and Zaninello, 2013; Squillante, 2014) or non-substitutability (Pearce, 2001; Farahmand and Henderson, 2016) of word combinations. There are also approaches that try to combine multiple source of information via machine learning techniques (e. g. Tsvetkov and Wintner, 2014).

The conventional approach to the statistical analysis of cooccurrences will be covered in greater detail in Sections 2.1 and 2.2.1 and in Chapter 3. Here, we only give a very brief summary.

1.2.1 Cooccurrences of words

As noted above, the two notions of collocation are both based on the cooccurrence of words: Empirical collocations are cooccurrences that satisfy some statistical criteria, lexical collocations are cooccurrences that satisfy some semantic criteria. In practice, a statistical analysis of cooccurrence data obtained from a corpus is usually also the first step if one is looking for lexical collocations for inclusion in a dictionary (cf. Lea, 2007). A prototypical extraction pipeline starts by obtaining frequency counts for cooccurrences of pairs of words. Then, the cooccurrences are ranked by some statistic that measures the association between the two cooccurring words. Additional filters, e. g. frequency thresholds, can be applied. Finally, the ranked list of “collocation candidates” (Heid, 1998: 301) is manually inspected by a lexicographer who assesses if a given candidate fulfills the necessary criteria for inclusion in the dictionary.

Let us illustrate the general approach with verb-noun pairs. First, we have to extract all cooccurrences of verbs and nouns from the corpus. Then, in order to determine the statistical association between a verb A and a noun B , we have to cross-classify all verb-noun pairs. For each pair $\langle V, N \rangle$, we have to answer two questions: “Is the verb slot of the pair filled by A ?” and “Is the noun slot of the pair filled by B ?”. We answer those two questions for every verb-noun pair and can summarize the answers in a contingency table (Table 1.2). In cell O_{11} of the contingency table, we write down how many verb-noun pairs have A as the verb and B as the noun. The other cells of the table are filled accordingly. O_{12} contains the number of pairs which have A as the verb and other words than B as the noun, in O_{21} we count how many pairs have the noun B and verbs other than A and O_{22} contains the number of pairs that contain neither the verb A nor the noun B . R_1, R_2, C_1 and C_2 are called the marginal frequencies and contain row or column sums. The sample size N corresponds to the total number of verb-noun pairs that have been classified. Once we have organized the cooccurrence frequencies in such a 2×2 contingency table, there is a wealth of association measures that we can apply to come up with an indicator of how strongly the two words are associated.⁴

⁴ Given the large number of available association measures, it is only natural that the question of which measure works best for a given purpose arose. In the literature, there is a variety

	$N = B$	$N \neq B$	
$V = A$	O_{11}	O_{12}	R_1
$V \neq A$	O_{21}	O_{22}	R_2
	C_1	C_2	N

Table 1.2: 2×2 contingency table

The big question is: What constitutes a cooccurrence? Or: How do we extract and count the verb-noun pairs? Early approaches considered all cooccurrences “of two items in a text within a specified environment” (Jones and Sinclair, 1974: 19). That specified environment was either a predetermined segment, e. g. a sentence or a paragraph, or a moving “window” of a predetermined number of words (called collocational span). With the availability of fast and robust automatic parsers, it became feasible to use a syntactic definition of cooccurrence and to consider, for example, all pairs of verbs and nouns that are connected by a syntactic dependency relation.

But no matter how cooccurrence is defined: If we want to perform a meaningful statistical analysis of cooccurrence data, we must have a formally worked out statistical model with a clear interpretation. For pairs of words, such models have been worked out for example by Evert (2004). Failure to correctly implement a proper statistical model leads to results that are in the best case difficult to interpret and in the worst case misleading. This can happen either by accident, e. g. due to lack of awareness or due to programming errors, or on purpose, e. g. when deviation from a proper statistical model is perceived to have practical benefits (cf., for example, Sinclair, 1991: 115).

1.2.2 Collostructional analysis

Collostructional analysis is a generalization of the traditional approach to cooccurrences of words to cooccurrences of words and constructions and is also based on the statistical analysis of 2×2 contingency tables. This is especially obvious for covarying collexeme analysis where we are interested in the association between two words cooccurring with each other within a construction. In this case, we need to extract all word pairs from the relevant

of approaches for the quantitative and qualitative evaluation of association measures, for example Evert and Krenn (2001), Pearce (2002), Pecina (2005), Pecina and Schlesinger (2006), Wermter and Hahn (2006), Pecina (2010), Uhrig and Proisl (2012), Kilgariff et al. (2014b) and Evert et al. (2017).

slots of the construction and cross-classify them exactly as described in the previous section. The construction only plays a role in defining what constitutes a cooccurrence.

The cooccurrence models for collostructional analysis have not been worked out in such great formal detail as those for cooccurrences of words. In Sections 2.1.2 and 2.2.1, where we take a closer look at collostructional analysis, we will see that simple collexeme analysis is a good example of what happens if an explicit formal model is not available.

1.3 Goals and limitations

1.3.1 Goals

The view that item-specific knowledge is central to an understanding of language is on a good way to become the prevalent view in linguistics. One perspective on item-specificity is the analysis of cooccurrences, i. e. to look for patterns of usage in authentic language data. The standard way to study cooccurrences is the approach sketched above that is based on the statistical analysis of 2×2 contingency tables. Currently, this method for analyzing cooccurrences is limited to cooccurrences of words with words, i. e. collocations, or to cooccurrences of words with linguistic structures, i. e. collostructions. (In this work, we will try to avoid the term construction mainly due to the very specific sense in which it is used in Construction Grammar where it implies a meaning component. We will speak of (purely formal) linguistic structures instead.) For the analysis of two-word cooccurrences, there exist clearly defined formal models, the models for the related approaches to collostructional analysis have not been formalized to the same degree. A methodological framework for a similar analysis of cooccurrences of linguistic structures with linguistic structures, e. g. of the *into*-causative with the ditransitive as in the following example, is currently not available:⁵

(1)_{BNC: FNU 2078} You can't bully him into giving you better parts.

And that is exactly the point where the current work starts. Focusing on the contingency table approach to cooccurrences, our goals are as follows:

1. Generalize the corpus-based methods for cooccurrences of words, i. e. collocations, and of words and structures, i. e. collostructions, so that cooccurrences of arbitrary linguistic structures can be analyzed.

⁵ Examples marked with the subscript 'BNC' have been extracted from the British National Corpus, distributed by Oxford University Computing Services on behalf of the BNC Consortium. All rights in the texts cited are reserved.

2. Develop an explicit formal model for the statistical analysis of those cooccurrences. This generalized cooccurrence model should be a true generalization of the existing approaches, i. e. the existing models for collocations and collocations should be special cases of the new model.
3. Evaluate this new model, i. e. make sure that it is robust and well-behaved both from a theoretical point of view and when applied to actual corpus data.
4. Present case studies that illustrate the possibilities that the new model offers.
5. Provide an actual implementation of the model as free software.

To achieve those goals, we will have to answer, i. a., the following questions: How can linguistic structures be represented in a formal model? How can we determine the frequencies of linguistic structures? (Spoiler: It is not quite as simple as we might expect.) What constitutes a cooccurrence of two linguistic structures? How can we determine the items that are to be cross-classified in a statistical analysis?

1.3.2 **Limitations**

The main goal of this work is to develop a formal model for the analysis of cooccurrences of arbitrary linguistic structures. To achieve that goal, we work within an established framework for the statistical analysis of cooccurrences that is based on 2×2 contingency tables. While it is our aim to generalize that approach to allow for cooccurrences of two arbitrary linguistic structures that can each contain an arbitrary number of words, it is not within the scope of this work to consider cooccurrences of an arbitrary number of structures, although extensions to multidimensional contingency tables (cf., for example, Villada Moirón, 2005 or Stefanowitsch and Gries, 2005) should be possible, of course. Similarly, we do not consider the extended approaches suggested in the literature on multiword expressions mentioned above. Before we can construct a complex architecture, we must lay solid foundations. Once a properly worked out formal model for the cooccurrences of arbitrary linguistic structures is available, the extensions and additional features used in the multiword expressions community for combinations of words can be transferred to the analysis of combinations of linguistic structures.

The restrictions imposed by sticking to the standard approach based on 2×2 contingency tables are offset by the advantage of using an established, well-understood statistical model for which a large inventory of association measures is available. (Note that this excludes measures taking into account other information than that represented in the contingency table, like for example entropy-based measures; cf. Pecina, 2005: 15 for an overview.) Also, by focusing on that particular framework, we can make sure that the generalized cooccurrence model developed in this work is compatible to the vast body of existing literature on collocation and collostruction.

2 Related work

There is a vast body of literature on cooccurrences. Prominent topics include methods for identifying or extracting cooccurrences, lexical, syntactic or semantic aspects of cooccurrences, their classification, their representation in lexical resources, their annotation in corpora, their treatment in downstream processing steps such as syntactic parsing, semantic analysis or machine translation and the evaluation of those extraction and processing techniques, the psycholinguistic reality of cooccurrences and their role in first and second language acquisition, the multilingual or cross-linguistic study of cooccurrences and their application in computer-aided language learning. As such, giving a complete overview of all things related to cooccurrences is neither feasible nor conducive to the aims of the current work. A good starting point for the interested reader are the proceedings of the annual Workshop on Multiword Expressions organized by a section of the ACL Special Interest Group on the Lexicon (SIGLEX-MWE).¹

In this chapter, we focus on work that is more closely related to our goal of developing a formal model for the statistical analysis of cooccurrences of arbitrary linguistic structures. The model we develop is a generalization of the traditional approaches to cooccurrences of pairs of words and of words and linguistic structures that are based on the statistical analysis of 2×2 contingency tables. We give an overview of those approaches in Sections 2.1 and 2.2.1. The statistical analysis of cooccurrence data is discussed in greater detail in Chapter 3. The one thing that the works presented in Section 2.2 have in common is that they deal, broadly speaking, with some kind of cooccurrence that involves some kind of larger grammatical structure that can be anything from a sequence of part-of-speech tags to a parse tree.

2.1 Cooccurrences of pairs of words

While there are approaches that rely solely on distributional methods for a “semantics-driven recognition of collocations” (Rodríguez-Fernández et al., 2016), the usual approach to the analysis of cooccurrences involves the statistical analysis of cooccurrence frequencies that are organized in a contingency

¹ <http://multiword.sourceforge.net/>

table (cf. Section 3.2). In order to populate the contingency table in a meaningful way, we have to be clear about what constitutes a cooccurrence and how we count them, i. e. what the unit of analysis is (the elements that will be cross-classified in the contingency table).

Evert (2004), among other things, identifies three classes of procedures used in the literature for obtaining cooccurrence data and works out formal statistical models for their interpretation. The three types of cooccurrences are segment-based cooccurrences, distance-based cooccurrences and relational cooccurrences. The first two can also be referred to as positional cooccurrences. The outline of this section follows that distinction.

2.1.1 Positional cooccurrences

Manually annotating corpora of substantial size with syntactic analyses is expensive and time-consuming. Therefore, when robust automatic parsers were not as readily available as nowadays, the analysis of positional cooccurrences, i. e. of words that cooccur either within some distance from each other or within some segment of text, was often the only viable option.

Even today and for well-resourced languages (meaning taggers and parsers are readily available), there are still proponents of that approach. Corpus-driven linguists² disapprove of “pre-corpus models of language” (Sinclair, 2004: 192) and reject tagging or parsing: “In corpus-driven linguistics you do not use pre-tagged text, but you process the raw text directly and then the patterns of this uncontaminated text are able to be observed.” (Sinclair, 2004: 191) In that spirit, Lehr (1996: 4) describes the major goal of her work as “[die] Entwicklung eines Kollokationsanalysemodells, welches sich ausschließlich [sic] auf die formlichen und quantitativen Eigenschaften von Korpora respektive der darin enthaltenen Texte stützt.”

Functionality for analyzing positional cooccurrences is often included in corpus linguistic tools like AntConc (Anthony, 2014), BNCweb (CQP-Edition) (Hoffmann and Evert, 2006; Hoffmann et al., 2008), CQPweb (Hardie, 2012) or the interface built by Mark Davies for his collection of corpora,³ e. g. the Corpus of Contemporary American English (COCA) (Davies, 2008–). The advantage is, of course, that it works on plain unannotated texts and does not require language-dependent components like parsers.

² For a discussion of corpus-based vs. corpus-driven approaches to corpus linguistics, see, for example, Tognini-Bonelli (2001: 65–100).

³ <http://corpus.byu.edu/>

Segment-based cooccurrences

The statistical analysis of words that cooccur within some segment of text is among the earliest approaches to the analysis of word cooccurrences. Even in the early literature on the subject, a broad range of segments has been suggested depending on the use case: Cooccurrence within sentences (Giuliano, 1965; Salton, 1965), within paragraphs (Dennis, 1965) or within documents (Maron and Kuhns, 1960; Salton, 1965). Giuliano (1965) even analyzes cooccurrences within syntactic subtrees of sentences (although he only has 7 000 words of syntactically analyzed text available). The segment-based approach to the analysis of cooccurrences seems to be less used in corpus linguistics but remains popular in information retrieval (e. g. Bunesco et al., 2006; Leydesdorff and Vaughan, 2006). Notable uses include the identification of antonymous adjectives via cooccurrence within sentences (Justeson and Katz, 1991) or the identification of synonyms via cooccurrence within documents (Terra and Clarke, 2003).⁴ For the purpose of extracting lexical collocations, Bartsch and Evert (2014) found that sentence-based cooccurrences have a higher coverage than distance-based or relational cooccurrences but yield lower precision values at a given level of recall. The segment-based approach has a clear statistical interpretation and is easy to implement.

For segment-based cooccurrences, the corpus has to be divided into non-overlapping segments, e. g. sentences, paragraphs, documents, etc. Those segments are the units of analysis and the total number of segments is the sample size N . If we want to determine the association between the two words A and B , we cross-classify the segments according to the following two criteria: “Does word A occur in the segment?” and “Does word B occur in the segment?”. In cell O_{11} of the contingency table, we enter the number of segments which contain both word A and word B . Similarly, cells O_{12} and O_{21} contain the numbers of segments which contain only either A or B . In cell O_{22} , we enter the number of segments which contain neither A nor B . This means that every segment in the corpus has been classified and counts towards one cell of the contingency table. Once the contingency table has been populated, we can apply an association measure to determine the strength of the association between words A and B .

For the fully worked out formal model, cf. Evert (2004: 66–68).

⁴ Terra and Clarke (2003) found that document-based cooccurrences work best with an additional constraint where only documents in which the two words cooccur within a maximal distance of 16 words from each other are counted as cooccurrences.

Distance-based cooccurrences

Distance-based cooccurrences were the most popular approach to the analysis of cooccurrences in corpus linguistics and lexicography until reasonably fast and robust tools for syntactic analysis became readily available. By distance-based cooccurrences we mean cooccurrences where the one word occurs within a short space of the other. That short space, the maximal distance between the two words, is often called the collocational span and is usually expressed as a number of orthographic tokens.⁵ A distinction is made between the word under investigation, the node, and the words cooccurring with it within the collocational span, the collocates. As we will see below, this is not merely a terminological distinction but has an influence on the statistical analysis of the cooccurrences.

A special case of distance-based cooccurrences are cooccurrences within a distance of 1, i. e. adjacent words. In the literature, the analysis of adjacent words is usually not limited to pairs of words. Instead, longer sequences of n adjacent words, called n -grams, are analyzed. Early approaches often only looked at n -gram frequencies and did not include a statistical analysis of the cooccurrence data. Examples include Choueka (1988), although he hints at experiments with calculating “a ‘binding degree’ of an expression as a measure of the degree with which the different words in the expression ‘attract each other’” (Choueka, 1988: 619), Kjellmer’s (1994) *Dictionary of English Collocations* and the work by Justeson and Katz (1995) on terminological noun phrases. Where part-of-speech tagged data is available, the extraction process usually involves a filter based on part-of-speech tags. Cooccurrences of adjacent words have also been analyzed with the help of statistical association measures. Smadja (1993), for example, uses the z -score (cf. Section 3.3.3) for extracting “rigid noun phrases” and Zaiu Inkpen and Hirst (2002) apply the association measures implemented in Ted Pedersen’s Bigram Statistics Package (BSP), a predecessor of the N -gram Statistics Package (NSP) (Pedersen et al., 2011), to cooccurrence data in order to differentiate between near-synonyms.

In the general case, the collocational span is larger than 1 and the analysis is not limited to pairs of adjacent words. This is probably the most common approach to distance-based cooccurrences and has been used, for example,

⁵ Chaudhari et al. (2011) devise a whole new significance test that could also be called distance-based. They compare the actual distribution of the distances between two words (within documents in which both of them occur) to the expected distribution under a random null model. According to this test, words are said to be associated if they cooccur closer to each other than expected under the null model.

by Jones and Sinclair (1974), Church and Hanks (1990), Sinclair (1991), Breidt (1993), Bartsch (2004) and Terra and Clarke (2004). The widespread use of this approach is also helped by the fact that it is implemented in some of the most popular pieces of corpus linguistic software, including AntConc (Anthony, 2014), BNCweb (CQP-Edition) (Hoffmann and Evert, 2006; Hoffmann et al., 2008), CQPweb (Hardie, 2012) or Mark Davies' interface at <http://corpus.byu.edu/>. Smadja (1993) also follows the window-based approach but applies additional heuristics and filters based, for example, on the distance between the two words and on syntactic patterns. Dias (2003) uses the window-based approach in combination with syntactic patterns (sequences of part-of-speech tags) that are learned from the corpus in an unsupervised fashion.

The formal model for distance-based cooccurrences is a bit more complicated than those for segment-based or relational cooccurrences, especially for the general case where the distance between the two words can be larger than 1, i. e. when non-adjacent pairs are also accounted for. For distance-based cooccurrences, we have to decide which word is the node and which is the collocate. Let us say we analyze adjective-noun cooccurrences and are interested in the cooccurrence of the adjective *A* with the noun *B* and decide that the noun *B* is our node. What we have to do to populate the contingency table is to classify all adjectives according to the following two criteria: "Does the adjective occur within the specified distance from an instance of the noun *B*?" and "Is the adjective an instance of *A*?"

For the analysis of distance-based cooccurrences it really matters which word is the node and which is the collocate. If we swap the roles of *A* and *B*, i. e. if the adjective *A* becomes the node and the noun *B* the collocate, we will get entirely different results since the sample size *N* will be different (number of nouns instead of number of adjectives), as well as the marginal frequencies.

"It is not easy to implement frequency counts for distance-based cooccurrences in an efficient manner." (Evert, 2004: 69) Quickly determining the marginal frequencies is complicated by the fact that local windows can overlap. The implementation becomes even (slightly) more difficult if the two cooccurring words do not come from two disjoint sets (here: adjectives and nouns).

For the fully worked out formal model, cf. Evert (2004: 68–70).

2.1.2 Relational cooccurrences (including covarying collexeme analysis)

With the availability of sufficiently fast and robust techniques for syntactic analysis, it became possible to syntactically define cooccurrences of words – typically by specifying the kind of grammatical relation that should hold between the two words, hence the name relational cooccurrences – and to analyze them on the basis of large, automatically annotated corpora. Having syntactic parsing as an additional preprocessing step is usually beneficial. It has been shown, for example, that when used for extracting collocation candidates for lexicographical purposes, relational cooccurrences based on dependency analyses generally outperform positional cooccurrences (see, for example, Uhrig and Proisl (2012) and Bartsch and Evert (2014)).⁶ The exact type of syntactic analysis can range from a partial syntactic analysis or shallow parsing to a full syntactic analysis with complete constituent or dependency structures.

A partial or shallow syntactic analysis can often be achieved with considerably lower requirements regarding computing power and processing time. Church et al. (1989) use the deterministic Fidditch parser (Hindle, 1983) that produces partial analyses of sentences it cannot completely analyze to extract different types of cooccurrences from a news corpus. Basili et al. (1994) extract pairs and triples of syntactically related words using what they call a “not-so-shallow’ parser” based on Dahl’s (1989) discontinuous grammar. Kermes and Heid (2003) use YAC, the recursive chunker by Kermes (2003), to identify German adjective-verb collocations. Wermter and Hahn (2006) perform a shallow syntactic analysis to extract German preposition-noun-verb combinations. In the SketchEngine (Gilgarriff et al., 2004), the grammatical relations used for producing so called word sketches, lists of cooccurrences grouped by grammatical relations, are defined by regular expressions over part-of-speech tags. That approach has its limits, though, and Ivanova et al. (2008) find that for German the shallow approach is inferior to richer parsing strategies. That is why the SketchEngine also uses dependency parsing for languages with less fixed word order, e. g. for Turkish (Ambati et al., 2012).

Phrase structure parsers typically output a bracketed version of the input text that represents the hierarchy of constituents found in the text. Blaheta and Johnson (2001) use the BLLIP corpus (Charniak et al., 2000) that is annotated in Penn-Treebank style to extract multi-word verbs. There are a

⁶ Although, as Bartsch and Evert (2014: 60) note, “the practical benefit of taking this approach depends on the accuracy of the parser and the set of syntactic dependency relations recognized.”

couple of studies that use the statistical grammar by Schulte im Walde (2003) in combination with the LoPar parser (Schmid, 2000): Schulte im Walde (2003) extracts collocations for German verbs and nouns, Zinsmeister and Heid (2003) extract adjective-noun-verb combinations and Zinsmeister and Heid (2004) do a study of noun-verb collocations and find differences in the collocational behavior of compositional and lexicalized compound nouns. Villada Moirón (2005) uses the Alpino parser (Bouma et al., 2001) but does not rely on its dependency output. Instead, she only uses the phrasal boundaries proposed by the parser to identify support verb constructions in Dutch. Seretan (2008; cf. also Nerima et al. (2003) and Seretan and Wehrli (2006)) uses Fips (Wehrli, 2007), a multilingual parser based on “a free-adaptation of Chomsky’s generative linguistics” (Wehrli, 2007: 120), to extract both two-word and multiword cooccurrences (cf. Section 2.2.3). Her extraction process is based on grammatical relations. Some of those relations are explicit in the parser output,⁷ others have to be inferred from the constituent structure. Sangati and van Cranenburgh (2015) use existing treebanks for French, Dutch and English to identify multiword expressions of arbitrary length based on tree fragments extracted with a tree kernel.

In their output, dependency parsers establish grammatical relations between individual words. Therefore, they lend themselves particularly well to an analysis of relational cooccurrences. Teufel and Grefenstette (1995) use the low-level dependency parser described by Grefenstette (1994: 33–46) to identify support verbs for nominalizations. Lin (1998, 1999) extracts collocations using Minipar, a descendant of Principar (Lin, 1994). Pearce (2001) utilizes WordNet (Miller et al., 1990) and a subset of the British National Corpus annotated with the parser by Carroll et al. (1998) to check if a base word is associated with one specific item from a synset but not with the others. Lü and Zhou (2004) propose a model for translating between English and Chinese collocations that they extract from monolingual corpora parsed with the NLPWin parser (Heidorn, 2000). Heid et al. (2008) extract German juridical terminology and use FSPAR (Schiehlen, 2003) to extract verb-object pairs. Weller and Heid (2010) use the same parser to extract German multiword expressions and their morphosyntactic features. Uhrig and Proisl (2012) use the British National Corpus, parsed with the Stanford Parser (Klein and Manning, 2003), to evaluate the lexicographical quality of relational and distance-based cooccurrences using different association measures. Bartsch

⁷ “The candidates of type subject-verb, verb-object, and verb-argument are easily identified from the predicate-argument tables built by Fips, even if they involve long-distance dependencies.” (Seretan, 2008: 89)

and Evert (2014), in their evaluation of different approaches to collocation extraction, find that the C&C parser (Clark and Curran, 2004) outperforms the MaltParser (Nivre et al., 2006) when it comes to extracting collocation candidates. Evert et al. (2017) perform a large-scale evaluation study covering multiple corpora, context sizes (including syntactic dependencies from the C&C parser), frequency thresholds and association measures. Uhrig et al. (2018) evaluate the impact of different parsers and dependency annotation schemes on the extraction quality of six different types of collocations.

In the formal model for relational cooccurrences, each instance of a grammatical relation between words A and B constitutes a cooccurrence of A and B . We can even restrict the analysis to a specific type of relation, e.g. to direct object relations. In that case, only direct object relations between A and B count as cooccurrences of A and B . A cooccurrence is an ordered pair and therefore $\langle A, B \rangle$ is different from $\langle B, A \rangle$, i. e. we assume that grammatical relations are directed and that “ B is the direct object of A ” is not the same as “ A is the direct object of B ”. In the following, we call the word where a relation originates its governor and the word where it points to its dependent. To populate the contingency table, we have to classify all relations that establish cooccurrences, i. e. all grammatical relations in the corpus if we do not have any restrictions or all direct object relations if we are only interested in cooccurrences of verbs and direct objects. For a cooccurrence of A and B where A governs B , we classify the relations according to the following two criteria: “Is A the governor of the relation?” and “Is B the dependent of the relation?”. Once we have classified all relations, cell O_{11} of the contingency table contains the number of relations that constitute cooccurrences of A and B , i. e. the number of relations where A governs B . Cell O_{12} contains the number of relations where A governs words other than B and cell O_{21} contains the number of relations where B is governed by words other than A . O_{22} contains the number of relations with governors other than A and dependents other than B .

For the fully worked out formal model, cf. Evert (2004: 33–37).

Covarying collexeme analysis

Methodologically, covarying collexeme analysis (Gries and Stefanowitsch, 2004a; Stefanowitsch and Gries, 2005) can be seen as a minor extension of relational cooccurrences. Instead of analyzing pairs of words connected by a grammatical relation, it investigates “the association between pairs of words occurring in two different slots in the same construction [...] – for example, the verb and the direct object in the ditransitive construction.” (Stefanowitsch

and Gries, 2009: 942) This means that the main difference between relational cooccurrences and covarying collexeme analysis is a minor extension of the notion of cooccurrence: Instead of via a single grammatical relation, the two cooccurring words can be related via a more complex grammatical structure. Apart from that, the statistical analysis is exactly the same: We have to classify all instances of the linguistic structure according to whether word *A* occurs in the one slot and word *B* in the other slot of the construction.

2.2 Cooccurrences involving grammatical structures

In this section, we discuss approaches to cooccurrences that go beyond pairs of words and involve some kind of grammatical structures. Our focus is on works where the cooccurrences with grammatical structures are subject to some kind of statistical analysis. We do not consider approaches where grammatical structures, for example morphosyntactic patterns (e. g. Ramisch, 2015) or subtrees from dependency trees (e. g. Martens and Vandeghinste, 2010), merely serve as a means for extracting cooccurring words and do not play a role in the subsequent analysis.

2.2.1 Simple collexeme analysis and distinctive collexeme analysis

There are two varieties of collocation analysis that investigate the association between words and grammatical structures: Simple collexeme analysis (Stefanowitsch and Gries, 2003) and distinctive collexeme analysis (Gries and Stefanowitsch, 2004b). Distinctive collexeme analysis is used to find out if the word under investigation has a statistically significant preference for one of a set of alternating constructions:

[D]istinctive collexeme analysis is used in investigating the association between a word and (one member of) two or more semantically or functionally equivalent constructions [...] – for example, between the verb *give* and the ditransitive construction as opposed to the prepositional dative;

(Stefanowitsch and Gries, 2009: 941)

In the case of two alternating constructions, the frequencies necessary for a statistical analysis can be represented in a 2×2 contingency table. To populate the table, we have to classify all instances of both alternating constructions. Instances of the first construction are classified in the first row of the table, instances of the second in the second row. For every instance, we have to decide if the word under investigation occurs in the relevant slot

of the construction. By applying a statistical test, e. g. Fisher's exact test (cf. Section 3.3.1) or an approximation to it (cf. Section 3.3.2), we can determine if the word under investigation has a preference for one of the constructions. If yes, we have to compare the actual cooccurrence frequencies to the expected frequencies (cf. Section 3.2) in order to find out which alternating construction is preferred.

Simple collexeme analysis is used to determine the association strength between a word and a construction:

[Simple] collexeme analysis is used in investigating the association between a construction and the words occurring in a particular slot in this construction [...] – for example, between the verb *give* and the ditransitive construction as opposed to all other constructions;

(Stefanowitsch and Gries, 2009: 941)

Analogously to distinctive collexeme analysis where instances of the first construction are classified in the first row and instances of the second, alternating construction are classified in the second row, the contingency table for simple collexeme analysis classifies instances of the construction under analysis in the first row and instances of “all other constructions in the corpus” (Stefanowitsch and Gries, 2009: 942) in the second:

[F]or [simple] collexeme analysis, A corresponds to a given construction, $\neg A$ corresponds to all other constructions in the corpus, B corresponds to a given word (lemma) occurring in a particular slot in A, and $\neg B$ corresponds to all other words in [sic] occurring in the corpus;

(Stefanowitsch and Gries, 2009: 942)

While the above quote sounds reasonably clear, it is not a formal model and this leads to the problems discussed below.

Problems with simple collexeme analysis⁸

For populating a contingency table, we have to cross-classify the units of analysis (Section 3.2). For simple collexeme analysis, the units of analysis seem to be all constructions in the corpus:

⁸ Note that this section focuses exclusively on problems surrounding the units of analysis. For a further discussion of simple collexeme analysis cf. Bybee (2010), Gries (2012), Schmid and Küchenhoff (2013), Gries (2015), Küchenhoff and Schmid (2015).

Thus, to calculate the collocation strength of a given collexeme L for a given construction C , we need four frequencies: the frequency of L in C , the frequency of L in all other constructions, the frequency of C with lexemes other than L and the frequency of all other constructions with lexemes other than L .

(Stefanowitsch and Gries, 2003: 218)

Only two of those four frequencies have a clear interpretation: O_{11} , the number of instances of the construction C that have the word L in the relevant slot, and O_{12} , all other instances of the construction C . The obvious problem with the remaining frequencies is that the total number of constructions is unknown:

Unfortunately, there is some uncertainty about the fourth factor mentioned above – the number of constructions that occur in the corpus. There is no known way to count the number of constructions in a corpus because a given clause may instantiate multiple constructions.

(Bybee, 2010: 98)

Not only do we not know how many constructions there are, we also do not know in how many constructions a given word occurs. Therefore, we cannot infer the value of O_{21} by subtracting O_{11} from the frequency of L – several instances of L could occur in a single construction or a single instance of L could be part of several constructions.

The two frequencies with a clear interpretation are those that classify the instances of C into those that have L in a particular slot and those that do not. But what exactly constitutes an instance of a linguistic structure? Let us assume we are looking for nouns modified by two adjectives and come across a sentence like the one in Example 2 (emphasis added). Are there 10, 5, 3 or 1 instances of the structure “noun with two adjectival modifiers” in this sentence? (For a detailed analysis of the counting methods that lead to those frequencies cf. Section 4.2.)

(2)_{BNC: A0U 882} [...] he had his *own dirty little war* on *distant barren islands*, his *own vicious murders* on the Rock.

Given those problems, how do we proceed in practice? In the case of the [N *waiting to happen*] construction, Stefanowitsch and Gries say: “[T]he total number of constructions was arrived at by counting the total number of verb tags in the BNC, as we are dealing with a clause-level construction centering around the verb *wait*.” (Stefanowitsch and Gries, 2003: 218) The implicit

	accident	¬accident	Row totals
[N <i>waiting to happen</i>]	14	21	35
¬[N <i>waiting to happen</i>]	8606	10197659	10206265
Column totals	8620	10197680	10206300

Table 2.1: Contingency table for *accident waiting to happen* taken from Stefanowitsch and Gries (2003: 219)

assumption behind this approximation is that it is not possible for a verb to be part of more than one construction.

The practical implication of the approximation is that we are now classifying verbs (cf. the contingency table for *accident waiting to happen* in Table 2.1). For each verb we have to decide if it is an instance of [N *waiting to happen*]. We also have to decide if the construction, i. e. the verb, cooccurs with *accident*. For instances of [N *waiting to happen*], this means we have to check if the N slot is filled by *accident* or by another noun. But what about the remaining verbs? While most verbs occur in grammatical structures that involve nouns, some have more than one noun slot and others have none and it is not at all clear what we should do. In sentences like “She cried.” or “Run!” there is no noun slot in which *accident* could occur. Should we ignore those verbs and only use verbs that are part of a construction involving nouns? If yes, how can we delimit the boundaries between constructions when there are several verbs in a sentence? Or should we just classify the verbs according to whether they cooccur with *accident* in the same sentence? If yes, we should really be using the segment-based cooccurrence model (Section 2.1.1) and be classifying sentences instead of verbs.

The latter option, using a segment-based approach and classifying sentences, is used by Stefanowitsch and Gries in their three-dimensional extension of covarying collexeme analysis where they “calculate the association strength of the elements of a trigram consisting of two collexemes and a construction” (Stefanowitsch and Gries, 2005: 23).

Another approach is taken in an R script for collostructional analysis by Stefan Gries.⁹ The script asks the user to “[e]nter the size of the corpus (in constructions or words)” and uses that as the sample size. The user is also asked to provide a table that includes the frequency of the word in the corpus and the frequency of the word in the construction. If the sample size is the size of the corpus in words, then this amounts to classifying all words in the

⁹ <http://www.linguistics.ucsb.edu/faculty/stgries/teaching/groningen/coll.analysis.r>

corpus. In the case of *accident waiting to happen*, we would have to classify all words in the corpus according to the following two criteria: “Does the word occur in the [N *waiting to happen*] construction?” and “Is it the word *accident*?”. Since we are only interested in words occurring in the N slot of the [N *waiting to happen*] construction and since, by definition, only nouns can occur in that slot, it makes sense to restrict the analysis to nouns (just like it makes sense for a study on erectile dysfunction to include only male test subjects).¹⁰ This means we have to classify all nouns in the corpus according to the following two criteria: “Does the noun fill the N slot of the [N *waiting to happen*] construction?” and “Is it the noun *accident*?”. This is what is suggested in the model developed in this work. Coincidentally, this is also what is done (for verbs) in a lot of case studies on simple collexeme analysis, specifically those where the sample size is approximated by the number of verbs and where the collexeme slot is also a verb, e. g. in the analysis of the *into*-causative (Stefanowitsch and Gries, 2003: 224–227) or the ditransitive (Stefanowitsch and Gries, 2003: 227–230).

In any case, the problems discussed in this section show how important it is to have an unambiguous formal model that specifies, among other things, what the units of analysis are, what the sample size is, how instances are counted and what constitutes a cooccurrence.

2.2.2 N-gram-based approaches

N-grams, sequences of n words, are among “the oldest and most broadly useful practical tools in language processing.” (Jurafsky and Martin, 2009: 153) However, n-grams only capture exact repetitions of a sequence of adjacent words, missing out on patterns involving slots with high variability. Therefore, many proposals for extensions have been made. Those proposals include, on the one hand, approaches that, like traditional n-grams, are based on word forms only but that go beyond sequences of adjacent words. Skipgrams (Guthrie et al., 2006), for example, allow gaps and ConcGrams (Cheng et al., 2006, 2009) allow both gaps and positional variants. On the other hand, there are approaches that stick with sequences of adjacent items but that abstract away from word forms and include, for example, part-of-speech tags or placeholders for phrases. Our focus in this section is on the latter type of n-grams, i. e. those that involve some notion of “grammar”.

The Xtract system presented by Smadja (1993) not only extracts “predicative relations”, i. e. cooccurrences of pairs of syntactically related words, and “rigid

¹⁰ Cf. Section 5.1.2 for how an unnecessary inflation of O_{22} , e. g. by including items that cannot possibly fulfill either classification criterion, impacts different association measures.

noun phrases”, i. e. n-grams of arbitrary length, but also “phrasal templates”, i. e. n-grams consisting of both words and part-of-speech tags. The phrasal templates are constructed in a simple way: A word or a part-of-speech tag is kept in a certain slot of the final n-gram if it occurs there with a probability greater than some threshold (Smadja, 1993: 158).

The GRASP system (Chen et al., 2010) considers all instances found by a query, e. g. “play role”, regardless of the order or adjacency of the query terms and identifies the most frequent part-of-speech n-grams to the left, to the right or between the query terms. The extracted n-grams are ranked by their frequencies.

The aim of both Xtract and GRASP is to extract “interesting” patterns. While those patterns abstract away from sequences of words, there is no statistical analysis taking place that assesses, for example, the association between the lexical and the more abstract parts of the patterns.

The approach to the identification of multiword expressions presented by Sangati and van Cranenburgh (2015) extracts fragments of phrase structure trees by using a tree kernel. The so-called “frontier nodes” of the tree fragments can be lexical or non-lexical, i. e. word forms or phrases. For ranking the tree fragments, they suggest two different approaches. The first is to treat the tree fragments as n-grams and to apply established association measures (pointwise mutual information and log-likelihood, cf. Section 3.3.2). In the second, they suggest a new association measure based on the probabilities with which a Probabilistic Tree Substitution Grammar (PTSG) (Bod, 1992; Bod et al., 2003) can generate a fragment either in a single step or by combining smaller fragments together. In both cases, the results are presented as n-grams consisting of words and phrase markers.

Similar to the approaches mentioned above, StringNet¹¹ (Tsao and Wible, 2009; Wible and Tsao, 2010; Tsao and Wible, 2013) presents the user with lexico-grammatical patterns represented as so called hybrid n-grams that can contain word forms, lemmata, part-of-speech tags and broader notions of word classes. The major feature that sets StringNet apart is that it is organized and can be navigated as a network. The hybrid n-grams are linked by subsumption relations (cf. Section 4.1.6) that are labeled as parent/child relations in the user interface. For a given n-gram, the user can navigate to more general or more specific hybrid n-grams that are ordered by association strength.¹²

¹¹ <http://nav.stringnet.org/>

¹² A similar idea of a navigable network of statistically associated patterns has been sketched by Proisl (2012).

The extraction process considers all hybrid n -grams that include at least one lexical item (a word form or a lemma) and that occur at least five times in the BNC. For practical purposes, the total number of hybrid n -grams has to be reduced, therefore two pruning steps are applied: Vertical pruning and horizontal pruning.

The vertical pruning step, also called subset pruning (Tsao and Wible, 2009), compares all hybrid n -grams of the same length. For a threshold ϵ with $0 < \epsilon < 1$ and two hybrid n -grams x and y with $\text{len}(x) = \text{len}(y)$ and $x \subset y$ (instances of x are a subset of the instances of y , i. e. y is more general than x), the more general hybrid n -gram y will be pruned if $\frac{|x|}{|y|} \geq \epsilon$. For example, if we set $\epsilon = 0.8$, then the more general hybrid n -gram y will be pruned if there is a more specific hybrid n -gram x such that at least 80% of the instances of the more general hybrid n -gram y are also instances of the more specific hybrid n -gram x .

The horizontal pruning step considers hybrid n -grams of different lengths. Given two hybrid n -grams that only differ in the value of n , i. e. their length (which means that the shorter one is part of longer one), the shorter hybrid n -gram is the more general one and is pruned if more than 80% of its instances are also instances of the longer and more specific hybrid n -gram.

The hybrid n -grams that are retrieved for a query, e. g. the larger hybrid n -grams in which a smaller one occurs, are ranked by “normalized MI”, a heuristic association measure that normalizes the pointwise mutual information of the associated n -grams ($\text{PMI} = \log \frac{P(x_1, \dots, x_n)}{\prod_{i=1}^n P(x_i)}$) by the maximal PMI value for a hybrid n -gram of the same length in the result set of the query. This means that normalized MI does not really measure the association between the query and the related hybrid n -grams. Instead, it merely rescales the PMI scores of the related n -grams to make them comparable across different values of n , i. e. different lengths (Wible and Tsao, 2010: 28).

Forsberg et al. (2014) extract hybrid n -grams quite similar to those in StringNet as construction candidates for inclusion into SweCcn, the Swedish construction.¹³ The major differences are that Forsberg et al. (2014) also include phrase markers in their hybrid n -grams and that they are only interested in a list of construction candidates and not in the network of their interactions.

In the extraction process, they only keep hybrid n -grams that start and end at phrase boundaries. They rank the resulting hybrid n -grams by a heuristic association measure that scales the pointwise mutual information (PMI)

¹³ <https://spraakbanken.gu.se/eng/sweccn>

score of the n -gram “by the unique instance frequency (UIF), the number of unique word sequences matched by the pattern in the corpus” (Forsberg et al., 2014: 122):

$$\text{UIF-PMI}(x_1, \dots, x_n) = \text{UIF}(x_1, \dots, x_n) \log \frac{P(x_1, \dots, x_n)}{\prod_{i=1}^n P(x_i)}$$

Additionally, they apply a pruning strategy similar to Wible and Tsao’s (2010) vertical pruning: “[I]f a more general pattern (e. g. **[PREPOSITION] [NP]**) is ranked higher than a more specific pattern (e. g. **in [NP]**), then the specific pattern is removed.” (Forsberg et al., 2014: 122)

2.2.3 Multiword collocations based on cooccurrences of syntactic structures

An interesting property of multiword collocations is that they can be recursive:

An additional problem of the interaction between syntactic and collocational description is the recursive nature of collocational properties: the components of a collocation can again be collocational themselves: next to the German collocation *Gültigkeit haben* ($n+v$), we have *allgemeine Gültigkeit haben*, with *allgemeine Gültigkeit*, a collocation ($n+a$), as a component.

Heid (1994: 231)

This observation serves as an inspiration for an approach to the analysis of multiword collocations that models cooccurrences of more than two words as cooccurrences of two larger units that are treated as single items. This approach not only captures the recursive nature of collocations but also has the advantage of being able to rely on 2×2 contingency tables.

As mentioned in Section 2.1.2, Seretan (2008) uses the Fips parser (Wehrli, 2007) to extract two-word cooccurrences based on grammatical relations. The extracted two-word cooccurrences include information on the syntactic type of the combination, i. e. the grammatical relation between the two words, and are used to build up larger multiword cooccurrences (Seretan et al., 2003, 2004; Seretan, 2008: 131–141). In this approach, two n -word combinations (called n -grams, despite the fact that the words need not be adjacent) can be combined into an $n + 1$ -word combination if they have $n - 1$ words in common. And of course, this can only happen if there are instances of the two n -word combinations where the two actually cooccur, i. e. where the $n - 1$ words they have in common are instantiated by the same tokens in the corpus.

Since the n -word combinations retain information about the grammatical relations between the words, this amounts to cooccurrences of syntactic structures. Seen from this perspective, Heid's example *allgemeine Gültigkeit haben* is the cooccurrence of two syntactic structures: Verb + direct object (*Gültigkeit haben*) and noun + adjectival modifier (*allgemeine Gültigkeit*). There are, however, some noteworthy restrictions: Both syntactic structures have to be fully lexicalized, must both consist of n words and must overlap in $n - 1$ words.

Seretan et al. (2003, 2004) propose four ranking methods for the extracted multiword collocations: i) The frequency of the multiword collocation, ii) the sum of the log-likelihood scores assigned to the two-word combinations that make up the larger multiword combination, i. e. $\sum_{i=1}^n \text{score}(\text{MWC}_i)$, iii) a heuristic combination of the log-likelihood scores of the two-word combinations that resembles the harmonic mean, i. e. $\frac{n \prod_{i=1}^n \text{score}(\text{MWC}_i)}{\sum_{i=1}^n \text{score}(\text{MWC}_i)}$, and iv) a statistical analysis of the association between the two n -word combinations that make up the $n + 1$ -word combination that treats the two n -word combinations as single items and computes the log-likelihood score based on a 2×2 contingency table.

There is, however, no formal model for the latter method and it is, therefore, unclear what, for example, the sample size for larger multiword combinations should be. This question is either not addressed at all (Seretan et al., 2004: 96), or it is only vaguely specified:

[The log-likelihood test] applies to term pairs to whom it assigns a collocativity score computed according to the contingency table of the pair (which contains the frequency of: i) the cooccurrence of the two terms together in the corpus, ii) the co-occurrence of one of the terms with a different one, and iii) all the other cooccurrences, not involving any of the terms in the given pair).

(Seretan et al., 2003: 428)

It is not clear from that quote, what the units of analysis should be. Should we use all instances of syntactically related $n + 1$ -word combinations in the corpus? Should we use the subset of syntactically related $n + 1$ -word combinations extracted by the method described above? In any case, by classifying $n + 1$ -word combinations, we ignore instances of the n -word combinations that do not occur in a larger combination. And should we really classify all $n + 1$ word combinations, even those that, due to their different syntactic structure, cannot possibly be instances of either of the two n -word combinations? Would this not be an unnecessary inflation of the sample size (cf. Sections 4.4.4 and 5.1 for a discussion of the effects this has on various

association measures)? How should we deal with overlapping $n + 1$ -word combinations? The independence assumption made by statistical tests is clearly violated by having overlapping units of analysis. In addition, since we are dealing with syntactic structures, we have to cope with the problems already hinted at in Section 2.2.1 (although, due to the fully lexicalized structures used in this approach, we can expect those problems to play a lesser role): How do we count instances of syntactic structures (cf. Section 4.2) and how do we cross-classify them (cf. Section 4.4)?

2.2.4 Valency pattern extraction

The notion of valency patterns originates from valency theory, a model of language developed out of Tesnière’s (1959/2015) dependency grammar. The fundamental assumption of valency theory is that valency carriers (especially verbs, but also adjectives, nouns or particles) largely determine the structure of a sentence.¹⁴ A distinction central to valency theory is that between complements, i. e. elements that are “determined in [their] form by the word functioning as the head of the clause or phrase or/and that [fill] one of its obligatory valency slots” (Herbst and Schüller, 2008: 207), and adjuncts, i. e. elements that are “independent of the valency of the governing verb” (Herbst and Schüller, 2008: 207). A valency pattern is a (typically sequential) representation of the valency carrier and its complements and can be seen as a blueprint for which syntactic arguments the valency carrier takes.

As an example, consider the active patterns for the verb *cure* in the “cure people” sense from the Erlangen Valency Patternbank:¹⁵

NP + VHC_{act}
 NP + VHC_{act} + NP
 NP + VHC_{act} + NP + by_V-ing
 NP + VHC_{act} + NP + of_NP
 NP + VHC_{act} + NP + of_V-ing
 NP + VHC_{act} + NP + with_NP

The patterns consist of the valency carrier (verbal head complex, active voice) preceded by a noun phrase (the subject) and followed by the remaining complements. In traditional dictionaries, valency patterns such as “NP + VHC_{act} + NP” or “NP + VHC_{act} + NP + of_NP” are usually paraphrased as “cure something” or “cure somebody of something”.

¹⁴ For an overview of dependency and valency approaches to English grammar, cf. Herbst (forthcoming 2019). For more detailed information on dependency and valency, cf. Ágel et al. (2003, 2006) and Gerdes et al. (2014).

¹⁵ <http://patternbank.fau.de>

Generally, a valency carrier is always used with one of its valency patterns, i. e. an occurrence of a valency carrier is usually a cooccurrence of this valency carrier with a syntactic structure, the complements that make up the valency pattern. This means that techniques similar to simple collexeme analysis can be applied to determine the degree of association between words and valency patterns. In practice, this can be used for extracting valency patterns – or subcategorization frames, as they are referred to in the Chomskyan linguistic tradition – from (usually automatically parsed) corpus data. In the remainder of this section, we highlight some of the approaches to the extraction or acquisition of valency patterns that have a strong focus on the statistical analysis of that association.

Those approaches fall into two categories: Approaches that use a fixed, predetermined inventory of valency patterns and approaches that try to infer the valency patterns from the data. For the former approaches, acquiring the valency patterns associated with a valency carrier is equivalent to assigning every instance of the valency carrier to one of the valency patterns in the inventory, i. e. we merely have to decide which valency pattern it instantiates. The disadvantage of this approach is that it requires an a priori specification of the inventory of valency patterns. The latter approaches do not need such a predefined inventory of valency patterns since they try to infer the valency patterns from the data. However, those patterns are not guaranteed to be “proper” valency patterns. While most systems filter the constituents that they take into account by their form and syntactic function, they usually do not try to make a distinction between complements and adjuncts that goes beyond the application of some statistical analysis of the patterns.

The statistical analyses that are commonly applied, usually to serve as a kind of noise filter, are the same for both kinds of approaches.

Binomial hypothesis test

Binomial hypothesis testing (cf. Section 3.3.1) for valency pattern extraction has been popularized by Brent (1993). The usual approach is to have the null hypothesis that a given valency carrier does not have a given valency pattern and to assume that valency carriers can, e. g. because of parsing errors, nevertheless cooccur with a probability of π with valency patterns they do not have. Now, if there are n instances of the valency carrier in the corpus, then

we can calculate the probability with which m or more of those instances cooccur with the given valency pattern:

$$P(m+, n, \pi) = \sum_{i=m}^n \binom{n}{m} \pi^m (1 - \pi)^{n-m}$$

If m of the n instances of the valency carrier cooccur with the valency pattern and if $P(m+, n, \pi)$ is small, typically below 0.05, then we can reject the null hypothesis and conclude that the valency carrier very likely has the given valency pattern. This approach uses information (the probability π) that cannot be derived from the usual 2×2 contingency table and has to be estimated by other means (cf. Brent (1993) and Briscoe and Carroll (1997) for descriptions of two popular methods).

Binomial hypothesis testing has been used, at least as one option among others, in a lot of studies, for example by Manning (1993), Ersan and Charniak (1996), Briscoe and Carroll (1997), Lapata (1999), Sarkar and Zeman (2000), Korhonen (2002), Han et al. (2004), Li et al. (2005), Korhonen et al. (2006) and Chesley and Salmon-Alt (2006).

Log-likelihood ratio test

Sarkar and Zeman (2000) were the first to use the log-likelihood ratio test (cf. Section 3.3.2) for extracting verbal valency patterns. Subsequently, it has also been used by Korhonen (2002) and Korhonen et al. (2006). The cooccurrence data for the analysis can be represented in a contingency table quite similar to the one in Table 2.1. The sample size corresponds to the number of verbs in the corpus and C_1 is the number of instances of the verb under investigation, C_2 is the number of instances of other verbs, R_1 is the number of verb instances cooccurring with the given valency pattern and R_2 is the number of verb instances not cooccurring with the given valency pattern. The log-likelihood ratio is computed as follows:

$$\text{log-likelihood} = 2 \sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}}$$

If the test statistic exceeds some threshold (typically determined on the basis of the χ^2 -distribution with 1 degree of freedom), then the null hypothesis that the verb under investigation and the valency pattern cooccur only by chance is rejected and it is concluded that the verb very likely has the given valency pattern.

This approach is very similar in spirit to simple collexeme analysis (Section 2.2.1) but, since it is only concerned with verbal valency patterns, it is much clearer about what the units of analysis are. In fact, if simple collexeme analysis is implemented as suggested in this work, the statistical analysis is the same. As such, this approach to the statistical analysis of valency patterns is a special case of the general cooccurrence model developed in the current work.

t-test

The scores of the one-sample *t*-test are a popular, albeit problematic, association measure for two-word cooccurrences (cf. Section 3.3.3). Sarkar and Zeman (2000) use the scores of a two-sample *t*-test, Welch's unequal variances *t*-test, for valency pattern extraction. A two-sample *t*-test is used to test if two populations have equal means. In this case, for the extraction of verbal valency patterns, the two populations are I) the instances of the verb under investigation and II) the instances of other verbs; the means are the relative frequencies of a given valency pattern within those populations. Let us assume that the frequencies derived from the sample data are organized in a contingency table like the one in the previous section, i. e. such that C_1 corresponds to the number of instances of the verb under investigation and C_2 to the number of instances of other verbs. Then, Welch's *t*-test is computed as follows, with $\sigma_1^2 = \frac{O_{11}}{C_1}(1 - \frac{O_{11}}{C_1})$ and $\sigma_2^2 = \frac{O_{12}}{C_2}(1 - \frac{O_{12}}{C_2})$ being the variances of the underlying Bernoulli distributions:

$$t\text{-score} = \frac{\frac{O_{11}}{C_1} - \frac{O_{12}}{C_2}}{\frac{\sigma_1^2}{C_1} + \frac{\sigma_2^2}{C_2}}$$

If the *t*-score exceeds some threshold (typically determined on the basis of the *t*-distribution with the appropriate degrees of freedom), then the null hypothesis that the two means are equal is rejected and it is concluded that the verb under investigation very likely has the given valency pattern.

It should be noted that Welch's *t*-test assumes that the two populations have normal distributions. This assumption is not met for cooccurrence frequency data with their binary indicator variables (an instance of a verb does either cooccur with a given valency pattern or it does not). Therefore, from a theoretical perspective, the *t*-test cannot be applied to this kind of data (cf. also Evert, 2004: 82–83). Additionally, it is assumed that the sample data have been drawn independently from the two populations, which is arguably not the case in a typical corpus linguistic setting.

Nevertheless, the t -test has been used by Sarkar and Zeman (2000), Korhonen (2002), Korhonen et al. (2006) and Lenci et al. (2008).

Maximum likelihood estimates

Another popular option is to do without a statistical hypothesis test and simply use a threshold on the relative frequency with which a given valency pattern p_i cooccurs with a given valency carrier c_j , i. e. to use the maximum likelihood estimate of the probability $P(p_i|c_j)$. This approach is taken, often as one option among others, for example by Lapata (1999), Korhonen (2002), Han et al. (2004), Korhonen et al. (2006), Preiss et al. (2007), Messiant et al. (2008) and Lenci et al. (2012).

Others

Yoshinaga (2004) and Yoshinaga and Tsujii (2004) estimate confidence values for how strong the evidence is that a given word has a given valency pattern. They work in the Bayesian framework and compute the confidence values by integrating over the a posteriori distribution $P(\theta_{ij}|D)$, where θ_{ij} is the probability of valency carrier c_i occurring with valency pattern p_j , i. e. $\theta_{ij} = P(p_j|c_i)$, and D is the observed data.

Lenci et al. (2012) find local mutual information, i. e. the cooccurrence frequency of valency carrier and valency pattern weighted with the mutual information between the two, “particularly useful for the identification of the most prototypical [subcategorization frames] for each predicate” (Lenci et al., 2012: 3714):

$$\text{local-MI} = O_{11} \cdot \log \frac{O_{11}}{E_{11}}$$

2.3 Summary

The traditional approaches to the statistical analysis of cooccurrences of pairs of words fall into three classes: I) Segment-based cooccurrences, where the words cooccur within some segment of text, II) distance-based cooccurrences, where the words cooccur within some orthographically defined distance, and III) relational cooccurrences, where the cooccurring words are related via some grammatical relation (covarying collexeme analysis can be seen as a variant of relational cooccurrences where the two words are related via a more complex grammatical structure). All three are well-understood and there are fully worked out statistical models for their interpretation.

This is not the case for the analysis of cooccurrences involving some kind of grammatical structure, e. g. cooccurrences of a word with a syntactic structure or of a pair of syntactic structures. While there are several approaches with different aims, the cooccurrence models and their interpretation are generally not formalized to the same degree as for two-word cooccurrences.

There are two variants of collostructional analysis that involve grammatical structures: Simple collexeme analysis is a method for analyzing cooccurrences of words with constructions and distinctive collexeme analysis is a method for analyzing the preference of a word for one of a set of alternating constructions.

There is a variety of n -gram-based approaches that involve “grammatical structure” in the form of part-of-speech tags or phrases but only StringNet deals with cooccurrences of larger linguistic structures. However, there is no statistical analysis of the cooccurrence of the larger structures but only of the item-level cooccurrences that make up the hybrid n -grams.

There is an approach to the analysis of multiword collocations that analyzes a collocation of $n + 1$ words as a cooccurrence of two n -word collocations. Since the analysis is based on a relational cooccurrence model, this amounts to the cooccurrence of two fully lexicalized syntactic structures, each with n words.

Another special type of cooccurrences, that of valency carriers with valency patterns, is often statistically analyzed in order to extract the valency patterns of a valency carrier from corpus data.

Out of the approaches involving grammatical structures, simple collexeme analysis, the use of the log-likelihood ratio test in valency pattern extraction and the analysis of syntactically related multiword collocations are the ones most closely related to the current work. Those approaches, however, are quite limited since they are either restricted to cooccurrences of a single word with a syntactic structure or to cooccurrences of completely lexicalized syntactic structures that both contain n words and overlap in $n - 1$ words. They also lack clear formal models.

3 Statistical analysis of cooccurrence data

The main goal of statistical analysis in the study of language is to make statements about certain properties of a language or language variety. Since it is infeasible to analyze or even collect everything that has been said or written by all the members of a speech community, claims about a language are necessarily based on only a tiny sample of all that material. Statistical analysis is used both to quantify the strength of effects found in a sample and to generalize findings beyond the sample.

In this chapter, we describe the typical approach to analyzing linguistic cooccurrence data using frequentist statistics. While categorical data can also be analyzed in the alternative framework of Bayesian statistics (cf. Agresti and Hitchcock, 2005 for a survey), the frequentist framework is clearly predominant in the analysis of the kind of linguistic cooccurrence data we are interested in. And while that approach does have some problems (cf. Section 3.4), sticking to it and using strictly formalized contingency tables comes with the advantage of being able to apply a wealth of established analytical methods (cf. Agresti, 2013 for a thorough presentation) – and any progress made in overcoming those problems could be directly transferred to the current work.

After a brief sketch of the basic concepts of inferential statistics (Section 3.1), we introduce contingency tables as a means for representing cooccurrence data (Section 3.2) and a selection of association measures for analyzing them (Section 3.3). Section 3.4 discusses some of the criticism that has been leveled at the approach and some suggestions for extending it.

3.1 Inferential statistics

The goal of inferential statistics is to confront a statistical model with data in order to estimate the parameters of the model from the data and to test hypotheses about the model against the data. In short, we want to draw inferences about the population from the examination of a sample.

This section gives a rather high-level introduction to the topic. For a more detailed presentation tailored to linguists' needs, see, for example, Baroni and Evert (2009).

3.1.1 Statistical models

In statistical terminology, a population is any well-defined “collection of statistical units of the same nature whose quantifiable information we are interested in” (Dodge, 2008: 428). In corpus linguistics, the population usually consists of all the utterances that have ever been or can be produced in a language or sublanguage.

In order to formally describe the population, a statistical model of the population is postulated which can either be parametric or nonparametric. Parametric models make assumptions about the distribution of the population, e. g. assuming normally distributed data. The parameters of the model characterize quantitative aspects of the population, e. g. its mean and standard deviation. Nonparametric models on the other hand make few or no assumptions about the distribution of the population. “[T]he model structure is not specified a priori but is instead determined from [sample] data.” (Dodge, 2008: 375) Note that nonparametric does not mean without parameters. In order to formulate a statistical model, we do have to postulate some population parameters (e. g. occurrence probabilities, cf. Section 3.2) – but we do not have to assume a special distribution of the population data. The standard approach to the analysis of cooccurrence data is based on nonparametric models (cf. Section 3.2).¹

The postulated model can be seen as a stochastic process that is able to generate any and all members of the population. Thus, we can regard the observed data in a sample as a random outcome of the underlying model; therefore, if samples are randomly drawn from the population, we would not be surprised to make slightly different observations in another randomly drawn sample from the same population. That corresponds to our intuitive expectation that for example the relative frequencies of a word in two different corpora of the same language variety should be quite similar but not necessarily identical although both corpora were created to be representative of the same language variety. Those variations between random samples from the same population are described by the sampling distribution that follows from the statistical model of the population.

¹ That is because parametric models usually require continuous data, i. e. “[d]ata that might take any value or any value within a range” (Boslaugh, 2012: 546), and cannot be applied to cooccurrence data, which are categorical (cf. Section 3.2).

3.1.2 Hypothesis testing

Statistical models can be used to test hypotheses about the population. The hypotheses are formulated in terms of model parameters and are tested on the observations made on a representative random sample drawn from the population. The key question in hypothesis testing is: What is the probability of observing data as extreme as or more extreme than in the sample, given the assumptions we have made about the model and its parameters, i. e. our hypothesis? If that probability is lower than a predefined threshold (usually 0.05), we can be reasonably sure that the hypothesis is false and can reject it.

That is also what is usually done when testing hypotheses. Instead of trying to directly accept a positively formulated hypothesis about some effect (called the alternative hypothesis), the standard approach is to try and reject a contradictory hypothesis of “no effect” (called the null hypothesis). If we want to prove that two linguistic units cooccur more often with each other than expected by chance, our null hypothesis should state that the two units cooccur only by chance. If we can reject the null hypothesis, we have indirectly proven our alternative hypothesis to be true (but cf. Section 3.4.1 for why we should also take the effect size into account).

To do that, a null hypothesis H_0 that we would like to disprove is formulated in terms of model parameters. The model provides a probability framework for the null hypothesis against which the observed data (or a statistic derived from it, e. g. a log-likelihood score) can be compared using the sampling distribution of the data (or the distribution of the statistic derived from it, e. g. a χ^2 -distribution). The values of the model parameters are estimated from the observed data (cf. Section 3.2). Since the observed data are subject to sampling variation, the estimated parameter values are also random outcomes subject to variation.

Using those parameter estimates in the null hypothesis, we can either determine the probability of observing data as extreme as or more extreme than observed directly via the sampling distribution or we can determine the equivalent probability of observing corresponding values of a derived statistic via the distribution of that statistic. That probability is called the *p*-value and common thresholds for rejecting the null hypothesis are *p*-values smaller than 0.05, 0.01 or 0.001. The interpretation of the phrase “as extreme as or more extreme than observed” depends on the alternative hypothesis. If the alternative hypothesis is two-sided (or two-tailed), observed values both smaller and larger than expected under the null hypothesis are taken into account, if the alternative hypothesis is one-sided (or one-tailed), only observed values either smaller or larger than expected are taken into account.

The intuition behind that is that if the observed data deviate substantially from what we expect under the null hypothesis, then the null hypothesis is rather improbable. For cooccurrence data, one-sided alternative hypotheses are usually more adequate, since we are usually only interested in positively associated items (attraction) and less so in negatively associated items (repulsion).

3.2 Contingency tables

Cooccurrence data are typically analyzed with the help of contingency tables. Contingency tables are a device for cross-classifying the members of a population or of a sample according to some variables. These classifications must be exhaustive and mutually exclusive. This type of data is called categorical or qualitative data, i. e. data with observable properties that cannot be measured numerically. Classifications with only two categories are called dichotomous classifications.

In the analysis of cooccurrence data, we are cross-classifying every unit of analysis according to two dichotomous variables which can be represented in a 2×2 contingency table, cf. Table 3.1: Presence or absence of linguistic unit A in the unit of analysis and presence or absence of linguistic unit B in the unit of analysis. The nature of these units of analysis (also called elements or sampling units) depends on the cooccurrence framework used in the analysis. For collocations, the sampling units could for example be dependency relations (relational cooccurrences) or sentences (segment-based cooccurrences). The total number N of elements in the sample is called the sample size.

	$B = b$	$B \neq b$	
$A = a$	O_{11}	O_{12}	R_1
$A \neq a$	O_{21}	O_{22}	R_2
	C_1	C_2	N

Table 3.1: 2×2 contingency table

Every element in the sample is classified according to the two variables. For segment-based cooccurrences that means answering the two questions “Does word A occur in the sentence?” and “Does word B occur in the sentence?”. The fields O_{11} to O_{22} (O stands for observed frequency) contain the corresponding counts. O_{11} contains the number of elements for which both variables

evaluate to true, i. e. the number of sentences in which both word *A* and word *B* occur. O_{12} contains the number of elements for which the first variable but not the second evaluates to true, i. e. the number of sentences in which word *A* occurs but not word *B*. O_{21} contains the number of elements for which the second variable but not the first evaluates to true, i. e. the number of sentences in which word *B* occurs but not word *A*. O_{22} contains the number of elements for which neither variable evaluates to true, i. e. the number of sentences in which neither word *A* nor word *B* occurs.

R_1 , R_2 , C_1 and C_2 are called the marginal frequencies or marginal totals and show the total number of elements for which a variable evaluates to true or false. R_1 is the sum of O_{11} and O_{12} and, in our example, represents the total number of sentences in which word *A* occurs. R_2 is the sum of O_{21} and O_{22} and is the total number of sentences in which word *A* does not occur. Similarly, C_1 , the total number of sentences that word *B* occurs in, is the sum of O_{11} and O_{21} and C_2 , the total number of sentences word *B* does not occur in, is the sum of O_{12} and O_{22} . The fields O_{11} to O_{22} sum up to N . The values for the whole contingency table can be derived from the four frequencies O_{11} , R_1 , C_1 and N which Evert (2004: 36) refers to as frequency signature.

The usual approach to the analysis of cooccurrence data is to formulate a nonparametric statistical model that assumes a multinomial sampling distribution for the contingency table and interdependent binomial distributions for the individual cells (cf. Evert, 2004: 44–47) and to formulate a null hypothesis of independence with maximum-likelihood estimates² for the necessary population parameters (cf. Evert, 2004: 49–51). Note that while we do postulate population parameters, we do not need to make any assumptions about the population distribution in the model, hence it is a nonparametric model. For our purposes, the population is fully described by three parameters: I) the occurrence probability of word *A*, $P(A = a)$, II) the occurrence probability of word *B*, $P(B = b)$, and III) the cooccurrence probability of words *A* and *B*, $P(A = a \wedge B = b)$. In terms of those model parameters, the null hypothesis can be expressed as follows:

$$P(A = a \wedge B = b) = P(A = a) \cdot P(B = b)$$

² “The term maximum likelihood refers to a method of estimating parameters of a population from a random sample. [...] The method consists in choosing an estimator of unknown parameters whose values maximize the probability of obtaining the observed sample.” (Dodge, 2008: 334) In our case the maximum likelihood estimates are the exact proportions of the relevant phenomena in the sample, i. e. it is a really straightforward way to estimate the parameter values.

3 Statistical analysis of cooccurrence data

This means that the occurrence probabilities of words *A* and *B* are independent of each other (think of rolling two dice) and that all cooccurrences of the two are due to chance. Same as when trying to get two sixes when rolling two dice.

The frequencies that are expected under that null hypothesis are given in Table 3.2 and can be calculated as follows:

$$E_{ij} = \frac{R_i C_j}{N}$$

The expected frequencies sum up to the same marginal frequencies as the observed frequencies.

	$B = b$	$B \neq b$	
$A = a$	$E_{11} = \frac{R_1 C_1}{N}$	$E_{12} = \frac{R_1 C_2}{N}$	R_1
$A \neq a$	$E_{21} = \frac{R_2 C_1}{N}$	$E_{22} = \frac{R_2 C_2}{N}$	R_2
	C_1	C_2	N

Table 3.2: 2×2 contingency table of expected values

Fully worked out statistical models for relational cooccurrences, segment-based cooccurrences and distance-based cooccurrences can be found in Evert (2004) on pages 44–54, 66–68 and 68–70, respectively.

3.3 Association measures

If the observed frequencies deviate from those expected under the null hypothesis of independence, then this might be an indicator that the two units cooccurring together are not independent but in some way associated. That association can either be positive (if two units cooccur more often than expected under the null hypothesis) or negative (if two units cooccur less often than expected). Association measures that capture the difference between positive and negative association are called one-sided (or one-tailed) association measures that do not distinguish between the two are called two-sided (or two-tailed). For one-sided measures, higher scores indicate more strongly positive associations, whereas for two-sided measures, higher scores just indicate a stronger association in either direction. The general suggestion for corpus linguistic problems is to always use two-sided measures unless there

are very good reasons to assume that a violation of the null hypothesis in the other direction is impossible (cf. Baroni and Evert, 2009: 782). As mentioned before, one-sided measures are usually preferred in the analysis of cooccurrence data, since we are usually only interested in positively associated items (attraction) and less so in negatively associated items (repulsion). It is also possible to convert two-sided measures that always yield a positive score into one-sided measures by multiplying their score by -1 if $O_{11} < E_{11}$.³

For the analysis of linguistic cooccurrence data, a lot of association measures have been proposed in the literature. Evert (2004: 75–91) extensively discusses more than 30 different measures (most of them are also listed on the companion website <http://www.collocations.de/AM/>), Pecina (2005) (also in Pecina and Schlesinger, 2006 and Pecina, 2010) gives a list of 57 measures,⁴ Wiechmann (2008: 253) compares 47 measures “in a task of predicting human behavior in an eye-tracking experiment”. In this section, we will not try to give a complete overview of all the available measures. Instead, we focus on introducing those measures that will be used in later chapters and any necessary prerequisite measures (for a motivation of the choice of measures cf. Section 5.1).

3.3.1 Exact tests

“Mathematicians generally agree that the most appropriate significance test for contingency tables is Fisher’s exact test” (Evert, 2009: 1235; for arguments cf. Yates, 1984), a hypothesis test that calculates exact p -values based on probabilities of the whole contingency table:

Fisher’s [exact test] is based on the hypergeometric distribution and calculates the exact probability of observing the distribution seen in the table or a more extreme distribution, hence the word ‘exact’ in the title.

Boslaugh (2012: 132)

Its use with linguistic cooccurrence data has been popularized by Pedersen (1996) and Stefanowitsch and Gries (2003: 218) recommend it for their collocation analyses as the “one statistic that is not subject to [...] theoretical and/or distributional shortcomings”. By summing over the probabilities of

³ If such a converted measure is used to test for significance, the obtained p -values “have to be divided by 2 in order to allow direct comparison with one-sided tests.” (Evert, 2004: 81) Generally, it is considered good practice to conduct one-sided tests at half the significance level of two-sided tests (cf. Agresti, 2013: 92).

⁴ All in all, Pecina (2005: 653) lists 84 “[l]exical association measures used for bigram collocation extraction”, but not all of them are based on a 2×2 contingency table.

observing contingency tables with cooccurrence frequencies greater than or equal to O_{11} , we arrive at the one-sided form of Fisher's exact test:

$$\text{Fisher} = \sum_{k=O_{11}}^{\min\{R_1, C_1\}} \frac{\binom{C_1}{k} \binom{C_2}{R_1-k}}{\binom{N}{R_1}}$$

As mentioned in Section 3.2, our statistical model assumes that the random variable for an individual cell of the contingency table follows a binomial sampling distribution. The most interesting cell for analyzing the association between the elements of a cooccurrence is the top left cell containing the cooccurrence frequency O_{11} . By summing over the probabilities of observing cooccurrence frequencies greater than or equal to O_{11} , we arrive at the one-sided form of the binomial test:

$$\text{binomial} = \sum_{k=O_{11}}^N \binom{N}{k} \left(\frac{E_{11}}{N}\right)^k \left(1 - \frac{E_{11}}{N}\right)^{N-k}$$

Like Fisher's exact test, the binomial test is an exact test that calculates exact p -values – but in contrast to Fisher's test, the binomial test does not take into account the whole contingency table but only a single cell.

The major downside of the binomial test and (even more so) Fisher's exact test is that their calculation is computationally expensive. Therefore (and historically earlier), asymptotic hypothesis tests have been much more commonly used. Asymptotic hypothesis tests calculate a test statistic that, for large samples, asymptotically approaches a known limiting distribution that can be used to determine an (approximate) p -value. In practice, for collocational analyses, the test statistics are usually not converted to p -values but are used directly to rank the cooccurrences.

3.3.2 Approximations to Fisher's exact test

Pearson's chi-squared test is the standard test of independence for categorical data in mathematical statistics. It is a two-sided test with the χ^2 -distribution as its limiting distribution.

$$\text{chi-squared} = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{N(O_{11} - E_{11})^2}{E_{11}E_{22}}$$

Despite its popularity in other fields, Pearson's chi-squared test “has not very often been used for cooccurrence analysis so far” (Evert, 2004: 81). One of

the reasons for that might be that chi-squared gives a poor approximation to its limiting χ^2 -distribution for rare events, overestimating their significance. That kind of skewed contingency tables (where $N \gg O_{11}$) is quite typical for linguistic cooccurrence data.

The usual criterion for the applicability of the chi-squared test, usually credited to Cochran (1954: 420), is “that no cell has an expected value less than 1, and no more than 20% of the cells have an expected value less than 5.” (Boslaugh, 2012: 131) Since we only have four cells in a 2×2 contingency table, that amounts to the condition that $E_{ij} \geq 5$.⁵

Dunning (1993) suggested the use of logarithms of likelihood ratios (under the name of “log-likelihood”) as an association measure for linguistic cooccurrence data. For the multinomial distribution of contingency tables, the log-likelihood ratio is:

$$\text{log-likelihood} = 2 \sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}}$$

Just like chi-squared, it is a two-sided measure and its test statistic will be asymptotically χ^2 -distributed when the sample size N approaches ∞ . However, Dunning (1993) argues that log-likelihood gives a much better approximation to the limiting χ^2 -distribution than chi-squared, especially for the kind of rare events that are problematic for chi-squared. Similarly, Williams (1976: 49) argues that chi-squared should not be computed if the condition $|O - E| < E$ does not hold in one or more cells of the contingency table and that therefore log-likelihood is generally preferable to chi-squared. Additionally, by conducting simulation experiments, Evert (2004: 112) finds that “log-likelihood gives an excellent approximation to the Fisher p -values”, much better than chi-squared. Therefore, log-likelihood is “probably the best or second-best measure on mathematical grounds” (Gries, 2013: 148).

3.3.3 Approximations to the binomial test

The binomial distribution that our model assumes for the individual cells of the contingency table can be approximated by a normal distribution with a mean of E_{ij} and a standard deviation of $\sqrt{E_{ij}(1 - \frac{E_{ij}}{N})} \approx \sqrt{E_{ij}}$ (cf. Baroni and Evert, 2009: 788). Using that information, we can calculate the z-score, the

⁵ There are of course several variations of that rule of thumb. Rayson et al. (2004), for example, subject Cochran’s rule to simulation experiments to derive more fine-grained recommendations.

number of standard deviations the observed frequency is above or below the mean:

$$\text{z-score} = \frac{O_{11} - E_{11}}{\sqrt{E_{11}}}$$

The z-score is a one-sided measure with the standard normal distribution (mean of 0, standard deviation of 1) as its limiting distribution.

The problem with the z-score is that the normal approximation to the binomial distribution is not always very accurate. Dodge (2008: 47–49) recommends it only for sample sizes larger than 25, a requirement that is usually met in corpus linguistics. Baroni and Evert (2009: 788), on the other hand, recommend an expected frequency of at least 9, the approximation becoming very inaccurate for expected frequencies smaller than 1 – a requirement that is often not met for linguistic cooccurrence data (cf. Evert, 2009: 1227).

Another one-sided measure is the *t*-score, the test statistic of Student's *t*-test. It has a *t*-distribution as its limiting distribution and has been shown to perform very well on a collocation extraction task (cf. Uhrig and Proisl, 2012).

$$\text{t-score} = \frac{O_{11} - E_{11}}{\sqrt{O_{11}}}$$

“From a theoretical perspective, Student's test is not applicable to cooccurrence frequency data” (Evert, 2004: 82), since it makes the impossible assumption that the underlying binary indicator variables have a normal distribution. However, in light of its good performance in collocation extraction tasks, the *t*-score measure might be interpreted as a heuristic variant of the z-score measure using O_{11} instead of E_{11} in the denominator.

3.3.4 Measures of effect size

A completely different approach to quantifying the association between two units is to measure the effect size. While significance tests are “more concerned with the amount of evidence provided by the sample” (Evert, 2004: 77), those measures “compute a coefficient of association strength” (Evert, 2004: 54). Among the measures of effect size, there is a family of measures that rely on the conditional probabilities of observing a cooccurrence given one of the two elements, i. e. $P(B = b|A = a)$ and $P(A = a|B = b)$. The maximum likelihood estimates for those conditional probabilities are the proportions $\frac{O_{11}}{R_1}$ and $\frac{O_{11}}{C_1}$. In order to arrive at a single value, the conditional probabilities can be combined via a “link function”, e. g. using the minimum

(leading to the minimum sensitivity measure) or the geometric mean of the two. Using the harmonic mean leads to the Dice coefficient:

$$\text{Dice} = \frac{2O_{11}}{R_1 + C_1}$$

The Dice coefficient is a monotonic transformation of the set-theoretically motivated Jaccard coefficient: $\text{Dice} = \frac{2 \cdot \text{Jaccard}}{1 + \text{Jaccard}}$ or $\text{Jaccard} = \frac{\text{Dice}}{2 - \text{Dice}}$, meaning that it produces identical results when used for ranking cooccurrences:

$$\text{Jaccard} = \frac{O_{11}}{O_{11} + O_{12} + O_{21}}$$

The main difference between the two is that Jaccard can be turned into a distance metric by subtracting it from 1, and Dice cannot, because it does not satisfy the triangle inequality (cf. Charikar, 2002).

3.4 Criticism

As mentioned at the beginning of this chapter, the analytical framework described above does have its problems. Some fundamental criticism has been leveled against statistical hypothesis testing in general. That criticism applies to the whole framework in general and is not specific to its application in corpus linguistics. Nevertheless, when summarizing that criticism and the alternatives suggested in the literature in Section 3.4.1, a special focus will be on that area.

A general problem in corpus linguistics that is discussed in Section 3.4.2 is that the whole statistical framework depends on the assumption that the data we are analyzing were independently and randomly drawn from the population – something that is usually not the case in corpus linguistics.

Finally, there is some doubt on whether 2×2 contingency tables and established association measures are sufficient for the analysis of linguistic cooccurrence data (cf. Section 3.4.3).

It is worth pointing out again that these problems are not specific to the current work and that it is neither the aim nor within the scope of this work to solve any of them.

3.4.1 Criticism of statistical hypothesis testing

Some of the criticism of statistical hypothesis testing in other disciplines centers around the fact that the resulting p -value is sometimes misinterpreted

as an indicator of an effect's size or its replicability or as the probability that the null hypothesis is true (cf. Loftus, 1991):

What's wrong with [null hypothesis significance testing]? Well, among many other things, it does not tell us what we want to know, and we so much want to know what we want to know that, out of desperation, we nevertheless believe that it does! What we want to know is "Given these data, what is the probability that H_0 is true?" But as most of us know, what it tells us is "Given that H_0 is true, what is the probability of these (or more extreme) data?"

Cohen (1994: 997)

Part of that criticism also applies to the analysis of cooccurrence data, where cooccurrences are ranked by their p -values and it is usually (implicitly) assumed that lower p -values indicate a greater association strength, i. e. a greater effect size, instead of merely a lower probability of observing the data if the null hypothesis were true (see, for example, Stefanowitsch and Gries, 2003).

Given that most null hypotheses cannot be really true in the first place (cf. Loftus, 1996: 162–163), especially in linguistics where we know that "language is not random" (Kilgariff, 2005: 264), the question is what we gain from rejecting it. If all we can learn from rejecting the null hypothesis of independence is that the cooccurrence frequency of two linguistic units is very unlikely to have arisen from chance, i. e. that language is not random, "we have not learned very much" (Cohen, 1994: 1001).

Additionally, the p -value is dependent on the sample size. An n times bigger corpus leads to n times larger chi-squared or log-likelihood scores and therefore to smaller p -values (cf. Section 5.1.1). That means that even tiny deviations from the null hypothesis (that are to be expected since we do not believe it to be literally true) can become statistically significant if the sample is large enough. As Kilgariff puts it:

Language is non-random and hence, when we look at linguistic phenomena in corpora, the null hypothesis will never be true. Moreover, where there is enough data, we shall (almost) always be able to establish that it is not true.

Kilgariff (2005: 272)

That is not a problem per se, but "the problem arises where the literal falsity of the [randomness] assumption is overlooked, and inappropriate inferences are drawn" (Kilgariff, 2005: 264). Taking the sample size out of the equation leads to measures of effect size like phi or Cramér's V (which, for 2×2 tables,

is equivalent). A core element of these measures is to divide the chi-squared score (that scales linearly with sample size) by the sample size:

$$\phi = V = \sqrt{\frac{\chi^2}{N}}$$

Most of the objections above, although valid in general, do not apply to the typical approach to analyzing cooccurrence data, where most researchers are not interested in a fully automated decision procedure for distinguishing between, e.g., collocations and non-collocations, but in a ranked list of potential collocations with the most promising candidates at the top of the list. The thinking behind the use of p -values in that context goes like this: “If it is very improbable to observe a cooccurrence frequency like this under the null hypothesis of independence (as indicated by a small p -value), then we assume the two items are associated and are a good candidate for a collocation. And the less probable it is to observe a given cooccurrence frequency, the better a candidate that cooccurrence is.” That results in the de facto interpretation of p -values as relative indicators of effect size.⁶ The absolute p -value and whether it is above or below any arbitrary threshold is usually not the main concern as becomes evident from the following suggestion:

The p -value is a probability in the range $pv \in (0, 1]$, with smaller values indicating more evidence for a (usually positive) association. It is often convenient to use the negative decadic logarithm $-\log_{10} pv \in (0, \infty)$ instead, which adheres to the convention that high scores should indicate strong association.

Evert (2004: 79)

The alternatives to hypothesis testing suggested in the literature include, among others, to use measures of effect size, to provide confidence intervals or to “use techniques from exploratory data analysis (plotting, lattice/Trellis graphs, etc.)” (Gries, 2005: 278; cf. Loftus, 1996: 165–169). As already mentioned in Section 3.3, measures of effect size are well-established in the analysis of cooccurrence data, with Dice and Mutual Information being two of the most prominent measures. The use of confidence intervals has also been explored (cf. Blaheta and Johnson, 2001 and Evert, 2004: 86–88), but does not seem to have become mainstream, yet.

⁶ Which, looking at the example of chi-squared and Cramér’s V , is perfectly all right: Within the same corpus, the sample size N is a constant and larger chi-squared values result in smaller p -values and larger values for Cramér’s V .

For a different family of problems – testing the significance of differences between groups – there are also proposals that completely abandon contingency tables. Lijffijt et al. (2016), for example, argue that the problems mentioned above and in Section 3.4.2 can be prevented by representing the data differently, i. e. not as 2×2 contingency tables (cf. also the next section) but as frequency lists or lists of inter-arrival times. For their use case of comparing frequencies between two corpora, they apply tests like Welch's *t*-test (an adaptation of Student's two-sample *t*-test that does not assume equal variance in the two populations), the nonparametric Wilcoxon rank-sum test (also known as the Mann-Whitney *U*-test) or a bootstrap test based on the mid-*p* test. They find that the *p*-values resulting from those tests give less spurious results than chi-squared or log-likelihood and are more appropriate in that context.

3.4.2 **Adequacy of the randomness assumption in corpus linguistics**

The random sampling model assumes that samples are independently and randomly drawn from the population and that the sampling unit corresponds to the unit of analysis. In the study of cooccurrences, the unit of analysis can, for example, be a sentence (for segment-based cooccurrences) or a pair of words connected via a dependency relation (relational cooccurrences). The sampling unit in a corpus, however, is virtually always a larger stretch of continuous text or a complete document (e. g. a news article, book or TV show transcript). While that sampling strategy is necessary to be able to use a corpus for investigations in text linguistic phenomena like cohesion (Halliday and Hasan, 1976), these very same phenomena lead to clustering or repetition effects, where, for example, a certain word is used more frequently in a text on a certain topic than in the overall population. Empirically, we find that for many words, especially content words, the probability of repeated occurrences in a text is much higher than expected under a random sampling model, i. e. the individual occurrences are not independent. Instead, the probability of observing a second occurrence in a text is much higher than the probability of observing a first occurrence, a phenomenon called burstiness (Katz, 1996) or adaptation (Church, 2000). These effects, introduced by the mismatch between sampling unit and unit of analysis, are a violation of the randomness assumption. They lead to inflated observed frequencies, an underestimation of the true variances in the population and an overestimation of the significance of the results (cf. Evert, 2006: 184–188).

The null hypothesis of independence also evokes the notion of randomness in that it postulates that the two linguistic units in the cooccurrence are independent, i. e. that they only cooccur by chance. Of course, such a hypothesis is quite unrealistic, “[h]owever, this does not affect the validity of the random sample model (but it will affect the interpretation of association scores [...])” (Evert, 2004: 59). See also the discussion in the following section.

3.4.3 Beyond contingency tables

More recently, the use of “ 2×2 tables that hide much of the interesting variability in the data” (Gries, 2013: 159) has been found lacking certain features that might be desirable for the analysis of cooccurrence data. The desiderata that Gries (2013: 159) has for a new methodology, include the following: i) It should be able to measure the inherent directionality or asymmetry of many collocations. The traditional measures based on contingency tables are all symmetric. ii) It should take into account the dispersion of cooccurrences within the corpus and not just the total frequencies. iii) The influence of (the entropy of) type-token distributions should be accounted for. iv) The approach should be extendable to multiword cooccurrences.

Such a methodology has not been worked out, yet. Therefore, in the current work, we will stick to the traditional representation of cooccurrence data as 2×2 tables that allow for a wide variety of established analytical methods. Nevertheless, there has been work on some of the above issues that will be summarized below.

Asymmetric association measures

Gries (2013) introduces the asymmetric ΔP measure that is the difference of the probability of observing the one word given the occurrence of the other minus the probability of observing the one word given the absence of the other. The resulting two measures are:

$$\Delta P_{B|A} = \frac{O_{11}}{R_1} - \frac{O_{21}}{R_2} = \frac{O_{11}O_{22} - O_{12}O_{21}}{R_1R_2}$$

$$\Delta P_{A|B} = \frac{O_{11}}{C_1} - \frac{O_{12}}{C_2} = \frac{O_{11}O_{22} - O_{12}O_{21}}{C_1C_2}$$

ΔP is based on the standard 2×2 contingency table and can therefore be trivially applied instead of or in addition to any other measure based on that table. $\Delta P_{A|B}$ is identical to the difference of proportions test statistic used by Liddell (1976), $\Delta P_{B|A}$ corresponds to the application of that test statistic to

the transposed contingency table. In an evaluation against two collocation dictionaries, Evert et al. (2017) find that ΔP performs quite well, although not as good as the best conventional symmetric measures in extracting collocation candidates.

Michelbacher et al. (2007) and Michelbacher et al. (2011) propose a method for turning any traditional symmetric association measure into an asymmetric rank measure. The forward rank measure for the cooccurrence of words A and B is determined by computing the association measure for all words cooccurring with A , ranking them and noting the rank of word B . Similarly, the backward rank measure is determined by computing the association measure for all words cooccurring with B , ranking them and noting the rank of word A . The difference between the ranks gives an indication of the asymmetry of the collocation. Depending on the number of words that cooccur with word A and the number of words that these words cooccur with in turn, that approach can become computationally expensive; and it can only be pursued if all cooccurring words are known.

Dispersion measures and adjusted frequencies

There are methods for measuring dispersion (cf. Gries, 2008 for a survey) or for determining whether a linguistic unit is underdispersed (cf. Baayen, 2001: 164–167; Evert, 2004: 60–63). Based on dispersion measures, it is also possible to calculate adjusted frequencies (cf. Gries, 2008 for a survey), though “it is often unclear what these values ‘mean’” (Gries, 2008: 426).

Type-based association measures

Daudaravičius and Marcinkevičienė (2004) propose a new association measure called lexical gravity that takes type frequencies into account and thus goes beyond a 2×2 contingency table:

$$\text{gravity} = \log \left(\frac{O_{11} \cdot n(A)}{R_1} \cdot \frac{O_{11} \cdot n(B)}{C_1} \right)$$

where $\frac{R_1}{n(A)}$ is the coefficient of variety of words cooccurring with word A and $\frac{C_1}{n(B)}$ is the coefficient of variety of words cooccurring with word B . $n(A)$ and $n(B)$ are the number of types cooccurring with words A and B .

Multiword cooccurrences

Since the measures introduced in Section 3.3 are based on a two-dimensional contingency table, they are limited to the analysis of cooccurrences of two items. A straightforward way to extend the methodology to the analysis of multiword cooccurrences is to simply cross-classify more than two variables and apply equivalent association measures to the resulting multidimensional contingency tables under the hypothesis of mutual independence⁷ (cf. Everitt, 1992: 60–72). Villada Moirón (2005: 80–81), for example, uses extended versions of mutual information and chi-squared for the analysis of three-word cooccurrences. Sangati and van Cranenburgh (2015) use extended versions of mutual information and log-likelihood for cooccurrences of an arbitrary number of words within a given syntactic tree fragment. The main problem with these approaches is that the expected frequencies become extremely small.

Other approaches include the use of configural frequency analysis (Lienert, 1969; von Eye, 1990), log-linear models (Everitt, 1992: 73–97) or other types of generalized linear models (Dobson and Barnett, 2008). Stefanowitsch and Gries (2005), for example, use a configural frequency analysis to measure the overall degree of attraction between two words and a construction and Blaheta and Johnson (2001) use log-linear models with confidence intervals for the extraction of multiword verbs.

Other extraction methods for multiword cooccurrences try to do without an extension of the statistical framework. They apply a recursive or cyclic extraction strategy, working their way up from pairs to longer cooccurrences or perform post-processing steps on the extracted binary cooccurrences to infer longer ones (cf. Seretan, 2008: 139–141 for an overview). To quantify association strength, these approaches typically treat multiword parts of longer cooccurrences as single items in order to be able to use 2×2 contingency tables.

⁷ In a multidimensional contingency table we could also formulate hypotheses of partial independence “where two of the variables are independent in each level of the third, but each may be associated with this third variable.” (Everitt, 1992: 68) There can also be second and higher order relationships between variables. “For example, in a three-way table an association between two of the variables may differ in degree or in direction in different categories of the third” (Everitt, 1992: 70).

3.5 Summary

This chapter described the predominant approach to the statistical analysis of cooccurrence data. We gave a high-level introduction to inferential statistics, briefly covering statistical models and the basics of hypothesis testing. We showed how 2×2 contingency tables are used for cross-classifying cooccurrence data and described the assumptions usually made when statistically analyzing those contingency tables. Following that, we introduced an inventory of association measures that are most commonly used in the analysis of linguistic cooccurrences. The inventory comprises both significance tests and measures of effect size.

We concluded the chapter by giving a brief summary of the criticism that has been leveled against the approach. This includes criticism of statistical hypothesis testing in general as well as problems related to sampling and the adequacy of the randomness assumption that are more specific to corpus linguistics. Finally, we mentioned a few proposals that have been made to address some of the shortcomings of the approach.

4 A generalized model for the cooccurrence of linguistic structures

This is the central chapter of the current work. Here we develop a methodologically sound, formalized generalization of the notion of cooccurrence that goes beyond existing approaches like relational cooccurrences or collocation analysis.

First, we need to find a suitable formal representation for linguistic structures (Section 4.1). That representation should be able to represent linguistic entities, their properties and the relations between the entities. The examples in this chapter are taken from the realm of syntax, but the representation is not limited to that. It is also applicable to data from other realms of linguistics, e. g. morphology or semantics, or indeed to any other type of non-linguistic data that is made up of related entities with properties.

We also need to find ways to determine the frequencies of those linguistic structures in a corpus (cf. Sections 4.2 and 4.4). It turns out that this is not as trivial as it might seem at first glance and that there is more than one possible method of doing this.

With the necessary prerequisites in place, we can develop a generalization of relational cooccurrences, covarying collexeme analysis and simple collexeme analysis that is able to analyze cooccurrences of arbitrary (linguistic) structures: The generalized cooccurrence model (cf. Sections 4.3 and 4.5). Relational and segment-based cooccurrences as well as simple and covarying collexeme analysis are merely special cases of that much more flexible and powerful model (cf. Section 4.6). Section 4.7 shows a couple of ways to overcome seeming limitations and to leverage the full power of the generalized cooccurrence model.

The presentation aims to be self-contained, i. e. the reader should not need to have any special prior knowledge to follow the presentation. The necessary mathematical foundations will be given in the text.

4.1 Representing linguistic structures

In this section, we develop a formal representation for arbitrary linguistic structures that is suitable for our purpose of analyzing association between

structures. A popular representation format for linguistic structures in (computational) linguistics are feature structures or attribute-value matrices. They have been used in grammar frameworks such as lexical functional grammar (LFG; Bresnan, 1982), generalized phrase structure grammar (GPSG; Gazdar et al., 1985) or head-driven phrase structure grammar (HPSG; Pollard and Sag, 1994) and are well-suited to represent syntactic and morphological structures. They are basically sets of attribute-value pairs, where values can either be atomic or complex, i. e. other feature structures. According to most definitions, feature structures are formally equivalent to rooted directed acyclic graphs (DAGs; cf. Shieber, 1986: 20), although some variants allow for the possibility of cycles (cf. Carpenter, 1992: 35).

The stemmata used in dependency grammar to represent syntactic analyses are also graphs, allowing linguists to use graph-theoretical notions in their arguments. Groß and Osborne (2009), for example, use the notion of a chain (Definition 8 in Section 4.1.2) in their treatment of discontinuities. In many models, dependency structures are not limited to trees. Examples include Tesnière's analysis of example sentence 3 in Figure 4.1 and Hudson's analysis of example sentence 4 in Figure 4.2.

(3)_{TESNIÈRE (2015: 343)} Alfred buys new books and folders.

(4)_{HUDSON (2010: 178)} He saw Arthur and Bill.

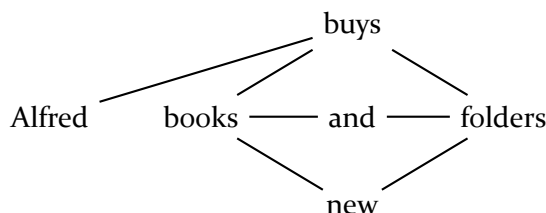


Figure 4.1: Analysis of example sentence 3 (Tesnière, 2015: 343)

In computational linguistics, the collapsed and the CCprocessed variants of the popular Stanford Typed Dependencies use cycles when there are relative clauses in a sentence, making it impossible to represent the dependency stemmata of those sentences as DAGs (cf. de Marneffe and Manning, 2015: 17–18). The same is true for the dependency relations in the Universal Dependencies project (Nivre et al., 2016) that supersedes the Stanford Dependencies¹ –

¹ The Universal Dependencies project (<http://universaldependencies.org/>) develops a cross-linguistic annotation scheme based on the Stanford Dependencies (de Marneffe et al., 2006;

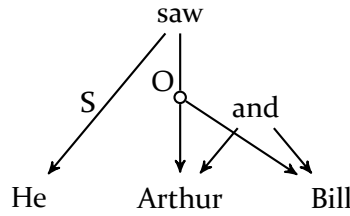


Figure 4.2: Analysis of example sentence 4 (Hudson, 2010: 178)

at least for the enhanced++ representation for English (Schuster and Manning, 2016). Example sentence 5 and the corresponding analysis in Figure 4.3 illustrate this kind of structure:

(5)_{BNC: A02 186} This is one lesson that could save their lives.

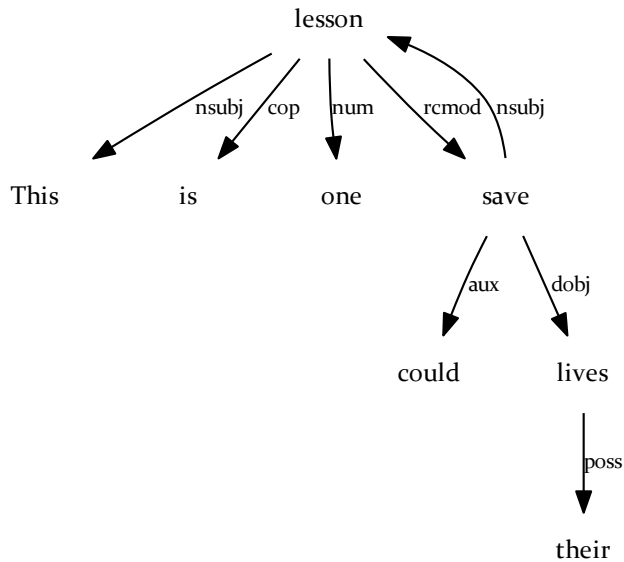


Figure 4.3: SD analysis of example sentence 5

The relative clause governed by *save* modifies *lesson* and at the same time *lesson* is, at least semantically, the subject of *save*. The rcmod dependency relation from *lesson* to *save* and the nsubj dependency relation from *save*

de Marneffe and Manning, 2008) and Universal Stanford Dependencies (de Marneffe et al., 2014), the Google universal part-of-speech tagset (Petrov et al., 2012) and Zeman's (2008) interlingua for morphosyntactic tagsets.

to *lesson* create a cycle. Since that use of cycles is linguistically motivated and potentially useful in applications, we should not prohibit the use of cycles in our representation of linguistic structures. We will therefore opt for representing linguistic structures as general directed graphs.

Graphs are also a popular and powerful² representation format for all kinds of linguistic data in Natural Language Processing and graph-based algorithms can be applied to a wide range of NLP problems (cf. Nastase et al., 2015 for a survey).

The remainder of this section formally defines our representation of linguistic structures and introduces the necessary prerequisites.

4.1.1 Relations and functions

First, we need to introduce the notion of a (binary) relation between two sets and the notions of domain and range (Definition 1). Based on that we can define functional relations (Definition 2) and finally functions (Definition 3), which we need for defining directed graphs in the next section.

Definition 1 (Gallier, 2011: 105). Given two sets A and B , a (binary) *relation between A and B* is any triple $\langle A, R, B \rangle$, where $R \subseteq A \times B$ is any set of ordered pairs from $A \times B$. When $\langle a, b \rangle \in R$, we also write aRb and we say that a and b are related by R . The set

$$\text{dom}(R) = \{a \in A \mid \exists b \in B, \langle a, b \rangle \in R\}$$

is called the *domain of R* and the set

$$\text{range}(R) = \{b \in B \mid \exists a \in A, \langle a, b \rangle \in R\}$$

is called the *range of R* . Note that $\text{dom}(R) \subseteq A$ and $\text{range}(R) \subseteq B$. When $A = B$, we often say that R is a (binary) *relation over A* .

Definition 2 (Gallier, 2011: 106). We say that a relation R between two sets A and B is *functional* if for every $a \in A$, there is *at most one* $b \in B$ so that $\langle a, b \rangle \in R$. Equivalently, R is functional if for all $a \in A$ and all $b_1, b_2 \in B$, if $a, b_1 \in R$ and $a, b_2 \in R$, then $b_1 = b_2$.

² Cf., for example, the graph-based Salt framework that can map between different linguistic annotation formats (Zipser and Romary, 2010).

Definition 3 (Gallier, 2011: 106). A *partial function* f is a triple $f = \langle A, G, B \rangle$, where A is a set called the *input domain of f* , B is a set called the *output domain of f* (sometimes *codomain of f*), and $G \subseteq A \times B$ is a functional relation called the *graph of f* [...]; we let $\text{graph}(f) = G$. We write $f : A \rightarrow B$ to indicate that A is the input domain of f and that B is the codomain of f and we let $\text{dom}(f) = \text{dom}(G)$ and $\text{range}(f) = \text{range}(G)$. For every $a \in \text{dom}(f)$, the unique element $b \in B$, so that $\langle a, b \rangle \in \text{graph}(f)$ is denoted by $f(a)$ (so, $b = f(a)$). Often we say that $b = f(a)$ is the *image of a by f* . The range of f is also called the *image of f* and is denoted $\text{Im}(f)$. If $\text{dom}(f) = A$, we say that f is a *total function*, for short, a *function with domain A* .

Let us illustrate the main concepts using a function for calculating the reciprocal of a natural number. We define the function $f = \langle \mathbb{N}_+, \{(x, \frac{1}{x}) \in \mathbb{N}_+ \times \mathbb{Q} \mid x \in \mathbb{N}_+\}, \mathbb{Q} \rangle$, which is more commonly written as $f : \mathbb{N}_+ \rightarrow \mathbb{Q}, f(x) = \frac{1}{x}$. The input domain of f is $\mathbb{N}_+$, the set of natural numbers without zero, i.e. all positive integers, and the output domain of f is $\mathbb{Q}$, the set of rational numbers. The domain of f is the set of elements from the input domain for which the function produces an output that is in the output domain. In our case, the domain of f is identical to the input domain because all reciprocals of positive integers are rational numbers, meaning that f is a total function. The range of f is the set of elements from the output domain which can actually be obtained by applying f to the elements of the input domain. In our case, the range of f is a subset of the output domain because not all rational numbers are reciprocals of positive integers. There is, for example, no positive integer x such that $f(x) = \frac{3}{7}$.

We also need to introduce a few further properties of functions (Definition 4). The concept of a surjective function is similar to that of a total function. While a total function produces an output in its output domain for all elements in its input domain, i.e. its domain equals its input domain, a surjective function produces all elements in its output domain from inputs in its input domain, i.e. its range equals its output domain. The function $f = \langle \mathbb{N}, \{(x^2, x) \in \mathbb{N} \times \mathbb{N} \mid x \in \mathbb{N}\}, \mathbb{N} \rangle$ that relates square numbers to their principal roots, for example, is surjective but not total because every natural number is the principal square root of another natural number (its square) but not every natural number has a principal square root that is a natural number, i.e. not every natural number is a square number.

A function is injective if it yields different outputs when given different inputs. For example, the function $f = \langle \mathbb{R}, \{(x, x^2) \in \mathbb{R} \times \mathbb{R} | x \in \mathbb{R}\}, \mathbb{R} \rangle$, which could also be written as $f : \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^2$, is not injective because two different inputs, e. g. 2 and -2 can yield the same output, in this case 4.

A bijective function is both injective and surjective. For example, the function $f = \langle \mathbb{R}, \{(x, x^3) \in \mathbb{R} \times \mathbb{R} | x \in \mathbb{R}\}, \mathbb{R} \rangle$, which could also be given as $f : \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^3$, is bijective. It is injective because no two different real numbers have the same cube and it is surjective because every real number can be expressed as the cube of another real number (that other real number can be obtained by the inverse of the function (Definition 5); in this case the inverse of f is given by $f^{-1}(x) = \sqrt[3]{x}$). It is also total because all real numbers can be cubed and have real numbers as cubes.

Definition 4 (Gallier, 2011: 124). Given any function $f : A \rightarrow B$, we say that f is *injective* (or *one-to-one*) iff for all $a, b \in A$, if $f(a) = f(b)$, then $a = b$, or equivalently, if $a \neq b$, then $f(a) \neq f(b)$. We say that f is *surjective* (or *onto*) iff for every $b \in B$, there is some $a \in A$ so that $b = f(a)$, or equivalently if $\text{Im}(f) = B$. The function f is *bijective* iff it is both injective and surjective. When $A = B$, a bijection $f : A \rightarrow A$ is called a *permutation of A*.

Definition 5 (cf. Gallier, 2011: 122, 125–126). If a function $f : A \rightarrow B$ is bijective, there is a unique function $g : B \rightarrow A$ such that $g \circ f = \text{id}_A$ and $f \circ g = \text{id}_B$.^{ab} Function g is called the *inverse of f* and is denoted f^{-1} .

^a $g \circ f$ denotes the *composition of f and g*. Simply put, if $h = g \circ f$, then $h(x) = g(f(x))$. More formally, if $f : A \rightarrow B$ and $g : B \rightarrow C$ then $g \circ f = \langle A, \{\langle a, c \rangle \in A \times C | \exists b \in B, \langle a, b \rangle \in \text{graph}(f) \text{ and } \langle b, c \rangle \in \text{graph}(g)\}, C \rangle$ (cf. Gallier, 2011: 117–118).

^b id_A denotes the the identity function on A , i. e. $\text{id}_A(a) = a$ for all $a \in A$ (cf. Gallier, 2011: 105).

4.1.2 Directed graphs

Directed graphs consist of vertices and directed edges between those vertices. Definition 6 is quite general and allows for the graph in Figure 4.4 which has a loop (edge 1) and parallel edges (edges 2 and 3). Usually, loops and parallel edges (Definition 7) are not needed for representing linguistic structures. In the realm of syntax, for example, vertices might represent tokens and edges might represent dependency relations between tokens. Most models of

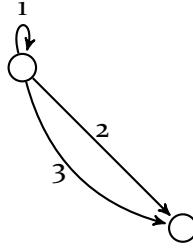


Figure 4.4: Directed graph with loop and parallel edges

grammar would assume that one token can govern another token only via a single dependency relation. On a similar note, it is usually not assumed that tokens govern themselves via a dependency relation. According to the area of application, we could therefore forbid parallel edges (restricting ourselves to simple directed graphs) and loops:

$$\begin{aligned} \forall e, e' \in E : s(e) \neq s(e') \text{ or } t(e) \neq t(e') \\ \forall e \in E : s(e) \neq t(e) \end{aligned}$$

The generalized cooccurrence model presented in this chapter, however, can and will be formalized without such restrictions. While some features of the model are not necessary for representing dependency structures, the greater generality of the model results in a much greater expressive power that also benefits the search for dependency structures (cf. Section 4.7).

We also want to make sure that the graph is connected (Definition 9), i. e. that it is possible to reach every vertex of the graph from any other vertex by walking along the edges of the graph while ignoring the orientation of the edges.

Sometimes it can be useful to refer to the number of edges starting or ending at a given vertex. This property of a vertex is called its degree (Definition 10).

Definition 6 (Gallier, 2011: 168). A *directed graph* (or *digraph*) is a quadruple $G = (V, E, s, t)$, where V is a set of *nodes* or *vertices*, E is a set of *arcs* or *edges*, and $s, t : E \rightarrow V$ are two functions, s being called the *source function* and t the *target function*. Given an edge $e \in E$, we also call $s(e)$ the *origin* or *source* of e , and $t(e)$ the *endpoint* or *target* of e .

Definition 7 (Gallier, 2011: 168). Given a directed graph G , an edge $e \in E$, such that $s(e) = t(e)$ is called a *loop* (or *self-loop*). Two edges $e, e' \in E$ are said to be *parallel edges* iff $s(e) = s(e')$ and $t(e) = t(e')$. A directed graph is *simple* iff it has no parallel edges.

Definition 8 (Gallier, 2011: 183). Given any digraph $G = (V, E, s, t)$ and any two nodes $u, v \in V$, a *chain* (or *walk*) *from u to v* is a sequence $\pi = (u_0, e_1, u_1, e_2, u_2, \dots, u_{n-1}, e_n, u_n)$, where $n \geq 1$; $u_i \in V$; $e_j \in E$ and

$$u_0 = u; u_n = v \text{ and } \{s(e_i), t(e_i)\} = \{u_{i-1}, u_i\}, 1 \leq i \leq n.^a$$

^a Note the set notation that states that e_i is either an edge from u_{i-1} to u_i or vice versa. An equivalent but more verbose formalization: $\langle s(e_i), t(e_i) \rangle = \langle u_{i-1}, u_i \rangle$ or $\langle s(e_i), t(e_i) \rangle = \langle u_i, u_{i-1} \rangle$.

Definition 9. A directed graph $G = (V, E, s, t)$ is *connected* iff for all $u, v \in V$ there is a chain from u to v .

Definition 10 (cf. Gallier, 2011: 169). For any vertex $v \in V$ of a directed graph $G = (V, E, s, t)$, we say that

$$\begin{aligned} d_G^+(v) &= |\{e \in E | s(e) = v\}| \text{ is the } \textit{outdegree} \text{ of } v, \\ d_G^-(v) &= |\{e \in E | t(e) = v\}| \text{ is the } \textit{indegree} \text{ of } v, \text{ and} \\ d_G(v) &= d_G^+(v) + d_G^-(v) \\ &= |\{e \in E | s(e) = v \text{ or } t(e) = v\}| \text{ is the } \textit{degree} \text{ of } v. \end{aligned}$$

4.1.3 Subgraph isomorphism

A central question when working empirically with syntactic structures is: Does a given sentence contain an instance of the syntactic structure we are looking for? As both the syntactic analysis of the sentence and the syntactic structure we are looking for can be represented as directed graphs, we can restate the question as the subgraph isomorphism problem: Does digraph $G_1 = (V_1, E_1, s_1, t_1)$ contain a subgraph (Definition 11) $G'_1 = (V'_1, E'_1, s'_1, t'_1)$ that is isomorphic (Definition 12) to digraph $G_2 = (V_2, E_2, s_2, t_2)$?

Definition 11 (Gallier, 2011: 181). Given any two digraphs $G = (V, E, s, t)$ and $G' = (V', E', s', t')$, we say that G' is a *subgraph of G* , written as $G' \subseteq G$ (cf. Diestel, 2010: 3) iff $V' \subseteq V$, $E' \subseteq E$, s' is the restriction of s to E' and t' is the restriction of t to E' . [...] Given any subset V' of V , the *induced subgraph $G\langle V' \rangle$* of G is the graph $(V', E_{V'}, s', t')$ whose set of edges is

$$E_{V'} = \{e \in E \mid s(e) \in V'; t(e) \in V'\}.$$

(Clearly, s' and t' are the restrictions of s and t to $E_{V'}$, respectively.) Given any subset, $E' \subseteq E$, the graph $G' = (V, E', s', t')$, where s' and t' are the restrictions of s and t to E' , respectively, is called the *partial graph of G generated by E'* . The graph $(V', E' \cap E_{V'}, s', t')$ is a *partial subgraph of G* (here, s' and t' are the restrictions of s and t to $E' \cap E_{V'}$, respectively).

Definition 12 (cf. Gallier, 2011: 171; Diestel, 2010: 3). Given two directed graphs $G_1 = (V_1, E_1, s_1, t_1)$ and $G_2 = (V_2, E_2, s_2, t_2)$, an *isomorphism $f : G_1 \rightarrow G_2$ from G_1 to G_2* is a pair $f = (f^v, f^e)$ of total, bijective functions $f^v : V_1 \rightarrow V_2$ and $f^e : E_1 \rightarrow E_2$ such that, for every edge, $e \in E_1$, we have

$$s_2(f^e(e)) = f^v(s_1(e)) \text{ and } t_2(f^e(e)) = f^v(t_1(e)).$$

If such an isomorphism exists, we say that G_1 and G_2 are isomorphic and write $G_1 \simeq G_2$.

Subgraph isomorphism is an NP-complete problem (cf. Garey and Johnson, 1979: 202). NP-complete problems are a class of problems for which no algorithm running in polynomial time, i. e. having a worst-case runtime of $O(n^k)$ with some constant k for inputs of size n , has been discovered, yet (cf. Cormen et al., 1990: 916). On the other hand, it has not been proven, yet, that this class of problems has a superpolynomial-time lower bound. The effect is that, for larger problem sizes, i. e. larger graphs in our case, a problem can become computationally intractable, requiring an exorbitant amount of time.

Let us illustrate the concepts of isomorphism and subgraph isomorphism with a few examples involving undirected graphs.³ If two graphs G_1 and G_2 are isomorphic, then all the vertices of G_1 can be mapped to vertices of G_2 such that adjacent vertices in G_1 are mapped to adjacent vertices in G_2 and vice versa, i. e. in such a way that incidence is preserved (cf. Definition 13). That means that if two graphs are isomorphic, they are structurally identical. An isomorphism is just a renaming of the vertices and edges. For a more visual analogy, if we think of the edges of a graph as rubber bands connecting the vertices, an isomorphism moves the vertices around, lengthening, shortening or crossing the rubber bands but leaving all of them intact. For practical purposes, we can treat isomorphic graphs as identical, or, as Diestel (2010: 3) puts it: “We do not normally distinguish between isomorphic graphs. Thus, we usually write $G = G'$ rather than $G \simeq G'$ [...]” More formally, isomorphism is an equivalence relation (Definition 14) and the set $[G] = \{H \mid G \simeq H\}$ of all graphs that are isomorphic to graph G is called the equivalence class of G (Definition 15). Thus, if $G_1 \simeq G_2$, then $[G_1] = [G_2]$.

Definition 13 (Gallier, 2011: 169). Given a directed graph G , for any edge $e \in E$, if $u = s(e)$ and $v = t(e)$, we say that [...] [t]he nodes u and v are *incident to the arc e* [...] [and that] [t]he arc e is *incident to the nodes u and v* .

Definition 14 (Gallier, 2011: 291). A binary relation R on a set X is an *equivalence relation* iff it is *reflexive*, *transitive*, and *symmetric*, that is:

1. (*Reflexivity*): aRa , for all $a \in X$.
2. (*Transitivity*): If aRb and bRc , then aRc , for all $a, b, c \in X$.
3. (*Symmetry*): If aRb , then bRa , for all $a, b \in X$.

³The difference between directed and undirected graphs is that the edges of the former are directed and those of the latter are not. For a formal definition of undirected graphs without loops we can change Definition 4.1.2 such that the source and target functions s and t are replaced with an endpoint function $st : E \rightarrow [V]^2$ that assigns a set of endpoints to every edge, where $[V]^2$ denotes the set of all two-element subsets of V (cf. Gallier, 2011: 185). Alternatively, we could think of an undirected graph as a directed graph with symmetric edges, i. e. between every pair of adjacent vertices u and v there is one edge from u to v and another one from v to u .

Definition 15 (Gallier, 2011: 292). Given an equivalence relation R on a set X for any $x \in X$, the set

$$[x]_R = \{y \in X \mid xRy\}$$

is the *equivalence class* of x . Each equivalence class $[x]_R$ is also denoted $\bar{x}_R$ and the subscript R is often omitted when no confusion arises.

Consider the two example graphs in Figure 4.5: Although they look quite different, they are really just two different ways of drawing a graph known as the Petersen graph (Weisstein, n. d.: s. v. “Petersen Graph”⁴) which is most commonly drawn as in Figure 4.5a. The reader should convince herself that the vertices from the two graphs can indeed be mapped to each other while preserving incidence, e. g. by mapping A to 9, B to 8, C to 7, D to 6, E to 5, F to 1, G to 3, H to 10, I to 2 and J to 4.

Subgraph isomorphism then means that the structure of a smaller graph is contained within a larger graph. Figure 4.6 shows some graphs that are isomorphic to subgraphs of the Petersen graph in Figure 4.5. For some of these graphs, that relationship is clearly recognizable from the way they are drawn. The graphs in Figures 4.6a and 4.6b, for example, can easily be mapped to subgraphs of Figure 4.5a. Such subgraphs also exist for the graphs in Figures 4.6c to 4.6e, although they are less easy to spot. They are visualized in Figure 4.7.

Figure 4.8 shows some graphs that are not isomorphic to subgraphs of the Petersen graph. The Petersen graph contains neither triangles nor squares and none of its vertices has a degree of 4 that would be necessary for the cross in Figure 4.8c.

4.1.4 Partial orders

The dependency relations in the Stanford Dependencies Scheme are organized in a hierarchy (cf. de Marneffe and Manning, 2015: 11–12). A direct object, for example, is a kind of object, an object is a kind of complement, a complement is a kind of argument and an argument is a kind of dependent, the most general dependency relation. We can utilize this hierarchical organization to underspecify queries: If we are interested in both direct and indirect objects, we could just look for relations with a label equal to or more specific than *object*.

⁴ <http://mathworld.wolfram.com/PetersenGraph.html>

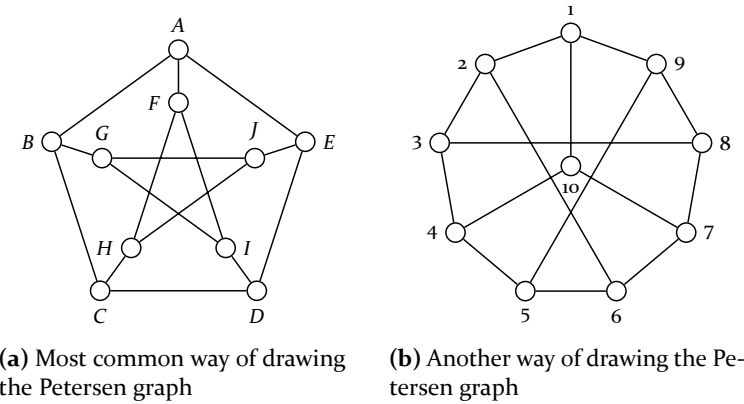


Figure 4.5: Two isomorphic graphs (Petersen graph)

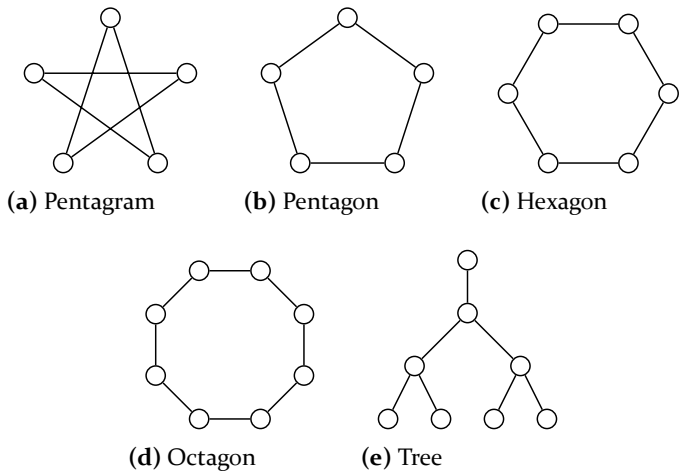
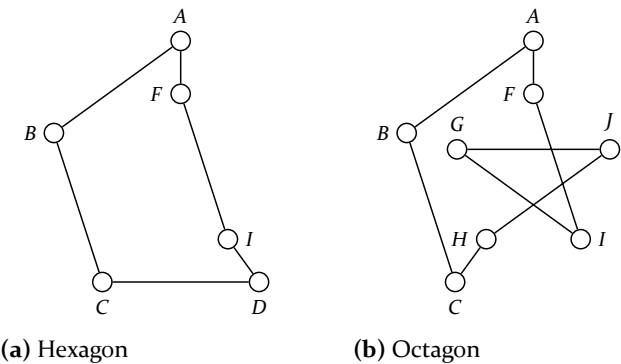


Figure 4.6: A couple of graphs isomorphic to subgraphs of the Petersen graph



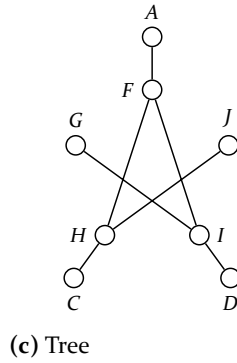


Figure 4.7: More easily recognizable drawings of subgraphs of the Petersen graph

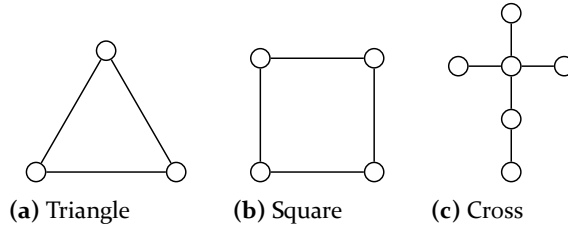


Figure 4.8: A couple of graphs not isomorphic to subgraphs of the Petersen graph

We can formalize such a hierarchy as a partially ordered set (Definition 16) and can visualize it as a Hasse diagram. In order to visualize a partially ordered set as a Hasse diagram, we represent it as a directed graph where the elements of the poset are the vertices of the digraph and where we add directed edges from every element to its immediate successor, i. e. we add an edge from a to b iff $a < b$ and there is no c such that $a < c < b$ (cf. Gallier, 2011: 260). Conventionally, Hasse diagrams are drawn in such a way that the edges are implicitly oriented, with predecessors (e. g. a) being drawn below successors (e. g. b).

Figure 4.9 shows a Hasse diagram visualizing a part of the poset representing the Stanford Dependencies hierarchy. As we can see, $\text{dep} < \text{arg} < \text{comp} < \text{obj} < \text{dobj}$. Elements that are not in the same chain, e. g. the two relations iobj and nsubj , are incomparable: neither is $\text{iobj} \leq \text{nsubj}$ nor $\text{nsubj} \leq \text{iobj}$.

Another example of a partial order is the subset ordering. The subsets of a set X can be ordered by the subset relation: $A \subseteq B$, where A and B are subsets of X . Figure 4.10 shows the Hasse diagram for the subset ordering of subsets of the set $\{a, b, c\}$, e. g. $\{b, c\} \subseteq \{a, b, c\}$, $\{b\} \subseteq \{b, c\}$ and $\{\} \subseteq \{b\}$.

Note that neither $\{b\} \subseteq \{a, c\}$ nor $\{a, c\} \subseteq \{b\}$. The two sets $\{b\}$ and $\{a, c\}$ are incomparable with respect to the subset relation.

Definition 16 (Gallier, 2011: 258). A binary relation $\leq$ on a set X is a *partial order* (or *partial ordering*) iff it is *reflexive*, *transitive*, and *antisymmetric*; that is:

1. (*Reflexivity*): $a \leq a$, for all $a \in X$.
2. (*Transitivity*): If $a \leq b$ and $b \leq c$, then $a \leq c$, for all $a, b, c \in X$.
3. (*Antisymmetry*): If $a \leq b$ and $b \leq a$, then $a = b$, for all $a, b \in X$.

A partial order is a *total order* (*ordering*) (or *linear order* (*ordering*)) iff for all $a, b \in X$, either $a \leq b$ or $b \leq a$. When neither $a \leq b$ nor $b \leq a$, we say that a and b are *incomparable*. A subset, $C \subseteq X$, is a *chain* iff $\leq$ induces a total order on C (so, for all $a, b \in C$, either $a \leq b$ or $b \leq a$). The *strict order* (*ordering*) $<$ associated with $\leq$ is the relation defined by: $a < b$ iff $a \leq b$ and $a \neq b$. If $\leq$ is a partial order on X , we say that the pair $\langle X, \leq \rangle$ is a *partially ordered set* or for short, a *poset*.

4.1.5 Labeled directed graphs

So far, we have introduced directed graphs that can represent structures and partially ordered sets that can represent hierarchies of linguistic categories. In order to combine the two, we have to extend our definition of directed graphs to include labels (Definition 17).

The notion of isomorphism can be easily transferred to labeled directed graphs. Two labeled directed graphs are isomorphic if their structures, i. e. their vertices and edges ignoring any labels, are isomorphic and if corresponding vertices and edges are labeled identically. Formally, two labeled directed graphs $G_1 = (V_1, E_1, s_1, t_1, \langle \Sigma, \leq \rangle, \lambda_1)$ and $G_2 = (V_2, E_2, s_2, t_2, \langle \Sigma, \leq \rangle, \lambda_2)$ are isomorphic, written as $G_1 \simeq G_2$, iff

1. $(V_1, E_1, s_1, t_1) \simeq (V_2, E_2, s_2, t_2)$, with $f : (V_1, E_1, s_1, t_1) \rightarrow (V_2, E_2, s_2, t_2)$ being the isomorphism from (V_1, E_1, s_1, t_1) to (V_2, E_2, s_2, t_2) ,
2. $\lambda_1(v) = \lambda_2(f^v(v))$ for all $v \in V_1$,
3. $\lambda_1(e) = \lambda_2(f^e(e))$ for all $e \in E_1$.

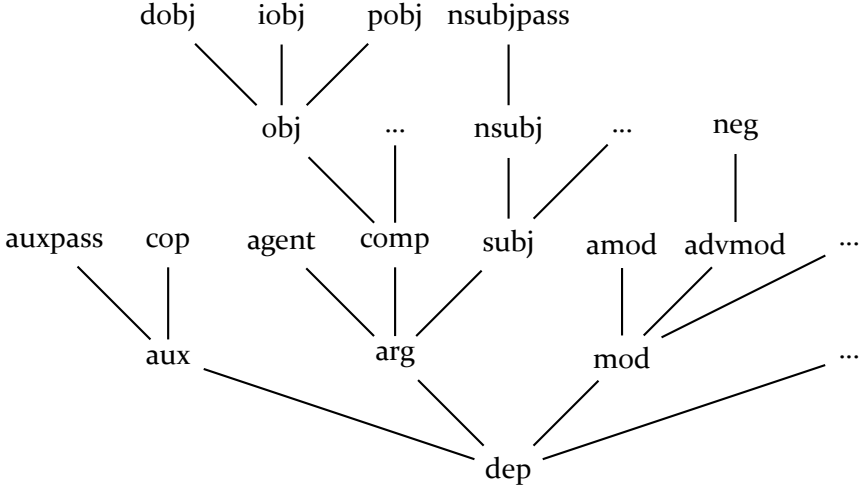


Figure 4.9: Hasse diagram of a part of the Stanford Dependencies hierarchy

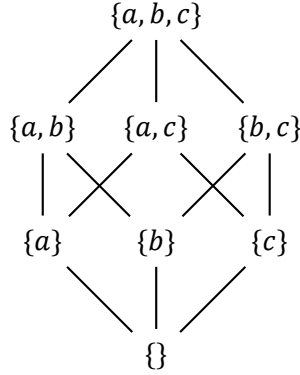


Figure 4.10: Hasse diagram of the subset ordering of subsets of $\{a, b, c\}$

Definition 17. A *labeled directed graph* is a sextuple $G = (V, E, s, t, \langle \Sigma, \leq \rangle, \lambda)$, where (V, E, s, t) is a directed graph, $\langle \Sigma, \leq \rangle$ is a partially ordered set of labels and $\lambda : V \cup E \rightarrow \Sigma$ is a function that assigns labels from Σ to the vertices and edges.

4.1.6 Subsumption

The notion of subsumption plays an important role for feature structures (Shieber, 1986: 14–16; Carpenter, 1992: 41–43; Jurafsky and Martin, 2009: 529–530; Carstensen et al., 2010: 101–102). The subsumption relation is an informational order, where a less specific feature structure subsumes a more specific (or equally specific) feature structure. That is, if feature structure F_1 subsumes feature structure F_2 but not vice versa, then F_1 is more general than F_2 . We could also say that F_1 contains a subset of the information in F_2 .

Adopting the notion of subsumption to our formalization of labeled directed graphs, we can say that a labeled digraph G_1 subsumes another labeled digraph G_2 if G_1 is structurally contained within G_2 (subgraph isomorphism) and if the labels on the vertices and edges of the corresponding subgraph of G_2 are either identical to those in G_1 or more specific. A formal definition is given in Definition 18.

Definition 18. A labeled digraph $G_1 = (V_1, E_1, s_1, t_1, \langle \Sigma, \leq \rangle, \lambda_1)$ subsumes a labeled digraph $G_2 = (V_2, E_2, s_2, t_2, \langle \Sigma, \leq \rangle, \lambda_2)$, denoted as $G_1 \sqsubseteq G_2$, iff

1. $G'_2 \subseteq (V_2, E_2, s_2, t_2)$ and $(V_1, E_1, s_1, t_1) \simeq G'_2$, with $f : (V_1, E_1, s_1, t_1) \rightarrow G'_2$ being the isomorphism from (V_1, E_1, s_1, t_1) to G'_2 ,
2. $\lambda_1(v) \leq \lambda_2(f^v(v))$ for all $v \in V_1$,
3. $\lambda_1(e) \leq \lambda_2(f^e(e))$ for all $e \in E_1$.

Intuitively, we would assume that the subsumption relation imposes an order on a given set of labeled digraphs. As shown below, that is indeed the case.

Theorem 1. *The subsumption relation $\sqsubseteq$ is a partial order on a set of labeled digraphs $\mathcal{G}$.*

Proof. We must show that $\sqsubseteq$ is reflexive, transitive and antisymmetric (cf. Definition 16).

1. (*Reflexivity*): Since G is a subgraph of itself and also isomorphic to itself, it follows that $G \sqsubseteq G$ for all $G \in \mathcal{G}$.
2. (*Transitivity*): If $G \sqsubseteq H$, then there is a $H' \subseteq (V_H, E_H, s_H, t_H)$ such that $(V_G, E_G, s_G, t_G) \simeq H'$. If $H \sqsubseteq I$, then there is a $I' \subseteq (V_I, E_I, s_I, t_I)$ such

that $(V_H, E_H, s_H, t_H) \simeq I'$. Therefore, if $G \sqsubseteq H$ and $H \sqsubseteq I$, then there is also a $I'' \sqsubseteq (V_I, E_I, s_I, t_I)$ such that $(V_G, E_G, s_G, t_G) \simeq I''$

If $G \sqsubseteq H$, then the labels in G must be more general than or equal to the corresponding labels in H . If $H \sqsubseteq I$, then the labels in H must be more general than or equal to the corresponding labels in I . Therefore, if $G \sqsubseteq H$ and $H \sqsubseteq I$, then $G \sqsubseteq I$, for all $G, H, I \in \mathcal{G}$.

3. (*Antisymmetry*): If $G \sqsubseteq H$, then there is a $H' \sqsubseteq (V_H, E_H, s_H, t_H)$ such that $(V_G, E_G, s_G, t_G) \simeq H'$. If $H \sqsubseteq G$, then there is a $G' \sqsubseteq (V_G, E_G, s_G, t_G)$ such that $(V_H, E_H, s_H, t_H) \simeq G'$. Therefore, if $G \sqsubseteq H$ and $H \sqsubseteq G$, then $(V_G, E_G, s_G, t_G) \simeq (V_H, E_H, s_H, t_H)$.

If $G \sqsubseteq H$, then the labels in G must be more general than or equal to the corresponding labels in H . If $H \sqsubseteq G$, then the labels in H must be more general than or equal to the corresponding labels in G . Therefore, if $G \sqsubseteq H$ and $H \sqsubseteq G$, then $G \simeq H$ for all $G, H \in \mathcal{G}$. This means that G and H are in the same equivalence class, i. e. $[G] = [H]$, and that for our purposes we can write $G = H$ for all $G, H \in \mathcal{G}$ (cf. Section 4.1.3).

□

The notion of subsumption will play an important role in the remainder of this chapter. If $G_1 \sqsubseteq G_2$, then we will frequently refer to the non-empty set F of subgraph isomorphisms that map the vertices and edges in G_1 to subsets of the vertices and edges in G_2 such that the constraints on the labels of G_1 and G_2 expressed in Definition 18 are satisfied. We will call those subgraph isomorphisms *embeddings*.

To give a few examples, consider the graph in Figure 4.11a, which is a labeled directed variant of the diamond graph (Weisstein, n. d.: s. v. “Diamond Graph”⁵). The labels are taken from the hierarchy of labels shown in Figure 4.11b which is the substring ordering of a subset of the substrings of the string *abcdef*. All the graphs in Figure 4.12 subsume that graph. The relevant edge in the diamond graph for the graphs in Figures 4.12a–4.12d is the one from *ab* to *f*. The labels in graph 4.12a are exactly identical, in the other three graphs, one (4.12b, 4.12c) or two (4.12d) labels are more general. The graph in Figure 4.12e is also isomorphic to a subgraph of the diamond graph and four of its seven labels are more general than the ones in the corresponding subgraph of 4.11a.

⁵ <http://mathworld.wolfram.com/DiamondGraph.html>

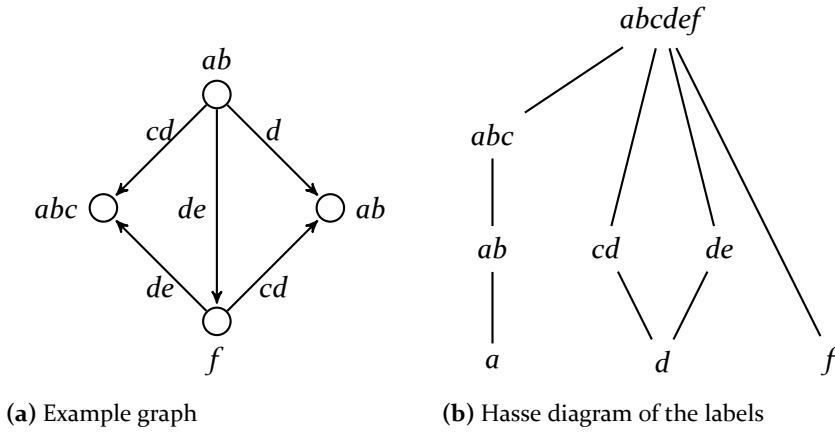


Figure 4.11: An example graph and its poset of labels

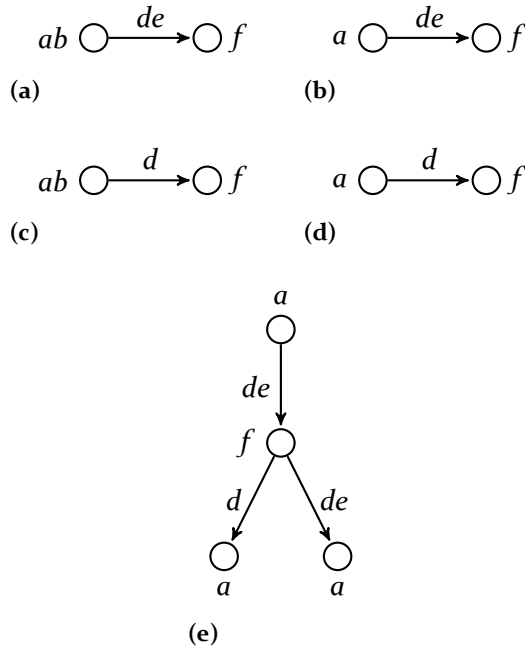


Figure 4.12: A couple of graphs subsuming the graph in Figure 4.11a

Figure 4.13 gives examples of graphs that do not subsume the diamond graph in Figure 4.11a. While the graphs in Figures 4.13a–4.13d are isomorphic to subgraphs of the diamond graph, at least one label is either more specific than in 4.11a or incomparable.

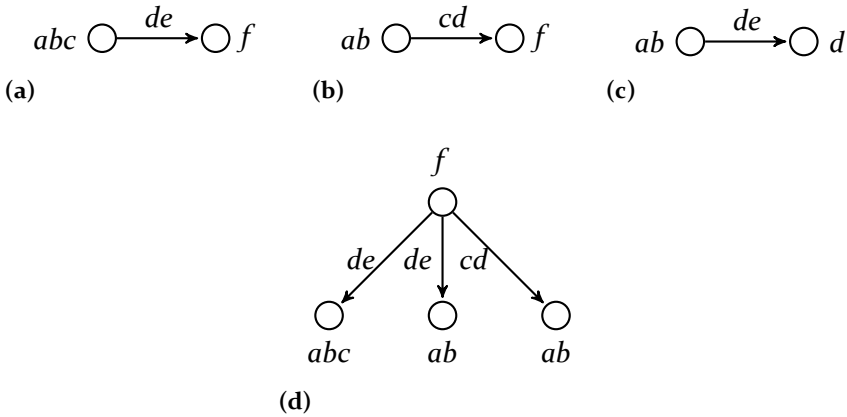


Figure 4.13: A couple of graphs not subsuming the graph in Figure 4.11a

4.1.7 Linguistic structures

Now we have all the necessary tools for representing linguistic structures. We define a linguistic structure as a labeled connected directed graph $G = (V, E, s, t, \langle \Sigma, \leq \rangle, \lambda)$. This choice does not limit our freedom to select the kinds of information to be included and to organize them in different ways. The following three examples illustrate some of the many possibilities for representing the Stanford Dependencies analysis of example sentence 6:

(6)_{BNC: J15 1406} Government attitudes towards monetary policy have undergone enormous changes since 1945.

The partially ordered set of labels could be organized as shown in Figure 4.14. To allow for a more elegant formal definition, the different “types” of labels, e. g. dependency relations, word forms or part-of-speech tags, are all contained in a single poset, although labels of different “types” are usually completely unrelated, i. e. not comparable and members of different chains.

It is worth stressing that the generalized cooccurrence model presented in this chapter does not assume any particular coding of linguistic information. It works with all graph-based representations and the following three examples are really just illustrations of some of the many possible representations.

First example: Words and dependency relations

If we are only interested in the word forms and the dependency relations between them, then we can use a straightforward representation where labeled

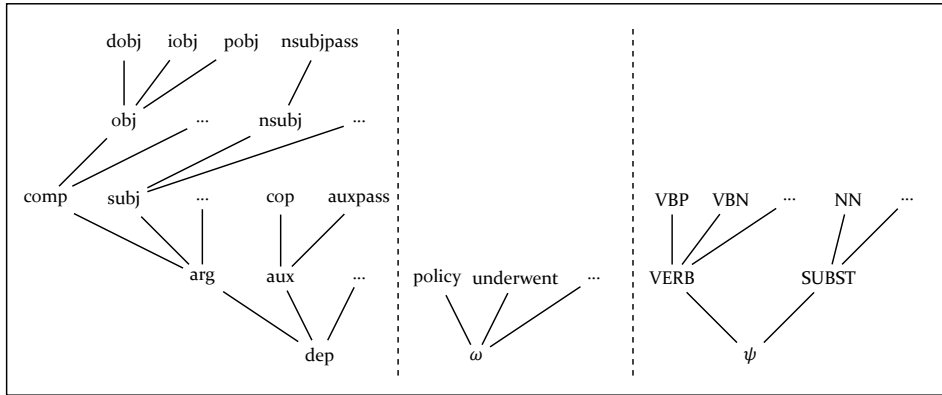


Figure 4.14: Hasse diagram of Σ

vertices represent individual words forms and labeled edges represent dependency relations (cf. Figure 4.15). The labels are simple strings.

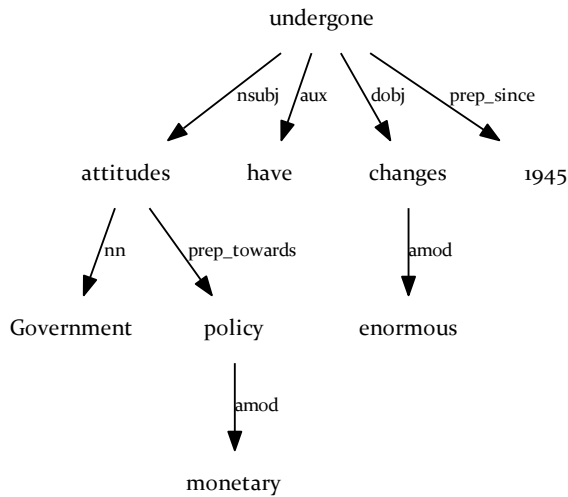


Figure 4.15: Words-and-dependencies representation of example sentence 6

Second example: Additional layers of annotation as graph structure

Sometimes it is necessary to include additional layers of annotation such as part-of-speech tags or lemmata into the representation. One possibility for

realizing such a representation is to encode the additional linguistic annotations in the graph structure. The representation in Figure 4.16 represents individual tokens as vertices labeled with the position of the token in the sentence. Dependency relations between tokens are represented as labeled edges, just as in the first example. In addition, token vertices have outgoing edges for each layer of token-based linguistic annotation, i. e. word forms and part-of-speech tags. The type of linguistic annotation is encoded in the edge labels, the annotation itself in the labels of the adjacent vertices.

Third example: Additional layers of annotation as complex labels

Another possibility for realizing a representation with additional layers of annotation is to encode those additional layers in the labels. The representation in Figure 4.17 represents individual tokens as labeled vertices and the dependency relations between them as labeled edges. The vertex labels are complex, i. e. contain several layers of annotation. In Figure 4.17, the vertex labels are realized as feature structures. As mentioned at the beginning of this chapter (p. 53), feature structures are equivalent to (acyclic) directed graphs, and therefore the two methods of representing additional layers of annotation, via graph structure or via complex labels, are equivalent.

4.2 Counting linguistic structures

The foundation of all quantitative investigation is the ability to observe and count events. Therefore, we have to establish I) what we mean when we say that a linguistic structure occurs in a larger linguistic unit, usually a sentence and II) how we can count those occurrences.

The first part is relatively straightforward as we can utilize the notion of subsumption (Section 4.1.6): We say that a linguistic structure $G_1 = (V_1, E_1, s_1, t_1, \langle \Sigma, \leq \rangle, \lambda_1)$ occurs in another linguistic structure $G_2 = (V_2, E_2, s_2, t_2, \langle \Sigma, \leq \rangle, \lambda_2)$, for example a sentence, if and only if $G_1 \sqsubseteq G_2$.

Let us illustrate that with a few example graphs. The partially ordered set of labels Σ for this and most of the following examples in this chapter is shown in Figure 4.18. The poset contains the four edge labels a, b, c, d and the general edge label ϵ as well as the four vertex labels r, s, t, u and the general vertex label v (lower case Greek upsilon). As a first example, consider the graphs in Figure 4.19. As we can see, G_1 (Figure 4.19a) is isomorphic to a subgraph of G_3 (Figure 4.19c; corresponding vertices and edges are set in bold) and therefore the relation $G_1 \sqsubseteq G_3$ holds. G_2 (Figure 4.19b) is not isomorphic

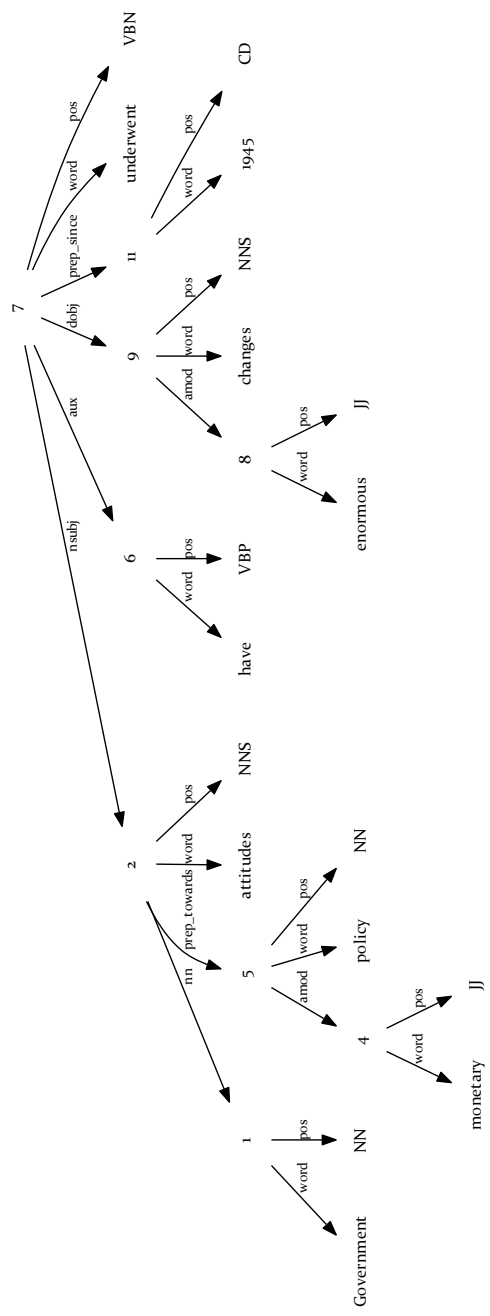


Figure 4.16: Representation of example sentence 6 with annotation as graph structure

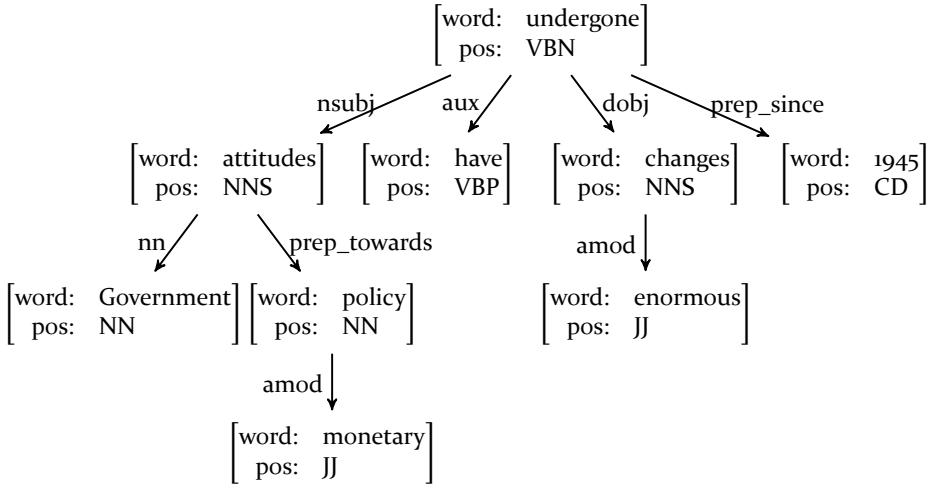


Figure 4.17: Representation of example sentence 6 with annotation as complex labels

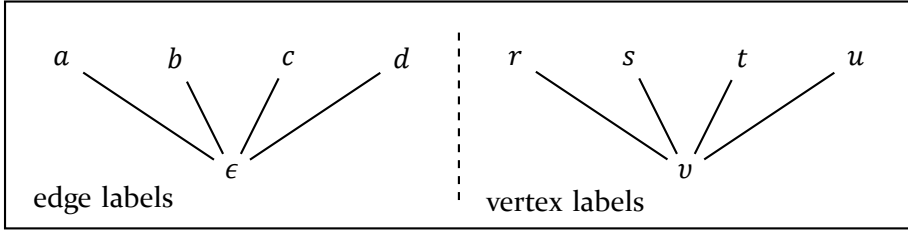


Figure 4.18: Example Σ for demonstration purposes

to a subgraph of G_3 (because the labels differ) but as ϵ is more general than a and v is more general than r , the relation $G_2 \sqsubseteq G_3$ still holds and we say that the linguistic structure expressed in G_2 occurs in G_3 .

The second part, counting the occurrences of a linguistic structure, i. e. determining its frequency, is less straightforward. Regarding Figures 4.19a and 4.19c it is obvious that G_1 occurs once in G_3 ; but looking at the example graphs in Figures 4.20a and 4.20b it is not so clear what the frequency of H_1 in H_2 should be. To clarify the discussion, the vertices in both graphs have been given unique IDs. Vertex 1, labeled r in H_1 , can correspond to vertices 4 or 9, both labeled r in H_2 . If vertex 1 corresponds to vertex 4, then vertex 2 (v) can correspond to vertex 5 (t) and vertex 3 (t) to vertex 7 (t), or vice versa. If vertex 1 corresponds to vertex 9, then vertex 2 can correspond to vertex 10 (t) and vertex 3 to vertex 11 (t), or vice versa. In other words, there are four

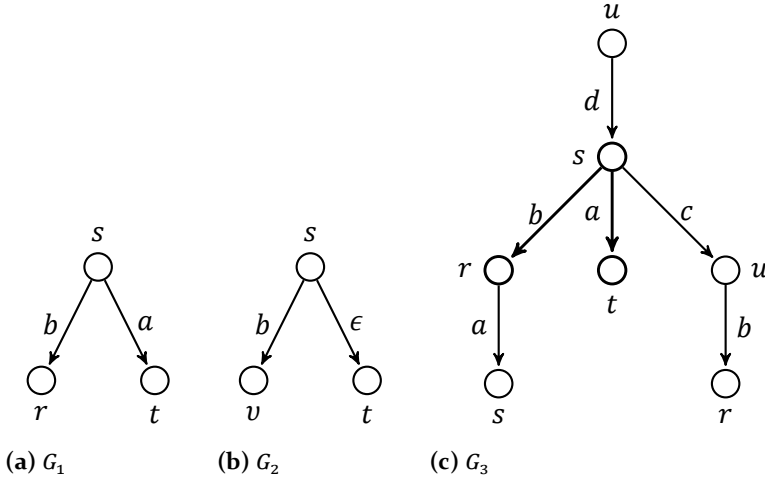


Figure 4.19: G_1 and G_2 occur in G_3

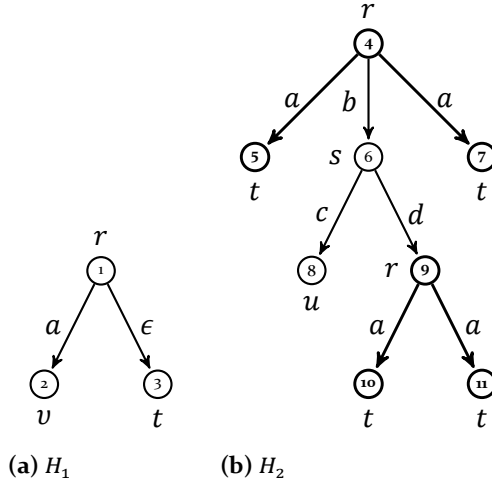


Figure 4.20: What is the frequency of H_1 in H_2 ?

distinct embeddings that relate H_1 and H_2 . Therefore, we could argue that the frequency of H_1 in H_2 is 4. However, those four embeddings correspond to just two subgraphs of H_2 , the subgraphs induced by the vertex sets $\{4, 5, 7\}$ and $\{9, 10, 11\}$. The greater number of embeddings stems from the symmetry of those two subgraphs. Therefore, we could also argue that the frequency of H_1 in H_2 is 2. We could also argue that both subgraphs occur in the same

graph, H_2 , and that we only want to know the number of graphs in which H_1 occurs. Then the frequency of H_1 would be 1. A final perspective could be to interpret the linguistic structure H_1 as constraints on the root vertex r , i. e. we are interested in instances of r that have an outgoing edge labeled a and a second outgoing edge to a vertex labeled t . In that case the frequency of H_1 in H_2 would be 2 since there are two vertices in H_2 that satisfy the constraints on r expressed in H_1 .

Those four variants are all well motivated and, for the purpose of counting *occurrences*, all of them are equally justified (while of course the usefulness of a variant in a given situation depends on the perspective of the researcher). In the following sections, we formalize those methods for determining the frequency of a linguistic structure and introduce a model that can abstract away from the individual counting methods. A central question in the remainder of this work will be to find out which counting method is most appropriate for the analysis of *cooccurrences* of linguistic structures (Sections 4.4.1–4.4.4, Chapter 5).

4.2.1 Counting embeddings

Counting embeddings is a straightforward way of determining the frequency of a linguistic structure $G_1 = (V_1, E_1, s_1, t_1, \langle \Sigma, \leq \rangle, \lambda_1)$ within another linguistic structure $G_2 = (V_2, E_2, s_2, t_2, \langle \Sigma, \leq \rangle, \lambda_2)$. Recall that if $G_1 \sqsubseteq G_2$, then there exists a non-empty set F of isomorphisms that map the vertices and edges in G_1 to subsets of the vertices and edges in G_2 such that the constraints on the labels of G_1 and G_2 expressed in Definition 18 are satisfied. We refer to those subgraph isomorphisms as embeddings (cf. Section 4.1.6). The embedding frequency of G_1 in G_2 is simply the number of embeddings:

$$\text{embedding frequency}_{\text{local}}(G_1, G_2) = |F|$$

The frequency of G_1 within a corpus C , i. e. a multiset of linguistic structures (sentences), is the sum of the frequencies of G_1 in the individual sentences:

$$\text{embedding frequency}_{\text{global}}(G_1, C) = \sum_{G_s \in C} \text{embedding frequency}_{\text{local}}(G_1, G_s)$$

4.2.2 Counting subgraphs

Another option, if we are not interested in the number of possible mappings from G_1 to G_2 , is to merely count the number of subgraphs of G_2 involved. For formalizing that method of counting, we need to define the set $\mathcal{P}(G_2)$, the set

of all possible subgraphs of G_2 (also called the *power set of G_2* ; cf. Gallier, 2011: 73 for a formulation of the power set axiom), whose members are exactly the subgraphs of G_2 ; that is:

$$G'_2 \in \mathcal{P}(G_2) \text{ iff } G'_2 \text{ is a subgraph of } G_2.$$

Further, we need to define a total function $g : F \rightarrow \mathcal{P}(G_2)$ that maps every embedding $f \in F$ of G_1 in G_2 to a subgraph $G'_2 \in \mathcal{P}(G_2)$; recall that an embedding, i. e. isomorphism, is a pair $f = (f^v, e^v)$ (cf. Section 4.1.3):

$$g(f) = G'_2 = (V'_2, E'_2, s'_2, t'_2, \langle \Sigma, \leq \rangle, \lambda'_2), \text{ where } G'_2 \text{ is the subgraph of } G_2 \text{ induced by } V'_2 = \text{range}(f^v) \text{ and } E'_2 = \text{range}(f^e).$$

The subgraph frequency of G_1 in G_2 is the number of subgraphs in $\text{range}(g)$:

$$\text{subgraph frequency}_{\text{local}}(G_1, G_2) = |\text{range}(g)|$$

Again, the frequency of G_1 within a corpus C is the sum of the frequencies of G_1 in the individual sentences:

$$\text{subgraph frequency}_{\text{global}}(G_1, C) = \sum_{G_s \in C} \text{subgraph frequency}_{\text{local}}(G_1, G_s)$$

4.2.3 Counting graphs (sentences)

Another straightforward way of determining the frequency of a linguistic structure G_1 in a corpus is to count in how many sentences it occurs, i. e. how many sentences it subsumes:

$$\text{graph frequency}_{\text{global}}(G_1, C) = |\{G_s \in C \mid G_1 \sqsubseteq G_s\}|$$

The graph frequency of G_1 in G_2 is either 1 or 0 (irrespective of any multiple occurrences according to the other counting methods):

$$\text{graph frequency}_{\text{local}}(G_1, G_2) = \begin{cases} 1 & \text{if } G_1 \sqsubseteq G_2 \\ 0 & \text{if } G_1 \not\sqsubseteq G_2 \end{cases}$$

4.2.4 Counting focus points

The last counting method is concerned with the frequency of special vertices that conceptually play a prominent role. Consider the frequency of G_1 in G_2 as given in Figure 4.21. There are three embeddings of G_1 in G_2 and

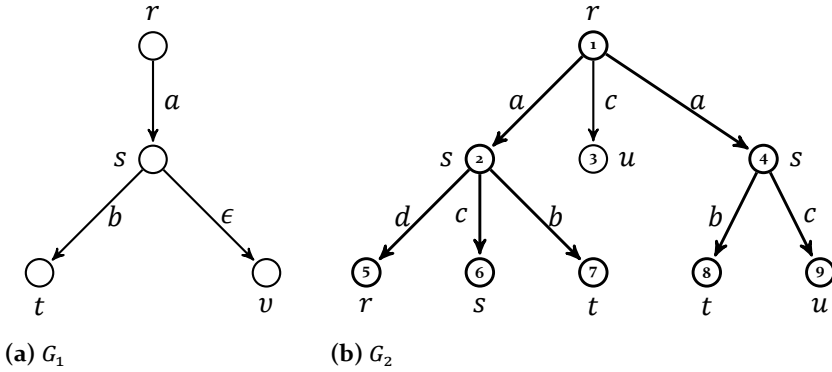


Figure 4.21: What is the frequency of G_1 in G_2 ?

three corresponding subgraphs of G_2 (induced by the vertex sets $\{1, 2, 7, 5\}$, $\{1, 2, 7, 6\}$ and $\{1, 4, 8, 9\}$). Let us assume that the vertex labeled r plays an important conceptual role in the linguistic structure we are looking for; maybe it is the verb that governs the whole structure. Now what we are really interested in is how many instances of r there are in G_2 that occur in the structure specified in G_1 . The answer to that question is 1. On the other hand, if we assume that the vertex labeled s is the most prominent vertex in G_1 (maybe it is a noun occurring in a specific structure), then we would be interested in the number of instances of s in G_2 that occur in the structure specified in G_1 . That frequency is 2.

For the last counting method we focus on conceptually prominent vertices and we refer to them as focus point vertices. Obviously, the question which vertex in a given query structure is the conceptually most prominent one that is to be treated as the focus point vertex depends not only on the query structure but also on the research question that lies behind the query. Therefore, this is a decision that has to be made on a case-by-case basis. Nevertheless, there are vertices that exhibit structural properties that make them particularly promising candidates. At the end of this section, we will make an attempt at automatically determining such potentially suitable focus point vertices.

While it is certainly possible and sometimes even desirable to specify more than one focus point vertex for a given query structure, we will restrict ourselves to a single focus point vertex here for two reasons: 1) If there is more than one focus point vertex, then we are effectively determining the frequency of another, smaller linguistic structure and it is unclear how we should best do that. We would have to choose an appropriate counting method out

of the four presented in this section. 11) As we will see in Section 4.4, the limitation to a single focus point vertex is an effective countermeasure against the problem of inflated frequencies that can occur with some of the other counting methods.

When we count focus point vertices, we do neither count all embeddings nor all subgraphs corresponding to those embeddings. What we are interested in is the number of different embeddings with respect to the focus point vertex. That method of counting seems to be particularly intuitive for linguistic applications where we are often interested in phenomena like verbs with outgoing subject or object relations, or nouns with outgoing modifier relations. Intuitively, it makes sense to count only distinct matches of those verb or noun vertices, i. e. to count each matching verb or noun only once.⁶

Given a focus point vertex $u \in V_1$ of G_1 and using the set F of embeddings of G_1 in G_2 , we can determine the focus point frequency of G_1 in G_2 :

$$\text{focus point frequency}_{\text{local}}(G_1, u, G_2) = |\{f^v(u) | f^v \in F\}|$$

The frequency of G_1 within a corpus C is the sum of the frequencies of G_1 in the individual sentences:

$$\text{focus point frequency}_{\text{global}}(G_1, u, C) = \sum_{G_s \in C} \text{focus point frequency}_{\text{local}}(G_1, u, G_s)$$

Automatic determination of focus point vertices

As argued above, the focus point vertex should ideally be selected by the user because it depends not only on the individual query structure but also on the underlying research question. Nevertheless, some vertices are structurally more prominent and might make good focus points. In the motivating examples given above (verbs with outgoing subject or object relations and nouns with outgoing modifier relations), the verb or noun vertices were the root vertices of the query graphs. The root vertex of a graph can be determined automatically (if it exists) and is certainly highly prominent structurally. Syntactic analyses using the Stanford Dependencies are not necessarily trees, but they are rooted (cf. de Marneffe and Manning, 2015: 17–18). Most queries in the realm of syntax will probably also be rooted graphs.

⁶ If we apply that counting method to relational cooccurrences, we have no frequency counts greater than one for single word occurrences in the corpus and we avoid inflated frequency counts for examples such as “a beautiful, beautiful, beautiful speech” (cf. Evert, 2004: 35).

Those roots will mostly be verbs or nouns, i. e. words that are very likely to be of linguistic interest.

We can generalize the related notions of root and antiroot vertices to that of choke point vertices (cf. Figure 4.22 for a few examples):⁷

Definition 19 (cf. Gallier, 2011: 193). Given a digraph $G = (V, E, s, t)$, a node $a \in V$ is a *root* (respectively, *antiroot*) iff for every node $u \in V$ there is a path from a to u (respectively, there is a path from u to a). [...]

Definition 20. Given a digraph $G = (V, E, s, t)$, a vertex $a \in V$ is a *choke point vertex*, iff for every node $u \in V$ there is either a path from a to u or a path from u to a .

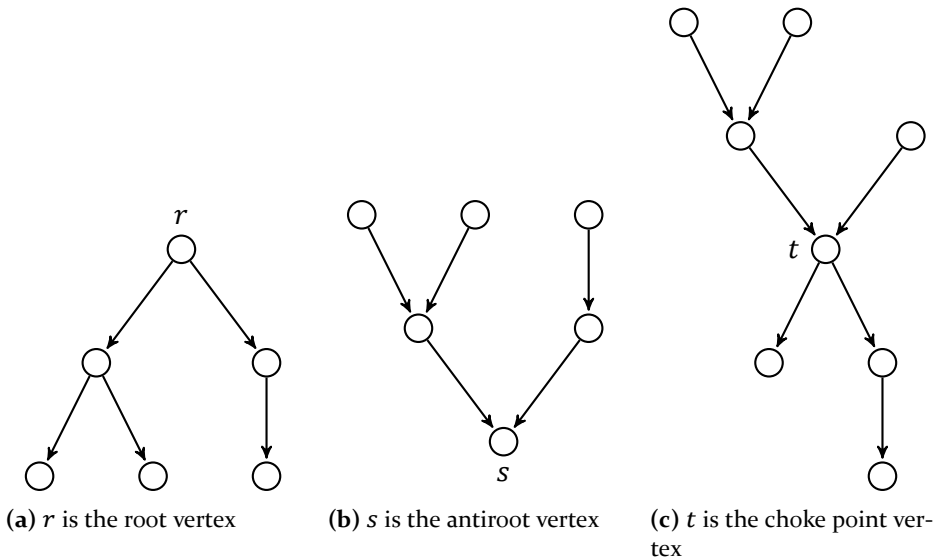


Figure 4.22: Examples of focus point vertices

Choke point vertices can be determined automatically and due to their special structural properties they are very likely to be of linguistic interest. If there

⁷ Choke point vertices are very similar to articulation points or articulation vertices, i. e. vertices whose removal will disconnect a graph (Weisstein, n. d.: s. v. “Articulation Vertex”, <http://mathworld.wolfram.com/ArticulationVertex.html>). Root and antiroot vertices are not necessarily articulation vertices, though.

is more than one choke point vertex in a graph, we can devise a ranking scheme that prioritizes, for example, roots over antiroots over other choke point vertices.

4.2.5 A generalized perspective on counting methods

All methods introduced above for determining the frequency of a linguistic structure G_1 in another linguistic structure G_2 are based on the notion of subsumption and the embeddings of G_1 in G_2 . They either count the number of embeddings directly or they group several embeddings together according to some criteria and count the number of groups. We can formalize that grouping together of embeddings as a partition of the set of embeddings of G_1 in G_2 (Definition 21). A partition of a set is characterized by the fact that all elements of the set are members of exactly one block and that no block is empty.

Definition 21 (adapted from Gallier, 2011: 293). Given a set X , a *partition* of X is a set $\Pi = \{B_i | i \in I\}$ of subsets of X such that

1. $B_i \neq \emptyset$, for all $i \in I$ (each B_i is nonempty)
2. If $i \neq j$ then $B_i \cap B_j = \emptyset$ (the B_i are pairwise disjoint)
3. $X = \cup\{B_i | i \in I\}$ (the union of the sets in Π is equal to X).

Each set B_i is called a *block* of the partition.

If F is the set of embeddings of G_1 in G_2 , the four counting methods partition F as follows:

Embeddings: Every embedding is a single block in the partition: $P = \{\{f\} | f \in F\}$.

Subgraphs: Embeddings that correspond to the same subgraph of G_2 are grouped together: $P = \{\{f | f \in F \text{ and } g(f) = G'_2\} | G'_2 \in \{g(f) | f \in F\}\}$, where g is the function defined in Section 4.2.2 that maps every embedding in F to a subgraph of G_2 .

Graphs: All embeddings are in the same block: $P = \{F\}$.

Focus point vertices: All embeddings that map the focus point vertex $u \in V_1$ to the same vertex $v \in V_2$ are grouped together: $P = \{\{f | f \in F \text{ and } f^v(u) = v\} | v \in \{f^v(u) | f \in F\}\}$.

Using that formalization and the set F of embeddings of G_1 in G_2 , we can say that the frequency of G_1 in G_2 is the number of blocks of the partition $P = \{F_i\}_{i \in I}$ of F that is determined by the specific counting method:

$$\text{frequency}_{\text{local}}(G_1, G_2) = |P|$$

The individual “matches” are then the blocks F_i of that partition. The frequency of G_1 within a corpus C is the sum of the frequencies of G_1 in the individual sentences:

$$\text{frequency}_{\text{global}}(G_1, C) = \sum_{G_s \in C} \text{frequency}_{\text{local}}(G_1, G_s)$$

4.3 A preliminary model based on relational cooccurrences

As an intermediate step towards the generalized cooccurrence model that will be introduced in Section 4.5, we will first take another look at relational cooccurrences (Section 4.3.1) and covarying collexeme analysis (Section 4.3.2). We will show that both phenomena (cooccurrence of two words in a single relation and cooccurrence of two words in an arbitrary linguistic structure) can be interpreted as the cooccurrence of two isomorphic, almost identically labeled linguistic structures that only differ in two word labels. Based on that insight, we will generalize the notion of relational cooccurrence and introduce a formal model that allows for the cooccurrence of two arbitrary linguistic structures within an arbitrary linguistic structure (Section 4.3.3).

4.3.1 Relational cooccurrences revisited

The conventional perspective on relational cooccurrences (cf. Section 2.1.2) is that two word forms or lemmata are connected via a direct binary relation. For the sake of the argument, let us assume that those relations are dependency relations in which a word can either be the governor or the dependent. Then, the unit of analysis is the dependency relation with the two words it connects. The sample size N is the number of dependency relations in the corpus and these dependency relations can be cross-classified according to the following two criteria: “Is word A the governor?” and “Is word B the dependent?”. The resulting cross-classification can be represented in the form of a contingency table (cf. Section 3.2). It is also possible to focus on special dependency relations, e. g. only direct or indirect objects, modifier relations, etc. In that case, we only cross-classify the relevant dependency relations, e. g. all object

relations, and the sample size N is the number of those dependency relations. Either way, instances of A without outgoing dependency relation will be ignored and instances of A with multiple outgoing dependency relations will be counted multiple times (the same is true for B and incoming dependency relations).

Let us assume we are interested in the association strength of word r governing word s via dependency relation a , as shown in G_1 (Figure 4.23a). We have to extract all a relations from the corpus (with arbitrary governors and dependents), cf. G_2 (Figure 4.23b), and cross-classify them. Therefore, the sample size N corresponds to the frequency of G_2 in the corpus. The cooccurrence frequency O_{11} , i. e. the number of instances of G_2 where the governor is r and the dependent is s , corresponds to the frequency of G_1 in the corpus. We also give graph representations for the marginal frequencies R_1 and C_1 : G_3 (Figure 4.23c) represents all instances of G_2 where the governor is r and G_4 (Figure 4.23d) represents all instances of G_2 where the dependent is s .

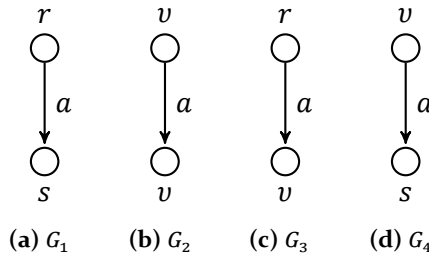


Figure 4.23: Query graphs for relational cooccurrences

This leads us to another perspective on relational cooccurrences. Returning to the above example, we can also say that we are interested in the cooccurrence of two very simple syntactic structures, r with an outgoing dependency relation a (G_A , Figure 4.24a) and s with an incoming dependency relation a (G_B , Figure 4.24b). The dependent of r and the governor of s are underspecified (in this case not specified at all). The two structures should cooccur in such a way, that they completely overlap, i. e. that they cooccur in another syntactic structure that is structurally identical, i. e. isomorphic, to both of them and in which the underspecified labels are replaced with their more specific counterparts (G_C , Figure 4.24c). Or, still more explicit, we have I) r governing another word via a dependency relation a and we have II) s being governed by another word via a dependency relation a and we are interested in their cooccurring such that the governor of structure I is the governor

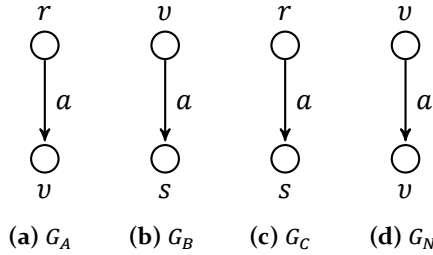


Figure 4.24: G_A and G_B cooccur in G_C

of structure II and that the dependent of structure II is the dependent of structure I.

That means we can interpret relational cooccurrences as cooccurrences of two (very simple) syntactic structures G_A and G_B that consist of one dependency relation and two words. The two structures are isomorphic, have the same dependency relation and only differ in their word labels – in the one structure the governor is underspecified and in the other the dependent. The two structures cooccur in such a way that they completely overlap, thus forming another, more specific syntactic structure G_C where none of the words are underspecified. The cooccurrence frequency of the two structures corresponds to the frequency of that structure G_C . For determining the marginal frequencies, we can directly use G_A and G_B . The sample size N can be obtained with the help of a structure that is isomorphic to G_C . In that structure, G_N (Figure 4.23b), the two more specific word labels from G_A and G_B have been replaced with their more general counterparts from the other structure (*any word*). Note that the resulting structures G_A , G_B , G_C and G_N are identical to G_3 , G_4 , G_1 and G_2 in Figure 4.23, i. e. our interpretation of relational cooccurrences leads to exactly the same frequencies as the original conception.

4.3.2 Covarying collexeme analysis revisited

In the previous section, we showed that relational cooccurrences can be interpreted as cooccurrences of two simple syntactic structures consisting of one dependency relation each. The dependency relation is identical in both structures, the two only differ in two vertex labels.

If we generalize that view and allow the two structures to be arbitrary syntactic structures, as long as they are structurally identical and only differ in two vertex labels, we have a reconstruction of covarying collexeme analysis

(cf. Section 2.1.2). In the analysis of relational cooccurrences we analyze two words cooccurring in a single dependency relation and in the analysis of covarying collexemes we analyze two words cooccurring in an arbitrary syntactic structure. The only difference is that we allow arbitrary structures instead of single dependency relations.

If, for example, we are interested in the cooccurrence of s and t in a structure reminiscent of the notorious ditransitive construction such as shown in G_C (Figure 4.25c), we can interpret that as the cooccurrence of the two linguistic structures G_A (Figure 4.25a) and G_B (Figure 4.25b) such that the two completely overlap and that instead of the general v labels the more specific s and t labels are used.

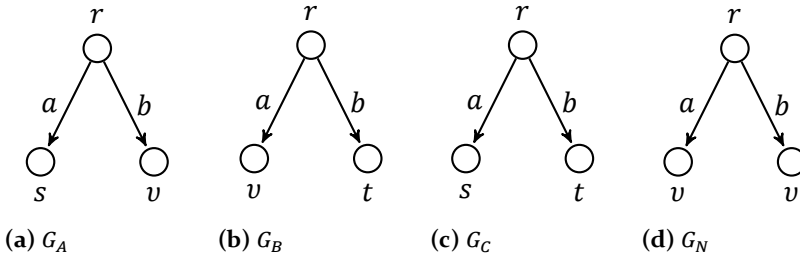


Figure 4.25: Query graphs for covarying collexeme analysis

As for relational cooccurrences, the sample size N can be determined via the frequency of the linguistic structure G_N (Figure 4.25d) that is derived from G_C by replacing the two more specific labels s and t in which G_A and G_B differ with their more general counterparts v from the other linguistic structure. That means labels that are the same in both cooccurring structures are preserved, the other (two) labels are generalized. The marginal frequency R_1 directly corresponds to the frequency of G_A , i. e. the number of instances of G_N where the dependent of the a relation is s . The marginal frequency C_1 directly corresponds to the frequency of G_B , i. e. the number of instances of G_N where the dependent of the b relation is t . The cooccurrence frequency O_{11} corresponds to the frequency of G_C , i. e. the number of instances of G_N where the dependent of the a relation is s and dependent of the b relation is t . Directly determining the frequency signature of the cooccurrence that way is completely equivalent to cross-classifying all instances of G_N .⁸

⁸ But bear in mind the problems mentioned in Section 2.2.1 that relate to the difficulty of determining the frequencies of certain structures. Cf. also Sections 4.2 and 4.4.

4.3.3 Generalization to arbitrary linguistic structures

Up to now, we have seen that covarying collexeme analysis can be interpreted as a generalization of relational cooccurrences. We have been looking at the cooccurrence of two words within a linguistic structure. For relational cooccurrences, this structure has to be a single dependency relation, for covarying collexeme analysis, this restriction is lifted and we can have an arbitrary linguistic structure. But we are still limited to pairs of words.⁹

It only seems logical to also lift that limitation and to further generalize the approach. Instead of merely analyzing the cooccurrence of two single labels within a linguistic structure, we will now extend the approach to allow for two collections of labels.

Suppose we are interested in the adjectives occurring in the structure *make life* ADJ and we want to determine the association between *make life* and the adjective *miserable*. This is something we cannot do with covarying collexeme analysis – we could only determine either the association between *life* and *miserable*, regarding *make* as part of the fixed linguistic structure, or the association between *make* and *miserable*, regarding *life* as part of the fixed linguistic structure. If we take our interpretation of relational cooccurrences and covarying collexeme analysis one step further, we can interpret *make life miserable* as the cooccurrence of the two structures *make life* ADJ and V N *miserable* and we can easily determine the association between the two by cross-classifying the instances of V N ADJ.

Basically, the idea is to allow for the cooccurrence of two arbitrary linguistic structures within an arbitrary linguistic structure and to interpret that as the cooccurrence of two linguistic structures that are structurally identical (i. e. isomorphic when ignoring the labels) and differ in an arbitrary number of vertex and edge labels.¹⁰ To put it another way: With the methods of relational cooccurrences and covarying collexeme analysis, we can determine the association strength between two individual vertex labels within an arbitrary linguistic structure. With the generalization, we can determine the association strength between two collections of labels within an arbitrary linguistic structure.

As an example, consider the graph in Figure 4.25d: If we want to measure the association between two words within that linguistic structure, we

⁹ The emphasis is on “words” and not on “pairs”. As mentioned in Section 3.4.3, there are extensions to some association measures that can deal with an arbitrary number of words cooccurring within a linguistic structure, e. g. Sangati and van Cranenburgh (2015).

¹⁰ The requirement of structural identity will be relaxed in the final, generalized cooccurrence model (cf. Section 4.5).

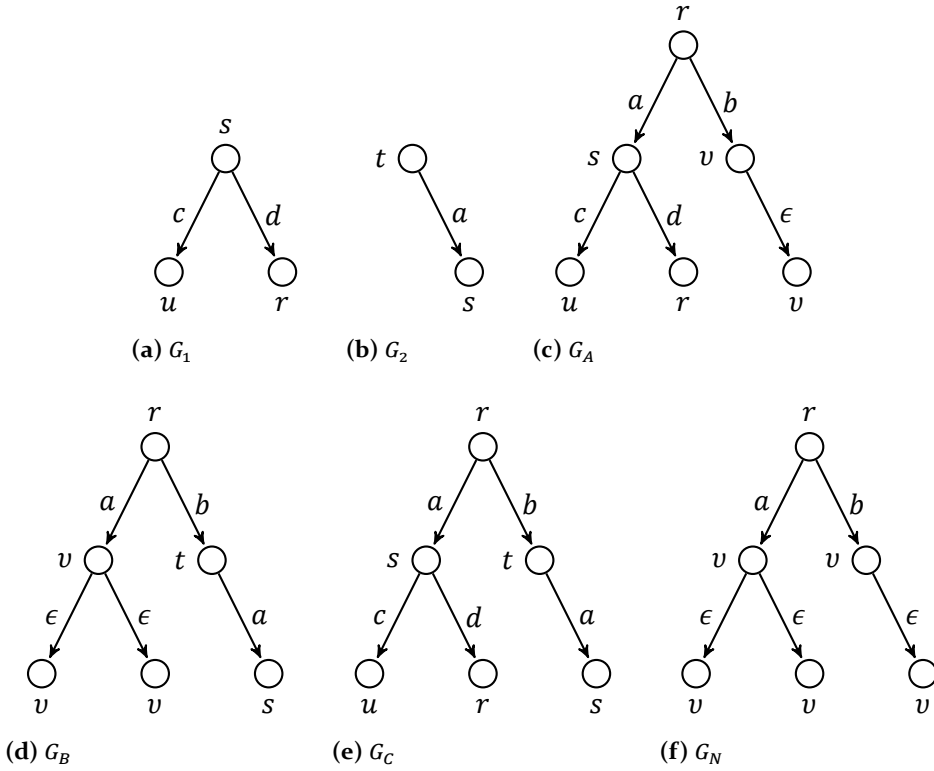


Figure 4.26: Two arbitrary linguistic structures G_1 and G_2 cooccur within an arbitrary linguistic structure as shown in G_C . That cooccurrence can be interpreted as the cooccurrence of G_A and G_B .

replace the v labels by more specific ones. If we are interested in the association between two other structures G_1 and G_2 cooccurring within that structure, we can replace the underspecified vertices with the structures in Figures 4.26a and 4.26b, yielding the linguistic structure G_C in Figure 4.26e. We can interpret G_C as the cooccurrence of G_A and G_B (Figures 4.26c and 4.26d).

The general approach remains the same as for relational cooccurrences and covarying collexeme analysis: G_C specifies how exactly the two linguistic structures G_A and G_B cooccur and G_N characterizes the units of analysis. G_N can be obtained from G_C by replacing all vertex and edge labels that are more specific in one of the two original structures by the more general ones from the other structure (usually *any word* and *any relation*). The sample size N corresponds to the frequency of G_N , the marginal frequencies R_1 and C_1 to

the frequencies of G_A and G_B , respectively, and the cooccurrence frequency O_{11} to the frequency of G_C .

The approach informally outlined above can be formally defined in terms of the graph-theoretical representation of linguistic structures developed in Section 4.1. What we are interested in is the cooccurrence of two linguistic structures G_A and G_B in a specific configuration. Usually, two linguistic structures can cooccur in more than one way, therefore our starting point is to specify the exact linguistic structure G_C we are looking for and how exactly the two cooccurring structures G_A and G_B relate to G_C :

$G_C = (V_C, E_C, s_C, t_C, \langle \Sigma, \leq \rangle, \lambda_C)$ is a linguistic structure in which linguistic structure $G_A = (V_A, E_A, s_A, t_A, \langle \Sigma, \leq \rangle, \lambda_A)$ cooccurs with linguistic structure $G_B = (V_B, E_B, s_B, t_B, \langle \Sigma, \leq \rangle, \lambda_B)$ such that the following constraints hold:

1. (V_A, E_A, s_A, t_A) is isomorphic to (V_C, E_C, s_C, t_C) and the isomorphism $f_A : G_A \rightarrow G_C$ from G_A to G_C specifies which elements of the two graphs correspond to each other.
2. (V_B, E_B, s_B, t_B) is isomorphic to (V_C, E_C, s_C, t_C) and the isomorphism $f_B : G_B \rightarrow G_C$ from G_B to G_C specifies which elements of the two graphs correspond to each other.
3. $\lambda_C(x) = \max(\lambda_A(f_A^{-1}(x)), \lambda_B(f_B^{-1}(x)))$.

This means that G_A , G_B and G_C are structurally identical and that the vertices and edges in G_C are labeled with the more specific label for that element from G_A and G_B . Of course, a prerequisite for that is that $\lambda_A(f_A^{-1}(x))$ and $\lambda_B(f_B^{-1}(x))$, the labels for corresponding elements in G_A and G_B , are comparable. Once the relationship between G_A , G_B and G_C has been defined, G_N can be derived as follows.

$G_N = (V_N, E_N, s_N, t_N, \langle \Sigma, \leq \rangle, \lambda_N)$ is a linguistic structure where (V_N, E_N, s_N, t_N) is isomorphic to (V_C, E_C, s_C, t_C) , the isomorphism $f_N : G_N \rightarrow G_C$ from G_N to G_C specifies which elements of the two graphs correspond to each other and $\lambda_N(x) = \min(\lambda_A(f_A^{-1}(f_N(x))), \lambda_B(f_B^{-1}(f_N(x))))$.¹¹

This means that G_N is structurally identical to G_A , G_B and G_C and that its vertices and edges are labeled with the more general label for that element

¹¹ It might not be entirely obvious why we go to these great pains to define the labels of G_N like this instead of simply using the most general labels on all vertices and edges – after all it would just increase the amount of seemingly irrelevant material that counts toward O_{22} in a contingency table. The reason we want G_N to be as similar to G_A and G_B as possible is that this seemingly irrelevant material can have quite some impact on some association measures (cf. Section 4.4.4 and especially Footnote 16 on p. 106) and we want to make sure that the sample size is as sensible as possible.

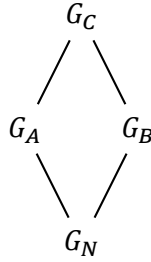


Figure 4.27: Hasse diagram of the linguistic structures

from G_A and G_B . As a result, the following relations (visualized in the Hasse diagram in Figure 4.27) always hold: $G_N \sqsubseteq G_A$, $G_N \sqsubseteq G_B$, $G_A \sqsubseteq G_C$ and $G_B \sqsubseteq G_C$. Since the directed graphs in G_A and G_B are isomorphic, it may also happen in some cases that $G_A \sqsubseteq G_B$ or $G_B \sqsubseteq G_A$.

Just as for the special cases of that model, relational cooccurrences and covarying collexeme analysis, the frequencies O_{11} , R_1 , C_1 and N correspond to the frequencies of the linguistic structures G_C , G_A , G_B and G_N . The model itself is agnostic about the counting method applied but each method introduced in Section 4.2 has its advantages and disadvantages that will be discussed in the following section.

4.4 Counting and cross-classifying cooccurrences

Determining the frequency of single vertices or words is a trivial task. If we are dealing with relational cooccurrences, there are already some not-so-easy cases. It can well happen that a single instance of a word has a frequency greater than one as a component of a relational pair. Consider the exclamation “A beautiful, beautiful, beautiful speech!” in example sentence 7:

(7)_{BNC: HR9_928} He stood back, a tight smile on his face, as Roger de Mornay said, ‘A beautiful, beautiful, beautiful speech!’

There is only a single instance of *speech*, but if we are counting adjective-noun pairs, we find three instances of *beautiful speech* (cf. Figure 4.28). This corresponds to the frequencies obtained by counting embeddings or subgraphs for that structure and those inflated frequencies might be regarded as a problem (cf. Evert, 2004: 35). Note that the suggested remedy for that situation is to count only a single instance – the same happens automatically if we count automatically determined focus point vertices, i. e. choke point vertices:

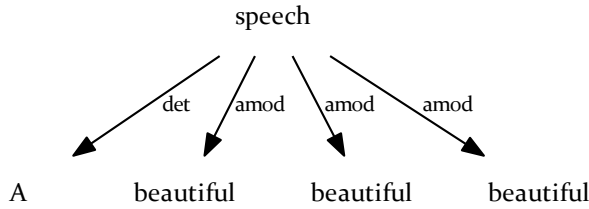


Figure 4.28: SD analysis of a part of example sentence 7

For optimal results, the repetition of the adjective should be identified as a rhetoric device during cooccurrence extraction (or syntactic pre-processing) and be replaced by a single adjective-noun relation (perhaps with an annotation indicating the rhetoric effect).

Evert (2004: 35)

The preliminary cooccurrence model outlined in Section 4.3 allows for much more complex structures than simple pairs of words, therefore the problem of inflated frequency counts might be more serious. In the following subsections, we will analyze the counting methods presented in Section 4.2 with respect to that problem.

Additionally, we will be concerned with the consistency of the frequency counts obtained by each counting method. As described in Section 3.2, cooccurrence data are analyzed with the help of contingency tables in which the members of a sample are cross-classified. In our case, the members of the sample are the instances of G_N and each instance can either constitute an occurrence of G_A , G_B , both or none of them. And while we have a clear definition of what constitutes a cooccurrence of linguistic structures G_A and G_B , i. e. an occurrence of G_C , we will see that there can be some problematic cases in which we have an occurrence of G_A and an occurrence of G_B but not a cooccurrence of the two, i. e. no occurrence of G_C .

4.4.1 Counting embeddings

A corpus $\mathcal{C} = \{G_1, G_2, \dots, G_{S-1}, G_S\}$ is a multiset of S linguistic structures. For each linguistic structure $G_S \in \mathcal{C}$ there exists a (possibly empty) set $F_N^{(s)}$ of embeddings of G_N in G_S . Each of the embeddings in $F_N^{(s)}$ is a mapping from the vertices and edges of G_N to a subset of the vertices and edges of G_S . There also exist the (also possibly empty) sets of embeddings $F_A^{(s)}$ of G_A in G_S , $F_B^{(s)}$ of G_B in G_S and $F_C^{(s)}$ of G_C in G_S . Given the subsumption relations between

the linguistic structures shown in Figure 4.27, we can infer that $F_C^{(s)} \subseteq F_A^{(s)}$, $F_C^{(s)} \subseteq F_B^{(s)}$, $F_A^{(s)} \subseteq F_N^{(s)}$ and $F_B^{(s)} \subseteq F_N^{(s)}$.

The preliminary cooccurrence model introduced in Section 4.3.3 also defines the isomorphisms f_A (from G_A to G_C), f_B (from G_B to G_C) and f_N (from G_N to G_C) that specify which elements of the graphs correspond to each other. With the help of those isomorphisms, it is possible to cross-classify the embeddings in $F_N^{(s)}$. An embedding $g \in F_N^{(s)}$ counts as an instance of G_A (G_B) if, for all vertices and edges in its domain, i. e. in G_N , the labels of the corresponding vertices and edges in G_A (G_B) are identical to or more general than the labels of the corresponding vertices and edges in G_S . Formally as Boolean functions:

$$\begin{aligned} a(g) &= \forall x \in \text{dom}(g), \lambda_A(f_A^{-1}(f_N(x))) \leq \lambda_S(g(x)) \\ b(g) &= \forall x \in \text{dom}(g), \lambda_B(f_B^{-1}(f_N(x))) \leq \lambda_S(g(x)) \end{aligned}$$

This is equivalent to stating that an embedding $g \in F_N^{(s)}$ counts as an instance of G_A (G_B) if there exists an embedding $h \in F_A^{(s)}$ ($h \in F_B^{(s)}$) such that g and h map corresponding vertices and edges of G_N and G_A (G_B) to the same vertices and edges in G_S . Formally (and equivalently to the definitions above):

$$\begin{aligned} a(g) &= \exists h \in F_A^{(s)}, h(f_A^{-1}(f_N(x))) = g(x) \text{ for all } x \in \text{dom}(g) \\ b(g) &= \exists h \in F_B^{(s)}, h(f_B^{-1}(f_N(x))) = g(x) \text{ for all } x \in \text{dom}(g) \end{aligned}$$

With those two Boolean helper functions, we can formally define the frequencies $O_{11}^{(s)}$ to $O_{22}^{(s)}$ within G_S :

$$\begin{aligned} O_{11}^{(s)} &= |\{f \in F_N^{(s)} | a(f) \text{ and } b(f)\}| = |F_C^{(s)}| \\ O_{12}^{(s)} &= |\{f \in F_N^{(s)} | a(f) \text{ and } \neg b(f)\}| \\ O_{21}^{(s)} &= |\{f \in F_N^{(s)} | \neg a(f) \text{ and } b(f)\}| \\ O_{22}^{(s)} &= |\{f \in F_N^{(s)} | \neg a(f) \text{ and } \neg b(f)\}| \end{aligned}$$

The overall frequencies can be calculated by summing over all sentences in the corpus:

$$O_{ij} = \sum_s O_{ij}^{(s)}$$

For O_{11} , this is equivalent to counting the number of embeddings of G_C :

$$O_{11} = \sum_s O_{11}^{(s)} = \sum_s |F_C^{(s)}|$$

The sample size N can be determined by summing over the four fields of the contingency table or, equivalently, by counting all embeddings of G_N :

$$N = \sum_{ij} O_{ij} = \sum_s |F_N^{(s)}|$$

The marginal frequencies can be determined by adding the respective fields of the contingency table. R_1 and C_1 can also be determined by counting all embeddings of G_A and G_B respectively:

$$R_1 = O_{11} + O_{12} = \sum_s |F_A^{(s)}|$$

$$R_2 = O_{21} + O_{22}$$

$$C_1 = O_{11} + O_{21} = \sum_s |F_B^{(s)}|$$

$$C_2 = O_{12} + O_{22}$$

Discussion

Classifying the embeddings of G_N in the sentences of the corpus yields consistent frequency counts – each embedding is only being counted towards a single cell of the contingency table. The only problem of the counting method is that it can lead to severely inflated frequencies.

Consider a query graph G_{N_k} that consists of one root vertex with k children. All vertices and edges are labeled with the most general labels, i. e. v and ϵ respectively (cf. Figure 4.29a for one such graph, G_{N_3}). Consider also a sentence graph G_{S_n} that consists of one root vertex with n children (e.g. G_{S_5} , the graph in Figure 4.29b). The number of embeddings of G_{N_k} in G_{S_n} corresponds to the number of suitable subgraphs of G_{S_n} (combinations) times the number of permutations of the k children:

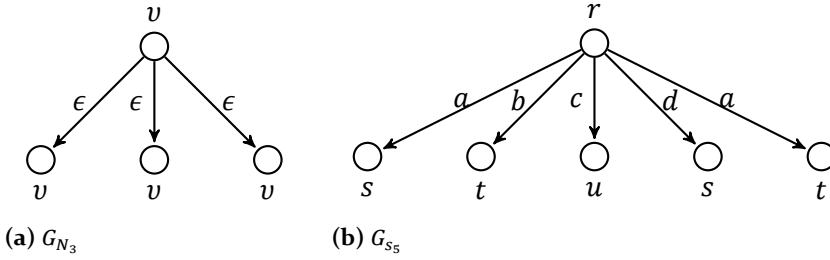


Figure 4.29: Examples of query and sentence graphs

$$\begin{aligned}
 |F| &= \binom{n}{k} k! \\
 &= \frac{n!}{k! (n-k)!} k! \\
 &= \frac{n!}{(n-k)!}
 \end{aligned}$$

The effect is visualized in Figure 4.30 (note the logarithmic scale on the y-axis). For the query graph G_{N_3} , there are 6 embeddings in G_{S_3} , 60 in G_{S_5} and 720 in for $G_{S_{10}}$. That is, $G_{S_{10}}$ provides 120 times more evidence than G_{S_3} . Imagine we are interested in the cooccurrence G_C of linguistic structures G_A and G_B such that G_N is isomorphic to G_{N_3} . Imagine further that G_C does indeed occur in sentence $G_{S_{10}}$ and that we can count 1 embedding among the 720 as an occurrence of G_C – this means that 719 embeddings are not counted as occurrences. The one sentence $G_{S_{10}}$ in which G_C does occur provides more negative evidence than 119 sentences of the G_{S_3} type in which G_C does not occur.

That is clearly counterintuitive. It also violates the randomness assumption since all those combinations and permutations are not independent. The same words will occur in many of the embeddings, leading to inflated frequency counts.

4.4.2 Counting subgraphs

When we are counting subgraphs in a linguistic structure $G_s \in \mathcal{C}$, we utilize a function $g_s : F_N^{(s)} \rightarrow \mathcal{P}(G_s)$ that maps every embedding $f \in F_N^{(s)}$ of G_N in G_s to a subgraph $G'_s \in \mathcal{P}(G_s)$ of G_s (cf. Section 4.2.2). In principle, all we have to do is to cross-classify the subgraphs in $\text{range}(g_s)$, i. e. the subgraphs of

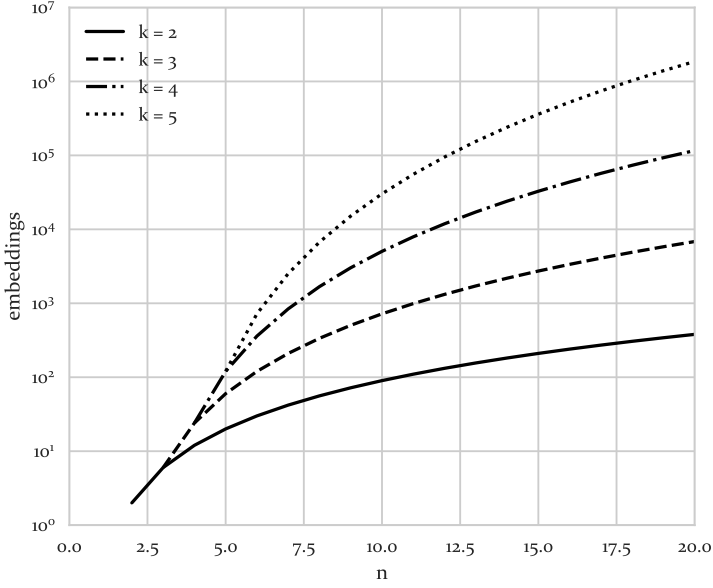


Figure 4.30: Number of embeddings of G_{N_k} in G_{S_n}

G_S that match G_N . This could be as easy as checking whether a subgraph is subsumed by G_A , G_B , both or none. The devil, however, is in the detail. Depending on G_A and G_B , it is possible that some subgraph $G'_S \in \text{range}(g_S)$ is subsumed by both G_A and G_B but not by G_C , the cooccurrence of the two, i. e.:

$$G_A \sqsubseteq G'_S \wedge G_B \sqsubseteq G'_S \Leftarrow G_C \sqsubseteq G'_S$$

As an example, consider the linguistic structures in Figure 4.31 where G_A is identical to G_B . G_S is subsumed by both G_A and G_B but not by G_C . The problem arises not only when $G_A \sqsubseteq G_B$ or $G_B \sqsubseteq G_A$ but seems to be related to symmetries in the query graphs, as can be seen from the linguistic structures in Figure 4.32. Here, although G_A and G_B do not subsume each other, G_S is still subsumed by both G_A and G_B but not by G_C .

The problem is that linguistic structures can cooccur in various ways and that, by looking at subgraphs instead of embeddings, we cannot always tell if an occurrence of G_A and an occurrence of G_B are consistent with each other and indicate an occurrence of G_C . As a consequence, there can be subgraphs that cannot be properly cross-classified. The only way to circumvent that problem is to extend the data representation to include information about the linear order of the tokens in the corpus (cf. Section 4.7) and to adapt the

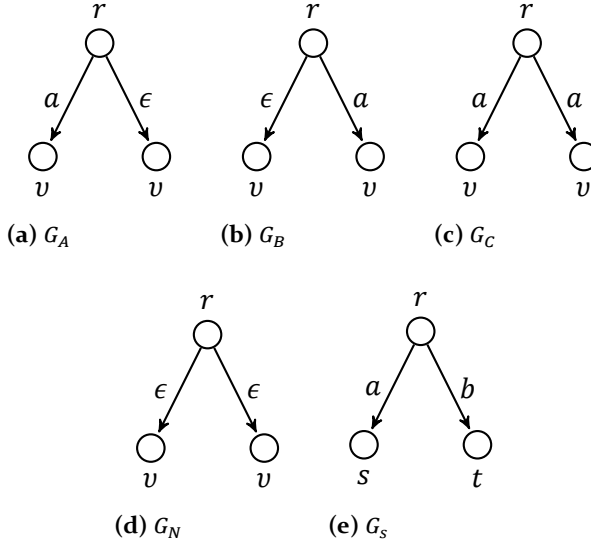


Figure 4.31: G_S is subsumed by G_A and G_B but not by G_C

query structures appropriately. If, for example, the two v vertices in G_A and G_B in Figure 4.31 could be distinguished by their linear order, only one of the two linguistic structures would subsume G_S . Unfortunately, that solution to the problem requires the user to recognize that there is a problem and to rephrase her query by specifying restrictions on the linear order of (some of) the vertices. If that does not happen, there is no satisfactory way to solve the problem. There are, however, two possible workarounds.

The first one is based on the fact that, while we cannot properly cross-classify some subgraphs that are subsumed by both G_A and G_B , we can still use measures that do not need that information. If $G_C \sqsubseteq G'_S$, then we can count G'_S as a cooccurrence, i. e. towards O_{11} . If $G_A \sqsubseteq G'_S$ and $G_B \sqsubseteq G'_S$ but $G_C \not\sqsubseteq G'_S$, then we do not have a cooccurrence we are interested in, but G'_S should still somehow count towards both R_1 and C_1 – which is impossible if we had to decide whether it counted towards O_{12} or O_{21} . Therefore, one possible workaround is to use the Jaccard coefficient (cf. Section 3.3.4) as association measure:¹²

¹² The Jaccard coefficient is a measure for the similarity of two sets and is defined as the quotient of the cardinalities of the intersection and the union of the two sets: $\text{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|}$. Since it can happen that $\{G'_S \in \text{range}(g_s) | G_C \sqsubseteq G'_S\} \neq \{G'_S \in \text{range}(g_s) | G_A \sqsubseteq G'_S\} \cap \{G'_S \in \text{range}(g_s) | G_B \sqsubseteq G'_S\}$, the application of the Jaccard coefficient might be regarded as slightly dubious in those cases.

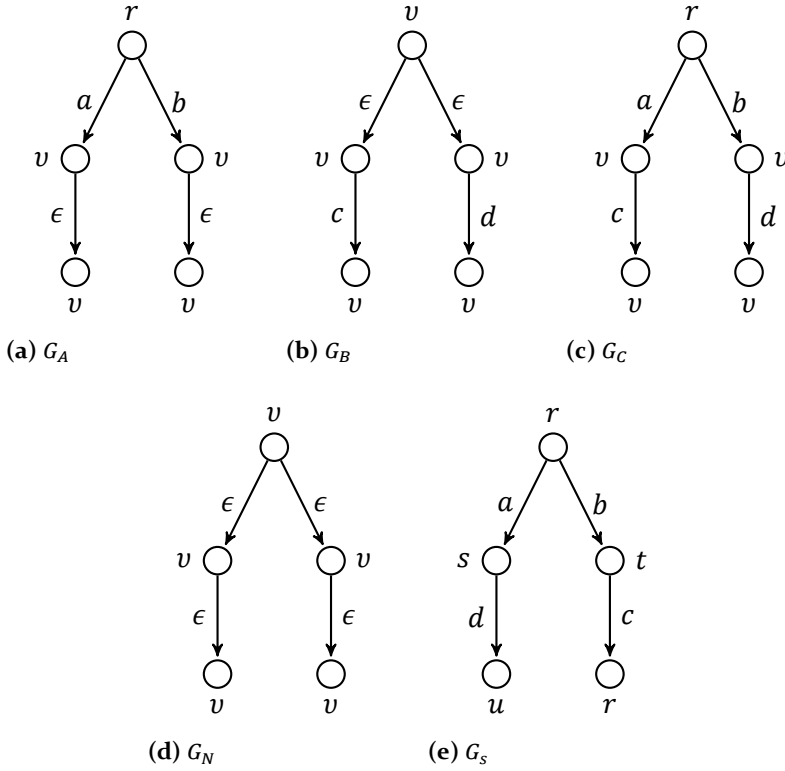


Figure 4.32: G_S is subsumed by G_A and G_B but not by G_C

$$\text{Jaccard} = \frac{o_{11}}{o_{11} + o_{12} + o_{21}} = \frac{\sum_s |\{G'_s \in \text{range}(g_s) | G_C \sqsubseteq G'_s\}|}{\sum_s |\{G'_s \in \text{range}(g_s) | G_A \sqsubseteq G'_s \text{ or } G_B \sqsubseteq G'_s\}|}$$

For calculating the Jaccard coefficient, we only need to know two things about a subgraph: I) is it subsumed by G_A or (in the sense of *and/or*) G_B and II) is it subsumed by G_C .

More generally, while it is not possible to properly populate the whole contingency table, we can still estimate the conditional probabilities $P(G_C|G_A)$ and $P(G_C|G_B)$ independently of each other and can use any association measure that is a combination of those two conditional probabilities, e.g. the Dice coefficient (cf. Section 3.3.4).¹³

¹³ Since those measures are based on conditional probabilities instead of set-theoretical interpretations, their application is not subject to the slight doubts that could be raised against the Jaccard coefficient, cf. the previous footnote.

$$P(G_C|G_A) = \frac{\sum_s |\{G'_s \in \text{range}(g_s) | G_C \sqsubseteq G'_s\}|}{\sum_s |\{G'_s \in \text{range}(g_s) | G_A \sqsubseteq G'_s\}|}$$

$$P(G_C|G_B) = \frac{\sum_s |\{G'_s \in \text{range}(g_s) | G_C \sqsubseteq G'_s\}|}{\sum_s |\{G'_s \in \text{range}(g_s) | G_B \sqsubseteq G'_s\}|}$$

$$\text{Dice} = \frac{2 \sum_s |\{G'_s \in \text{range}(g_s) | G_C \sqsubseteq G'_s\}|}{\sum_s |\{G'_s \in \text{range}(g_s) | G_A \sqsubseteq G'_s\}| + \sum_s |\{G'_s \in \text{range}(g_s) | G_B \sqsubseteq G'_s\}|}$$

The second possible workaround takes an approach that is rather more pragmatically motivated than theoretically justified. As noted above, it is impossible to decide, towards which cell of the contingency table a subgraph G'_s should be counted if that subgraph is subsumed by both G_A and G_B but not by G_C . It is not an instance of G_C and therefore should not be counted towards O_{11} . On the other hand it is subsumed by both G_A and G_B and it is therefore inadequate to count it towards O_{12} or O_{21} . G'_s should not be counted towards O_{11} but it should still count towards both marginal frequencies R_1 and C_1 . The second workaround suggested here is to count such subgraphs as half an instance towards both O_{12} and O_{21} . Over the whole corpus, that is equivalent to randomly assigning each of those subgraphs to either O_{12} or O_{21} . The result is that the subgraphs count equally towards R_1 and C_1 while the sample size N is kept stable and the frequency counts in O_{11} are reserved for real instances of G_C . Using this counting scheme, we can determine the frequencies $O_{11}^{(s)}$ to $O_{22}^{(s)}$ within G_s as follows:

$$O_{11}^{(s)} = |\{G'_s \in \text{range}(g_s) | G_C \sqsubseteq G'_s\}|$$

$$O_{12}^{(s)} = |\{G'_s \in \text{range}(g_s) | G_A \sqsubseteq G'_s \text{ and } G_B \not\sqsubseteq G'_s\}| + \frac{|\{G'_s \in \text{range}(g_s) | G_A \sqsubseteq G'_s \text{ and } G_B \sqsubseteq G'_s \text{ and } G_C \not\sqsubseteq G'_s\}|}{2}$$

$$O_{21}^{(s)} = |\{G'_s \in \text{range}(g_s) | G_A \not\sqsubseteq G'_s \text{ and } G_B \sqsubseteq G'_s\}| + \frac{|\{G'_s \in \text{range}(g_s) | G_A \sqsubseteq G'_s \text{ and } G_B \sqsubseteq G'_s \text{ and } G_C \not\sqsubseteq G'_s\}|}{2}$$

$$O_{22}^{(s)} = |\{G'_s \in \text{range}(g_s) | G_A \not\sqsubseteq G'_s \text{ and } G_B \not\sqsubseteq G'_s\}|$$

$O_{11}^{(s)}$ is the number of subgraphs in $\text{range}(g_s)$ that are subsumed by G_C . $O_{12}^{(s)}$ is the number of subgraphs in $\text{range}(g_s)$ that are subsumed by G_A but not by G_B plus half the number of subgraphs that are subsumed by both G_A and G_B but not by G_C . $O_{21}^{(s)}$ is the number of subgraphs in $\text{range}(g_s)$ that are subsumed by G_B but not by G_A plus half the number of subgraphs that are

subsumed by both G_A and G_B but not by G_C . $O_{22}^{(s)}$ is the number of subgraphs in $\text{range}(g_s)$ that are subsumed by neither G_A nor G_B .

The overall frequencies can be calculated by summing over all sentences in the corpus:

$$O_{ij} = \sum_s O_{ij}^{(s)}$$

The sample size N can be determined by summing over the four fields of the contingency table or, equivalently, by counting all subgraphs corresponding to G_N :

$$N = \sum_{ij} O_{ij} = \sum_s |\text{range}(g_s)|$$

Due to the split counts of some subgraphs, the marginal frequencies R_1 , C_1 , R_2 and C_2 cannot easily be stated as subsets of $\text{range}(g_s)$ and are best expressed as sums of the respective cells of the contingency table:

$$R_1 = O_{11} + O_{12}$$

$$C_1 = O_{11} + O_{21}$$

$$R_2 = O_{21} + O_{22}$$

$$C_2 = O_{12} + O_{22}$$

Discussion

Counting subgraphs has two problems. The first problem is that, mostly because of symmetries in the query structures, the cross-classification of subgraphs is not in all cases well-defined (see above). This problem can either be circumvented by adding further restrictions to the linguistic structures, e. g. on the linear order of the vertices, or it can be worked around to some extent by either choosing an appropriate association measure or by introducing an unorthodox counting method.

The second problem is a weaker form of the problem that arises when counting embeddings (cf. Section 4.4.1). Coming back to the G_{N_k} and G_{s_n} example from Section 4.4.1, we do no longer have to deal with the permutations of the k child vertices of G_{N_k} but there are still the combinations of subgraphs of G_{s_n} :

$$\begin{aligned} \text{subgraphs}(G_{N_k}, G_{S_n}) &= \binom{n}{k} \\ &= \frac{n!}{k!(n-k)!} \end{aligned}$$

The effect is that, compared to embeddings, the numbers are divided by $k!$ when counting subgraphs. Nevertheless, there is a combinatorial explosion as can be seen in Figure 4.33.

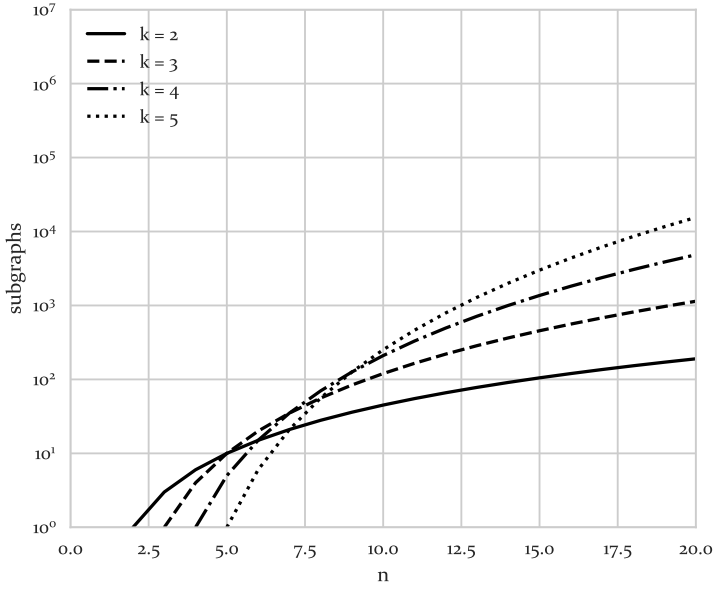


Figure 4.33: Number of subgraphs for G_{N_k} and G_{S_n}

Returning to the example query graph G_{N_3} from Section 4.4.1, there is 1 subgraph in G_{S_3} , but there are already 10 in G_{S_5} and 120 in $G_{S_{10}}$. While the numbers are six times lower ($3! = 6$) than when counting embeddings, the ratios are still the same ($G_{S_{10}}$ provides 120 times more evidence than G_{S_3}) and we still can have inflated frequencies.

4.4.3 Counting focus points

We can also cross-classify the vertices of G_S that correspond to the focus point vertex of G_N (cf. Section 4.2.4). Since we are cross-classifying individual vertices, we cannot have inflated frequencies, i. e. the frequency counts obtained by this counting method are methodologically sound from that perspective. However, the inconsistency problem that can occur when counting subgraphs can also occur when counting focus points: A vertex $v \in V_S$ that corresponds to the focus point vertex $u \in V_N$ can be part of both G_A and G_B but not of G_C (cf. Figure 4.34; the focus point vertex is labeled r). Or, put differently, for the linguistic structures in Figure 4.34, there are two embeddings of G_N in G_S and both are in the same block when we partition the set of embeddings according to the counting method. Now we have, in a single block of the partition, an embedding of G_A and an embedding of G_B but no embedding of G_C .

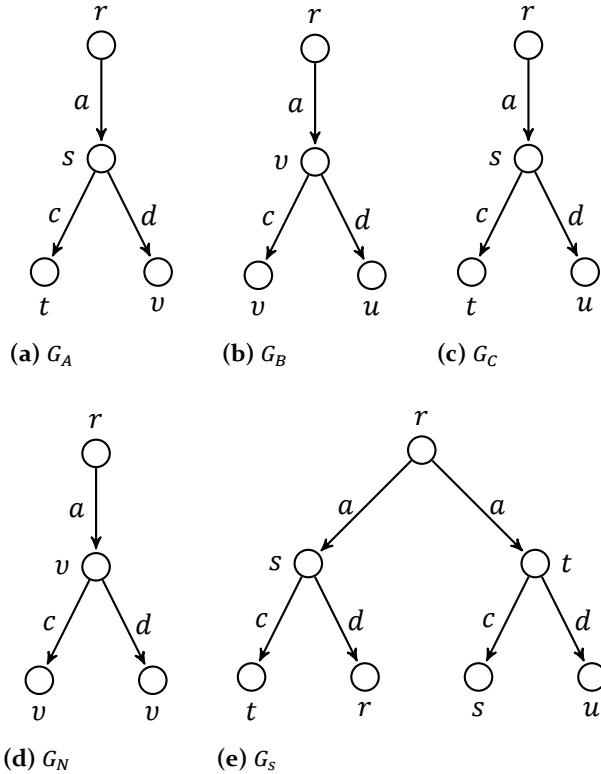


Figure 4.34: The focus point vertex r in G_S is part of G_A and G_B but not of G_C

The strategies for working around that problem that were introduced in Section 4.4.2 for counting subgraphs can also be applied when counting focus points. Note, however, that additional constraints on the linear order of the vertices will not necessarily solve the problem if G_A and G_B are mapped to distinct subgraphs of G_S that only overlap in a single vertex (as in Figure 4.34).

In order to determine the frequencies $O_{11}^{(s)}$ to $O_{22}^{(s)}$ in G_S , we define a series of auxiliary vertex sets that are subsets of the vertices V_S of G_S . We assume that $u \in V_N$ is the focus point vertex of G_N and we define the set $U_N^{(s)}$ as those vertices of G_S that u is mapped to by the embeddings in $F_N^{(s)}$. The sets $U_A^{(s)}$, $U_B^{(s)}$ and $U_C^{(s)}$ can be similarly defined for the vertices corresponding to u in G_A , G_B and G_C :

$$\begin{aligned} U_N^{(s)} &= \{v \in V_S \mid \exists (f^v \in F_N^{(s)}), v = f^v(u)\} \\ U_A^{(s)} &= \{v \in V_S \mid \exists (f^v \in F_A^{(s)}), v = f^v(f_A^{-1}(f_N(u)))\} \\ U_B^{(s)} &= \{v \in V_S \mid \exists (f^v \in F_B^{(s)}), v = f^v(f_B^{-1}(f_N(u)))\} \\ U_C^{(s)} &= \{v \in V_S \mid \exists (f^v \in F_C^{(s)}), v = f^v(f_N(u))\} \end{aligned}$$

Since $G_N \sqsubseteq G_A$, $G_N \sqsubseteq G_B$, $G_A \sqsubseteq G_C$ and $G_B \sqsubseteq G_C$, the following relations between the vertex sets hold: $U_A^{(s)} \subseteq U_N^{(s)}$, $U_B^{(s)} \subseteq U_N^{(s)}$ and $U_C^{(s)} \subseteq U_A^{(s)} \cap U_B^{(s)}$. With the help of those vertex sets it is possible to express the cross-classification of the vertices in $U_N^{(s)}$ into the four cells of the contingency table using set operations:

$$\begin{aligned} O_{11}^{(s)} &= |U_N^{(s)} \cap U_C^{(s)}| = |U_C^{(s)}| \\ O_{12}^{(s)} &= |U_N^{(s)} \cap (U_A^{(s)} \setminus U_B^{(s)})| + \frac{|(U_N^{(s)} \cap U_A^{(s)} \cap U_B^{(s)}) \setminus U_C^{(s)}|}{2} \\ &= |U_A^{(s)} \setminus U_B^{(s)}| + \frac{|(U_A^{(s)} \cap U_B^{(s)}) \setminus U_C^{(s)}|}{2} \\ O_{21}^{(s)} &= |U_N^{(s)} \cap (U_B^{(s)} \setminus U_A^{(s)})| + \frac{|(U_N^{(s)} \cap U_A^{(s)} \cap U_B^{(s)}) \setminus U_C^{(s)}|}{2} \\ &= |U_B^{(s)} \setminus U_A^{(s)}| + \frac{|(U_A^{(s)} \cap U_B^{(s)}) \setminus U_C^{(s)}|}{2} \\ O_{22}^{(s)} &= |U_N^{(s)} \setminus (U_A^{(s)} \cup U_B^{(s)})| \end{aligned}$$

The overall frequencies can be calculated by summing over all sentences in the corpus:

$$o_{ij} = \sum_s o_{ij}^{(s)}$$

The sample size N can be determined by summing over the four fields of the contingency table or, equivalently, by counting all vertices in the corpus that correspond to the focus point vertex in G_N :

$$N = \sum_{ij} o_{ij} = \sum_s |U_N^{(s)}|$$

4.4.4 Counting graphs (sentences)

Instead of classifying embeddings, subgraphs or focus point vertices, we can also classify all graphs, i. e. sentences, that are subsumed by G_N . It should be intuitively clear that the inconsistency problem that can occur when classifying subgraphs (cf. Section 4.4.2) or focus point vertices (cf. Section 4.4.3) is an even bigger problem when dealing with whole graphs or sentences. For the problem to arise when classifying subgraphs, linguistic structures G_A and G_B have to occur in the same subgraph of G_S , i. e. the very same set of vertices and edges has to match both G_A and G_B while not matching G_C , – a rather unlikely scenario for authentic language data. When classifying focus point vertices, the subgraphs of G_S that G_A and G_B occur in have to overlap in at least one vertex. This scenario is a superset of that for the classification of subgraphs and will therefore be more frequent (also, the structures in Figure 4.34 indicate that it might be quite likely for linguistically plausible data). When classifying graphs, all cases that are problematic when classifying subgraphs and focus point vertices remain problematic. In addition, all occurrence restrictions are lifted and G_A and G_B can occur in totally different subgraphs of G_S without any overlap (cf. Figure 4.35), leading to a still greater number of problematic cases. In these cases, when the occurrences are independent of each other, even specifying additional constraints on the linear order of the vertices does not help. The inconsistency problem can be formally expressed as follows:

$$G_A \sqsubseteq G_S \wedge G_B \sqsubseteq G_S \Leftarrow G_C \sqsubseteq G_S$$

On the plus side, since we are classifying whole sentences and each sentence is only counted once, there are no inflated frequencies.

We are classifying all sentences in the corpus that are subsumed by G_N , i. e. $\{G_S \in \mathcal{C} \mid G_N \sqsubseteq G_S\}$. Because of the subsumption relations visualized in

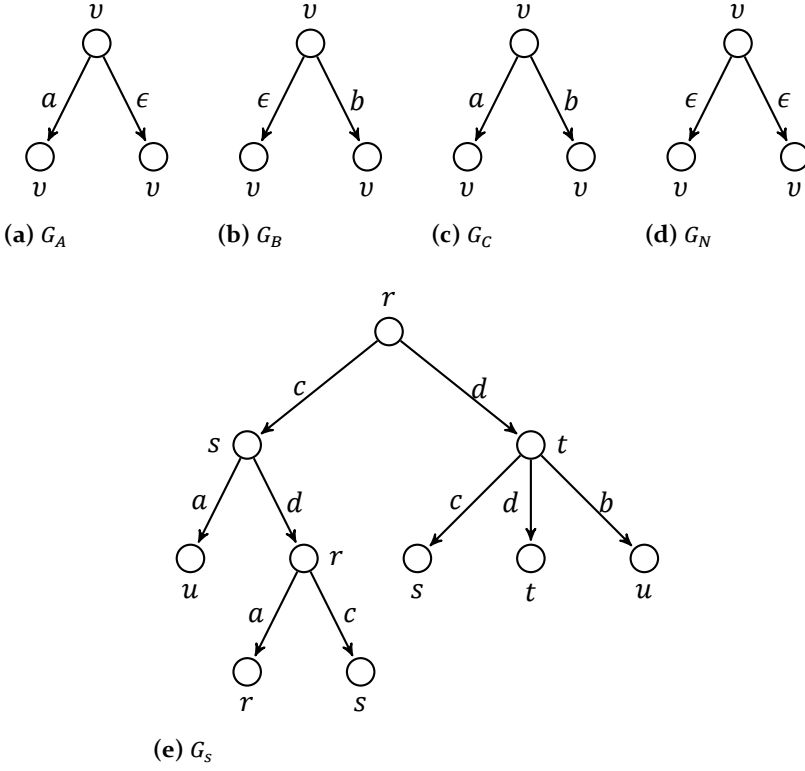


Figure 4.35: G_S is subsumed by G_A and G_B but not by G_C

Figure 4.27 (p. 90), we do not have to state explicitly that $G_N \sqsubseteq G_S$ in most cases and can express the frequencies $O_{11}^{(s)}$ to $O_{22}^{(s)}$ as follows:

$$O_{11}^{(s)} = \begin{cases} 1 & \text{if } G_C \sqsubseteq G_S \\ 0 & \text{otherwise} \end{cases}$$

$$O_{12}^{(s)} = \begin{cases} 1 & \text{if } G_A \sqsubseteq G_S \text{ and } G_B \not\sqsubseteq G_S \\ 0.5 & \text{if } G_A \sqsubseteq G_S \text{ and } G_B \sqsubseteq G_S \text{ and } G_C \not\sqsubseteq G_S \\ 0 & \text{otherwise} \end{cases}$$

$$O_{21}^{(s)} = \begin{cases} 1 & \text{if } G_A \not\sqsubseteq G_S \text{ and } G_B \sqsubseteq G_S \\ 0.5 & \text{if } G_A \sqsubseteq G_S \text{ and } G_B \sqsubseteq G_S \text{ and } G_C \not\sqsubseteq G_S \\ 0 & \text{otherwise} \end{cases}$$

$$O_{22}^{(s)} = \begin{cases} 1 & \text{if } G_N \sqsubseteq G_S \text{ and } G_A \not\sqsubseteq G_S \text{ and } G_B \not\sqsubseteq G_S \\ 0 & \text{otherwise} \end{cases}$$

The overall frequencies can be calculated by summing over all sentences in the corpus:

$$o_{ij} = \sum_s o_{ij}^{(s)}$$

The sample size N can be determined by summing over the four fields of the contingency table and equals the number of sentences subsumed by G_N :

$$N = \sum_{ij} o_{ij} = |\{G_s \in \mathcal{C} | G_N \sqsubseteq G_s\}|$$

Here we are only classifying graphs that are subsumed by G_N . One might be tempted to classify *all* the graphs in the corpus instead, following the example of segment-based cooccurrences (Evert, 2004: 66–68; cf. Section 2.1.1). It would be trivial to do so, the only necessary change affects the determination of $O_{22}^{(s)}$:

$$O_{22}^{(s)} = \begin{cases} 1 & \text{if } G_A \not\sqsubseteq G_s \text{ and } G_B \not\sqsubseteq G_s \\ 0 & \text{otherwise} \end{cases}$$

The sample size N would then correspond to the cardinality of $\mathcal{C}$, i. e. the number of sentences in the corpus:

$$N = \sum_{ij} o_{ij} = |\mathcal{C}|$$

That small change would constitute a methodological break, though. The preliminary cooccurrence model developed in Section 4.3 is an extension of relational cooccurrences and covarying collexeme analysis and is based on the classification of instances of a structure that is an abstraction from the cooccurrence we are interested in. In the case of classifying graphs we have a rather broad interpretation of what an instance is, namely a whole graph or sentence – but nevertheless we have to stick to those instances.¹⁴ This is not a weakness of our model but simply a consequence of the fact that it is different from the model of segment-based cooccurrences.¹⁵ If

¹⁴ When we are analyzing segment-based cooccurrences, we determine the association between two individual words. We do not care whether the two words are connected via some relation, syntactic or other. We do not even care whether the words are a part of any special kind of relation at all. Since the cooccurrences specified in our model are necessarily based on relations between entities, a model that is based on ignoring those relations is not adequate.

¹⁵ Note, however, that segment-based cooccurrences, including classification of all sentences, can be reconstructed in the generalized cooccurrence model if we have the proper annotation in place, cf. Section 4.6.1.

we break with our model and classify all sentences in the corpus instead, the number of sentences subsumed by neither G_A nor G_B (O_{22}) will increase. What impact that increase has, depends on the association measure employed (cf. Section 5.1.2 for a detailed analysis). While it will not have any effect on certain measures of effect size like the Dice and Jaccard coefficients, it will increase the values for the chi-squared and log-likelihood measures (making cooccurrences more significant). For those measures, increasing O_{22} can also influence the ranking of associated linguistic structures.¹⁶ Regarding the drastic influence that the sample size can have, the advantage of the current preliminary model (and of course the final, generalized cooccurrence model that will be introduced in Section 4.5) is that the sample size is derived automatically in a sensible manner from the query structures.

4.4.5 Adequacy of the counting methods

When counting individual linguistic structures, the four counting methods can be generalized using the notions of blocks and partitions (cf. Section 4.2.5). The same can be done when counting and cross-classifying linguistic structures (cf. Section 4.5.2). The big question, however, is which of those counting methods is best suited for determining the association strength between two linguistic structures. Table 4.1 summarizes to what extent the counting methods are affected by inflated frequencies and the inconsistency problem. Given that inflated frequencies are a serious violation of the randomness assumption and we cannot have inflated frequencies when classifying focus points or graphs, those two counting methods might be superior to the other two. Since the inconsistency problem is a bigger issue when classifying graphs than when classifying focus points, we might conclude that counting and classifying focus point vertices has the least disadvantages. For a more detailed analysis of the interplay between counting methods and association measures that also involves real corpus data cf. Chapter 5.

¹⁶ This can be illustrated with the following example. The frequency signature for the cooccurrence of linguistic structures X and Y is (50, 200, 100, 1000), leading to a log-likelihood value of 73.8. The frequency signature for the cooccurrence of linguistic structures X and Z is (150, 200, 600, 1000), leading to a log-likelihood value of 35.5. Clearly X and Y are more strongly associated than X and Z . If we now increase O_{22} by 500, we get the following frequency signatures: (50, 200, 100, 1500) for X and Y and (150, 200, 600, 1500) for X and Z . Those frequency signatures lead to a log-likelihood value of 124.1 for X and Y and of 168.8 for X and Z . The addition of seemingly irrelevant material has changed the ranking: X and Z are now more strongly associated than X and Y .

counting method	inflated frequencies	inconsistency
embeddings	very bad	very good
subgraphs	very bad	bad
focus points	very good	bad
graphs	very good	very bad

Table 4.1: Counting methods and their problems

4.5 The generalized cooccurrence model

The preliminary cooccurrence model introduced in Section 4.3 is limited to the analysis of cooccurrences of linguistic structures that are structurally identical (i. e. isomorphic when ignoring the labels) and only differ in some vertex and edge labels. Due to that limitation, the cooccurrence of a smaller linguistic structure with a larger one cannot be analyzed properly. As an example, consider simple collexeme analysis, i. e. the cooccurrence of a word, e. g. the verb *sing*, with a syntactic structure, e. g. the ditransitive. Due to the requirement of structural identity in the preliminary model, we can only analyze the cooccurrence of *sing* with at least two outgoing dependency relations with the ditransitive, i. e. instances of *sing* with less than two outgoing dependency relations will not be taken into account. That is clearly not desirable. Therefore, in the remainder of this section, we will take a closer look at simple collexeme analysis and will formulate a truly generalized cooccurrence model of which relational cooccurrences, covarying collexeme analysis and simple collexeme analysis are merely special cases that will be reconstructed in Section 4.6.

4.5.1 Simple collexeme analysis revisited

We saw in Sections 4.3.1 and 4.3.2 that relational cooccurrences analyze the cooccurrence of two words within the simplest syntactic structure, namely a single dependency relation. Covarying collexeme analysis is a generalization of that approach that allows for the cooccurrence of two words within arbitrary syntactic structures. In both cases we can interpret the cooccurrence as the cooccurrence of two completely overlapping, structurally identical syntactic structures that only differ in two words, i. e. two vertex labels.

Simple collexeme analysis (cf. Section 2.2.1) cannot be interpreted in that way. It analyzes the cooccurrence of a single word with a larger structure and classifies individual tokens according to whether they equal that word and

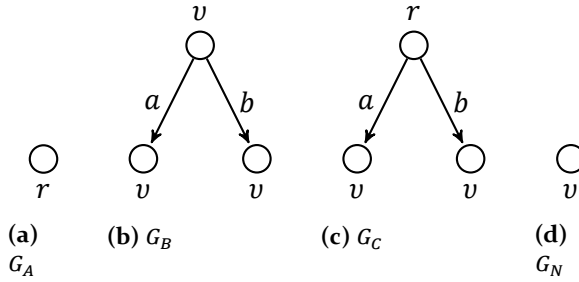


Figure 4.36: Query graphs for simple collexeme analysis

whether they are part of that larger structure. The main difference, therefore, is that the two cooccurring structures cannot be interpreted as being isomorphic when ignoring their labels. But they are also not completely different. The smaller structure (the single word) is part of the larger structure and this means that one of the key concepts of the preliminary model is still applicable: Overlap. For relational cooccurrences and covarying collexeme analysis, the two cooccurring structures overlap completely, for simple collexeme analysis, there is a partial overlap which means that the smaller structure that only consists of a single vertex is contained within the larger one.

Consider the graphs in Figure 4.36. G_C is the cooccurrence of G_A and G_B . Although G_A and G_B are not isomorphic, they do overlap in one vertex. Similar, to what we did in the preliminary model, we take that overlapping part and use the more general label from G_A and G_B , resulting in linguistic structure G_N . We classify all occurrences of G_N , i. e. all vertices in the corpus, according to whether they are labeled r (and constitute an occurrence of G_A) and whether they have outgoing a and b relations (and constitute an occurrence of G_B).

In the special case of simple collexeme analysis, the smaller of the two structures G_A and G_B is just a single vertex with a more specific label than the corresponding vertex in the other structure. If we treat that vertex as a focus point vertex (cf. Sections 4.2.4 and 4.4.3) and apply the corresponding counting method, the frequency signature (O_{11}, R_1, C_1, N) equals the frequencies of G_C , G_A , G_B , and G_N . For other counting methods, only the structures consisting of a single vertex yield the correct frequencies corresponding to a cross-classification of G_N .

4.5.2 Generalization to arbitrary linguistic structures

We can lift the restrictions on structural identity and complete overlap introduced in the preliminary model. Instead of having two isomorphic structures that differ only in some vertex and/or edge labels and that overlap completely, we can allow for arbitrary linguistic structures. The only requirement is that the two structures have to overlap. It does not matter to what extent, anything from a single vertex to complete overlap (as in the preliminary model) is fine. The part that overlaps constitutes G_N , the linguistic structure whose instances will be cross-classified. This means that we can analyze any cooccurrence of two linguistic structures as long as the overlapping part is connected and therefore itself a linguistic structure.¹⁷ The resulting model is much more flexible and general than the preliminary model and can indeed be used to reconstruct that model (cf. Section 4.6).

Let us illustrate the generalized model with the graphs in Figure 4.37. The root vertex of G_A , the vertex labeled r , has two outgoing edges, one labeled a and one labeled b . Edge a connects r to a vertex labeled s , edge b to a vertex labeled t . In G_B , there is an edge a connecting a vertex labeled r with another labeled v and an edge c from the v vertex to a vertex labeled u . The two linguistic structures cooccur as shown in G_C , overlapping in the a edge and its incident vertices. That overlapping part of G_A and G_B , i. e. the a edge connecting the vertices labeled r and v (the label v is more general than the label s), constitutes G_N . To analyze the association between G_A and G_B , we have to cross-classify all instances of G_N . If an instance of G_N has its v vertex labeled s and has an outgoing edge labeled b from its r vertex to a t vertex, it is also an instance of G_A . If an instance of G_N has an outgoing c edge from its v vertex to a vertex labeled u , it is also an instance of G_B .

Ideally, if an instance of G_N is an instance of both G_A and G_B , it is also an instance of G_C . As has been shown in Section 4.4, this is only the case for the preliminary model if we are classifying the embeddings of G_N in the sentences in the corpus. In the generalized model, we lift the restriction that G_A , G_B , G_C and G_N have to be isomorphic and that leads to inconsistent cases even when we are classifying embeddings. Consider the linguistic structures in Figure 4.38. G_A and G_B are identical and cooccur in G_C in such a way that they overlap in their root vertices. Therefore, G_N is a single vertex labeled r . There is one instance of G_N in G_S and that instance is also an instance of G_A

¹⁷ Without this restriction, it could happen that two linguistic structures overlap in n unconnected vertices, meaning that we would have to classify all combinations of n vertices. Each vertex in the corpus would be part of many combinations, violating the randomness assumption (cf. also Section 4.4).

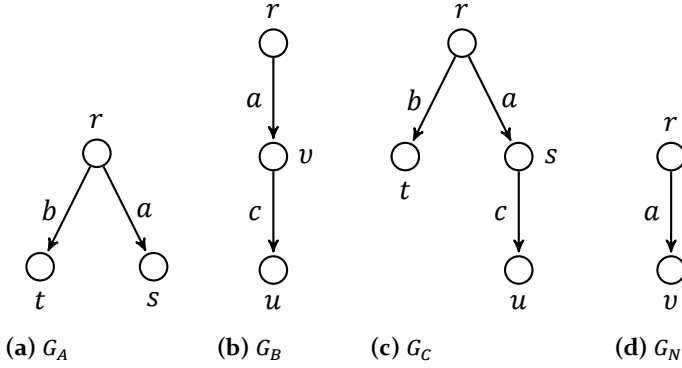


Figure 4.37: Query graphs for the generalized model

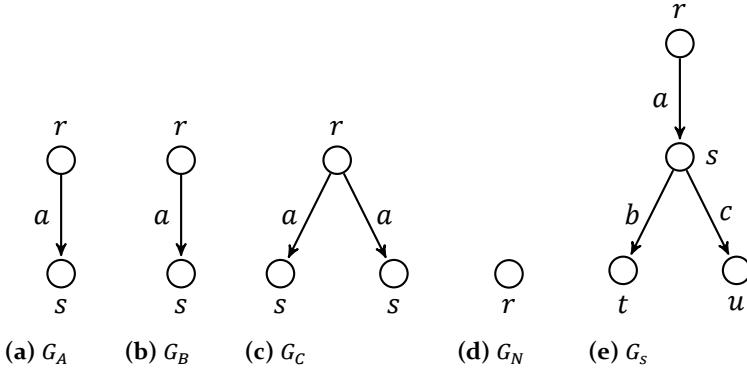


Figure 4.38: G_S matches G_A and G_B but not G_C

and G_B (since they are identical). However, it is not an instance of G_C . That means we will have to resort to the strategies developed in Section 4.4.2 for all counting methods.

4.5.3 Formalization

Similar to what we did when formalizing the preliminary model, we have to exactly specify the linguistic structure G_C in which G_A and G_B cooccur and the relations between those structures and G_N . What follows is a formal definition of the generalized model for the cooccurrence of linguistic structures:

$G_C = (V_C, E_C, s_C, t_C, \langle \Sigma, \leq \rangle, \lambda_C)$ is a linguistic structure in which linguistic structure $G_A = (V_A, E_A, s_A, t_A, \langle \Sigma, \leq \rangle, \lambda_A)$ cooccurs with linguistic structure $G_B = (V_B, E_B, s_B, t_B, \langle \Sigma, \leq \rangle, \lambda_B)$ such that the following constraints hold:

1. $G_\alpha = (V_\alpha, E_\alpha, s_\alpha, t_\alpha)$ is a connected subgraph of (V_C, E_C, s_C, t_C) .
2. $G_\beta = (V_\beta, E_\beta, s_\beta, t_\beta)$ is a connected subgraph of (V_C, E_C, s_C, t_C) .
3. $V_\alpha \cup V_\beta = V_C$ and $E_\alpha \cup E_\beta = E_C$.
4. $G_l = (V_l, E_l, s_l, t_l)$ is a connected subgraph of G_C where $V_l = V_\alpha \cap V_\beta$, $E_l = E_\alpha \cap E_\beta$ and s_l and t_l are the restrictions of s_C and t_C to E_l .
5. (V_A, E_A, s_A, t_A) is isomorphic to G_α and the isomorphism $f_A : G_A \rightarrow G_\alpha$ from G_A to G_α specifies which elements of the two graphs correspond to each other.
6. (V_B, E_B, s_B, t_B) is isomorphic to G_β and the isomorphism $f_B : G_B \rightarrow G_\beta$ from G_B to G_β specifies which elements of the two graphs correspond to each other.
7. $\lambda_C(x) = \begin{cases} \lambda_A(f_A^{-1}(x)) & \text{if } x \in G_\alpha \text{ and } x \notin G_\beta \\ \lambda_B(f_B^{-1}(x)) & \text{if } x \notin G_\alpha \text{ and } x \in G_\beta \\ \max(\lambda_A(f_A^{-1}(x)), \lambda_B(f_B^{-1}(x))) & \text{if } x \in G_\alpha \text{ and } x \in G_\beta, \\ & \text{i.e. } x \in G_l. \end{cases}$

This means that all vertices and edges in G_C come from G_A and/or G_B . Those from G_A are labeled as in G_A , those from G_B are labeled as in G_B . Vertices or edges in G_C that are present in both G_A and G_B are labeled with the more specific of the two labels. Of course, a prerequisite for that is that $\lambda_A(f_A^{-1}(x))$ and $\lambda_B(f_B^{-1}(x))$, the labels of corresponding elements in G_A and G_B , are comparable. Because of condition 3, G_A and G_B have at least one vertex in common and because of condition 4, the vertices and edges that they have in common form a connected graph. In an attempt at mnemonic naming, the cooccurrence of G_A and G_B is called G_C and the intersection of G_α and G_β is called G_l . Now that the relationship between those graphs has been defined, G_N can be derived as follows.

$G_N = (V_N, E_N, s_N, t_N, \langle \Sigma, \leq \rangle, \lambda_N)$ is a linguistic structure where (V_N, E_N, s_N, t_N) is isomorphic to G_l , the isomorphism $f_N : G_N \rightarrow G_l$ from G_N to G_l specifies which elements of the two graphs correspond to each other and $\lambda_N(x) = \min(\lambda_A(f_A^{-1}(f_N(x))), \lambda_B(f_B^{-1}(f_N(x))))$. This means that G_N is structurally identical to G_l and that the vertices and edges in G_N are labeled with the more general label from either G_A or G_B .

As a result, the following relations (visualized in the Hasse diagram in Figure 4.39) always hold: $G_N \sqsubseteq G_A$, $G_N \sqsubseteq G_B$, $G_A \sqsubseteq G_C$ and $G_B \sqsubseteq G_C$.

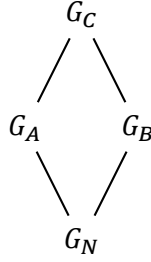


Figure 4.39: Hasse diagram of the linguistic structures in the generalized cooccurrence model

For the statistical analysis of the cooccurrence we assume that a corpus $C = \{G_1, G_2, \dots, G_{S-1}, G_S\}$ is a multiset of S linguistic structures. For each linguistic structure $G_s \in C$, i. e. for each sentence in the corpus, there exists a (possibly empty) set $F_N^{(s)}$ of embeddings of G_N in G_s . Those embeddings can be partitioned into blocks depending on the chosen counting method (cf. Section 4.2.5). The units of analysis are the blocks of the partition $P_s = \{F_N^{(s,k)}\}_{k \in K}$ of $F_N^{(s)}$ that corresponds to the applied counting method. The four counting methods partition $F_N^{(s)}$ as follows:

Embeddings: Every embedding is a single block in the partition: $P_s = \{\{f\} | f \in F_N^{(s)}\}$.

Subgraphs: Embeddings that correspond to the same subgraph of G_s are grouped together: $P = \{\{f | f \in F_N^{(s)} \text{ and } g(f) = G'_s\} | G'_s \in \{g(f) | f \in F_N^{(s)}\}\}$, where g is a function that maps every embedding in $F_N^{(s)}$ to a subgraph of G_s (cf. Section 4.2.2).

Graphs: All embeddings are in the same block, i. e. there is only a single block: $P = \{F_N^{(s)}\}$.

Focus point vertices: All embeddings that map the focus point vertex $u \in V_N$ to the same vertex $v \in V_s$ are grouped together: $P = \{\{f | f \in F_N^{(s)} \text{ and } f^v(u) = v\} | v \in \{f^v(u) | f \in F_N^{(s)}\}\}$.

In addition to $F_N^{(s)}$, the following three (possibly empty) sets of embeddings exist for each sentence $G_s \in C$: I) $F_A^{(s)}$ is the set of embeddings of G_A in G_s , II) $F_B^{(s)}$ is the set of embeddings of G_B in G_s and III) $F_C^{(s)}$ is the set of embeddings of G_C in G_s . As described in Chapter 3, the cross-classification of the units of analysis, i. e. the blocks of P_s , can be represented in a 2×2

contingency table. The values of the four cells O_{11} to O_{22} can be determined with the help of indicator variables that represent the classification of a single block $F_N^{(s,k)} \in P_s$. These indicator variables are defined in terms of the three helper sets $H_A^{(s,k)}$, $H_B^{(s,k)}$ and $H_C^{(s,k)}$. $H_A^{(s,k)}$ is the subset of $F_A^{(s)}$ that is compatible with block $F_N^{(s,k)}$. More formally, the members of $H_A^{(s,k)}$ are those embeddings from $F_A^{(s)}$ for which there is an embedding in the current block $F_N^{(s,k)}$ such that corresponding vertices and edges in G_A and G_N are mapped to the same vertices and edges in G_s . $H_B^{(s,k)}$ and $H_C^{(s,k)}$ are defined analogously:

$$\begin{aligned} H_A^{(s,k)} &= \{g \in F_A^{(s)} \mid \exists (h \in F_N^{(s,k)}), g(f_A^{-1}(x)) = h(f_N^{-1}(x)) \text{ for all } x \in G_i\} \\ H_B^{(s,k)} &= \{g \in F_B^{(s)} \mid \exists (h \in F_N^{(s,k)}), g(f_B^{-1}(x)) = h(f_N^{-1}(x)) \text{ for all } x \in G_i\} \\ H_C^{(s,k)} &= \{g \in F_C^{(s)} \mid \exists (h \in F_N^{(s,k)}), g(x) = h(f_N^{-1}(x)) \text{ for all } x \in G_i\} \end{aligned}$$

The interpretation of these helper sets is straightforward: If a helper set is nonempty, then the block constitutes an instance of the corresponding linguistic structure. If, for example, $|H_A^{(s,k)}| > 0$, then $F_N^{(s,k)}$ is an instance of G_A .

The problem is that it is possible for a block $F_N^{(s,k)}$ to be an instance of both G_A and G_B but not of G_C , the cooccurrence of the two (the inconsistency problem, cf. Sections 4.4.2– 4.4.5):

$$H_A^{(s,k)} \neq \emptyset \wedge H_B^{(s,k)} \neq \emptyset \iff H_C^{(s,k)} \neq \emptyset$$

This problem affects all counting methods. We can have inconsistencies even when classifying embeddings, since $G_i \subseteq G_\alpha$ and $G_i \subseteq G_\beta$ and the embeddings need only be consistent for the elements in G_i but can be inconsistent for the other elements. As an example, consider the linguistic structures in Figure 4.40. There is a single embedding of G_N in G_s . There are also embeddings of both G_A and G_B in G_s that are compatible with G_N . Nevertheless, there is no embedding of G_C in G_s .

The significance of the inconsistency problem lies in the fact that there can be blocks that cannot properly be cross-classified, i. e. that the contingency table cannot properly be populated. In Section 4.4.2, we presented two ways for working around this problem.

For the first workaround, we simply do not rely on the contingency table. While it is not possible to properly populate the whole contingency table, we can still estimate the conditional probabilities $P(G_C|G_A)$ and $P(G_C|G_B)$ independently of each other and can use any association measure based on combining those two conditional probabilities. To do that, we define the

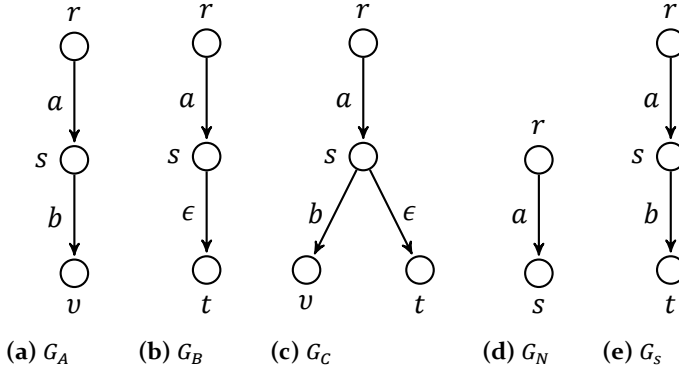


Figure 4.40: Inconsistencies even when counting embeddings

following three indicator variables for whether a block constitutes an instance of G_A , G_B or G_C :

$$I_A^{(s,k)} = \begin{cases} 1 & \text{if } |H_A^{(s,k)}| \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$I_B^{(s,k)} = \begin{cases} 1 & \text{if } |H_B^{(s,k)}| \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$I_C^{(s,k)} = \begin{cases} 1 & \text{if } |H_C^{(s,k)}| \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

The total number of instances of G_A , G_B and G_C can be obtained by summing over all blocks of all sentences. Using these frequencies, we can give the maximum-likelihood estimates of the conditional probabilities:

$$P(G_C|G_A) = \frac{\sum_{sk} I_C^{(s,k)}}{\sum_{sk} I_A^{(s,k)}}$$

$$P(G_C|G_B) = \frac{\sum_{sk} I_C^{(s,k)}}{\sum_{sk} I_B^{(s,k)}}$$

The two conditional probabilities can be combined by a suitable linkage function to arrive at a single association measure. Using the harmonic mean, for example, yields the Dice coefficient (cf. Section 3.3.4):

$$\begin{aligned} \text{Dice} &= \frac{2 \cdot P(G_C|G_A) \cdot P(G_C|G_B)}{P(G_C|G_A) + P(G_C|G_B)} \\ &= \frac{2 \cdot \sum_{sk} I_C^{(s,k)}}{\sum_{sk} I_A^{(s,k)} + \sum_{sk} I_B^{(s,k)}} \end{aligned}$$

The second workaround is based on a pragmatic treatment of the inconsistency problem that leads to a well-behaved contingency table. If a block $F_N^{(s,k)}$ is an instance of both G_A and G_B but not of G_C , then it should not be counted towards O_{11} but it should still count towards both marginal frequencies R_1 and C_1 – and, like every block, it should only be counted once. The second workaround is to count such blocks as half an instance towards both O_{12} and O_{21} . Over the whole corpus, that is equivalent to randomly assigning each of those subgraphs to either O_{12} or O_{21} . The result is that the subgraphs count equally towards R_1 and C_1 while the sample size N is kept stable and the frequency counts in O_{11} are reserved for real instances of G_C . For implementing this counting scheme, we define the four indicator variables $I_{11}^{(s,k)}$ to $I_{22}^{(s,k)}$ that indicate how a given block $F_N^{(s,k)}$ is classified.:

$$\begin{aligned} I_{11}^{(s,k)} &= \begin{cases} 1 & \text{if } |H_C^{(s,k)}| \geq 1 \\ 0 & \text{otherwise} \end{cases} \\ I_{12}^{(s,k)} &= \begin{cases} 1 & \text{if } |H_A^{(s,k)}| \geq 1 \text{ and } |H_B^{(s,k)}| = 0 \\ 0.5 & \text{if } |H_A^{(s,k)}| \geq 1 \text{ and } |H_B^{(s,k)}| \geq 1 \text{ and } |H_C^{(s,k)}| = 0 \\ 0 & \text{otherwise} \end{cases} \\ I_{21}^{(s,k)} &= \begin{cases} 1 & \text{if } |H_A^{(s,k)}| = 0 \text{ and } |H_B^{(s,k)}| \geq 1 \\ 0.5 & \text{if } |H_A^{(s,k)}| \geq 1 \text{ and } |H_B^{(s,k)}| \geq 1 \text{ and } |H_C^{(s,k)}| = 0 \\ 0 & \text{otherwise} \end{cases} \\ I_{22}^{(s,k)} &= \begin{cases} 1 & \text{if } |H_A^{(s,k)}| = 0 \text{ and } |H_B^{(s,k)}| = 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

The values $O_{11}^{(s)}$ to $O_{22}^{(s)}$ for a sentence G_s can be obtained by summing over all the blocks:

$$O_{ij}^{(s)} = \sum_k I_{ij}^{(s,k)}$$

The overall frequencies can be calculated by summing over all the sentences in the corpus:

$$o_{ij} = \sum_s o_{ij}^{(s)} = \sum_{sk} I_{ij}^{(s,k)}$$

The sample size N can be determined by summing over the four cells of the contingency table or, equivalently, by counting all the blocks for all sentences in the corpus:

$$N = \sum_{ij} o_{ij} = \sum_s |P_s|$$

4.6 Special cases

As described in Sections 4.3 and 4.5, the generalized cooccurrence model is an extension of existing models. Therefore, collocations and collostructions are merely special cases of the generalized cooccurrence model that occur naturally if the relevant queries are submitted. To indicate the cooccurrence we are interested in, it is sufficient to specify the cooccurring linguistic structures G_A and G_B and the way in which they cooccur, i. e. G_C . G_N can be automatically derived as described in Section 4.5.3.

4.6.1 Collocation as a special case

Relational cooccurrences

Relational cooccurrences (cf. Section 2.1.2) were the starting point for the preliminary cooccurrence model introduced in Section 4.3. The term is used for the cooccurrence of two items that are connected via some relation, e. g. the lemma *lay* governing the lemma *table* via a direct object dependency relation (G_C , Figure 4.41c). That cooccurrence can be interpreted as the cooccurrence of linguistic structures G_A (*lay* with a direct object) and G_B (*table* as a direct object). The two structures overlap completely and taking the overlapping part, i. e. both vertices and the edge between them, using the most general labels from the two structures, yields G_N .

Classifying the instances of G_N results in a complete reconstruction of relational cooccurrences if we either count and classify embeddings or subgraphs. The two counting methods are fully equivalent for those simple linguistic structures and we can neither have inflated frequencies (except where the original method has them, too (cf. Evert, 2004: 35)) nor problems with inconsistencies. If we count and classify appropriately chosen focus points,

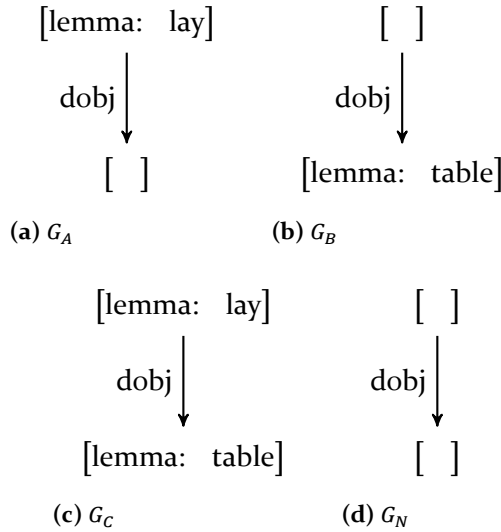


Figure 4.41: Query graphs for relational cooccurrences

we even get rid of the cases where relational cooccurrences yield inflated frequencies (cf. Section 4.4).

Segment-based cooccurrences

Although we did not use them to derive the generalized cooccurrence model, we can still reconstruct segment-based cooccurrences (cf. Section 2.1.1) using that model. However, we need to have the proper annotation in place to do that. Segment-based cooccurrences look at the cooccurrence of two items within one segment, e. g. a sentence. One way to reconstruct that approach using the generalized cooccurrence model is to add vertices for the segments and to connect all tokens within a segment to the corresponding segment vertex. In Figure 4.42e, for example, we added a vertex for a sentence-level segment and connected all tokens in the sentence to this vertex.

The two items we are interested in are G_A , the lemma *lay* within a segment, and G_B , the lemma *table* within a segment. The two should cooccur in such a way that the segment is the same (G_C). The overlapping part of the two linguistic structures is the segment vertex (G_N), i. e. we are classifying all the segments, resulting in a complete reconstruction of segment-based cooccurrences. Since G_N is just a single vertex, all counting methods are

fully equivalent¹⁸ and there are no problems with inflated frequencies or inconsistencies.

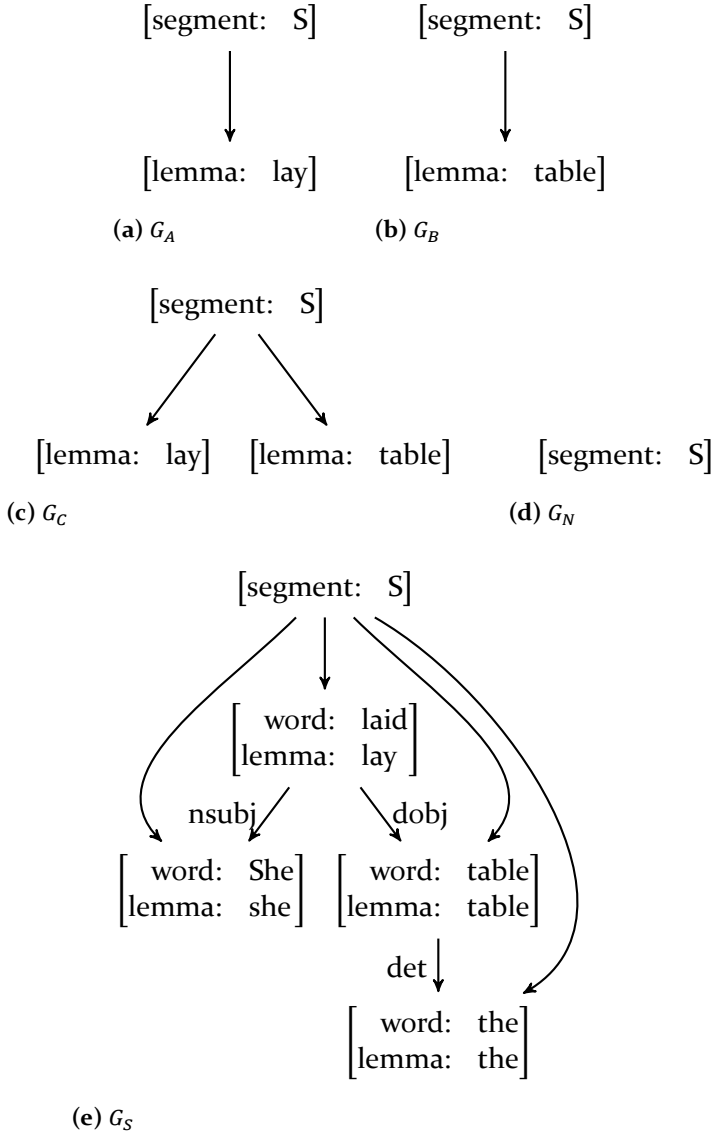


Figure 4.42: Query graphs for segment-based cooccurrences

¹⁸ Note that the counting methods are only equivalent because we are classifying sentence-level segments here. If the we have different segments, e. g. phrases, only three of the four counting methods are equivalent (embeddings, subgraphs, focus points) and the classification of whole graphs will yield different results.

4.6.2 Collostruction as a special case

Simple collexeme analysis

Simple collexeme analysis (Section 2.2.1) has been the starting point for the generalization of the preliminary model to the generalized cooccurrence model (cf. Section 4.5.1). In simple collexeme analysis, we are interested in the cooccurrence of a single word with a syntactic structure. The two cooccurring structures G_A and G_B as well as their cooccurrence G_C can be modeled straightforwardly (Figure 4.43). The overlapping part, the instances of which will be classified, is a single vertex (G_N). Therefore, counting and classifying embeddings, subgraphs and focus point vertices yields fully equivalent results and there are no problems with inflated frequencies or inconsistencies. The application of any of those three counting methods results in a complete reconstruction of simple collexeme analysis.

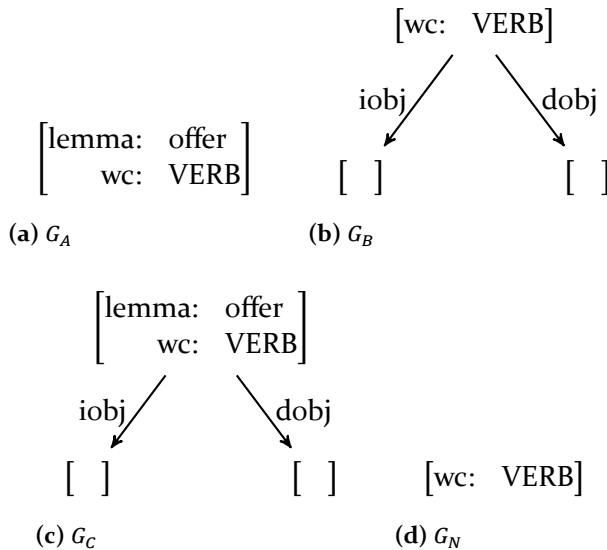


Figure 4.43: Query graphs for simple collexeme analysis

Covarying collexeme analysis

Covarying collexeme analysis (cf. Section 2.1.2) has been discussed as an intermediate step from relational cooccurrences to the preliminary cooccurrence model (cf. Section 4.3.2). As such, the cooccurrence of two words, e.g. the verbs *trick* and *believe*, within some syntactic structure, e.g. the

into-causative, as shown in Figure 4.44c, can be interpreted as the cooccurrence of the linguistic structures shown in Figures 4.44a and 4.44b. If we classify the instances of the derived linguistic structure G_N , we arrive at a reconstruction of covarying collexeme analysis. However, it is not entirely clear, which counting method corresponds to an exact reconstruction because the original counting method is not explained (cf. Section 2.2.1). The two most sensible options for a reconstruction are probably subgraphs and focus points.

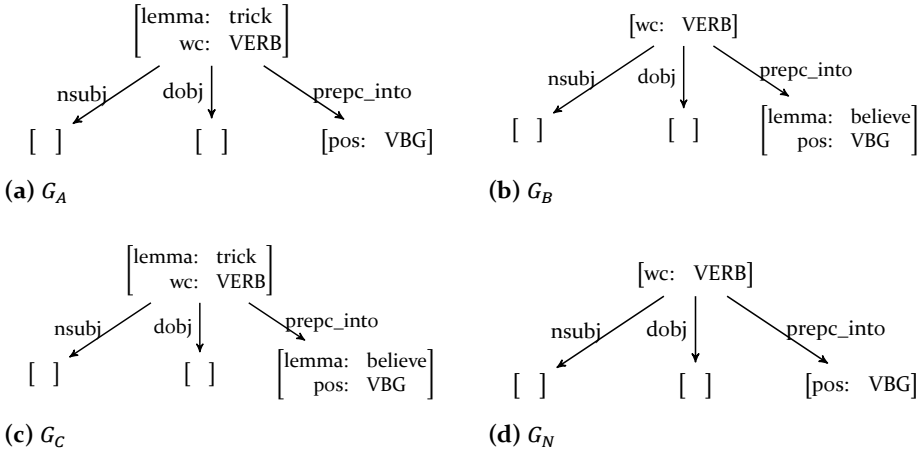


Figure 4.44: Query graphs for covarying collexeme analysis

4.7 Seeming limitations

The generalized cooccurrence model introduced in Section 4.5 is very flexible and powerful. Established ways of analyzing cooccurrence data, e. g. collocations or collostructions, can be regarded as mere special cases of that model (cf. Section 4.6). However, there still seem to be a couple of limitations.

The main characteristic of the general cooccurrence model that could be seen as a limitation is that a cooccurrence has to be a single linguistic structure, i. e. a single connected graph. That means we have to specify exactly and explicitly how two structures cooccur. Should the two be connected via some dependency relation? Should they cooccur within some segment? Which vertices and edges do they share? However, this is not really a limitation. If two items cooccur, there has to be *some* connection between them – and we need to operationalize that connection anyway, otherwise we would not be

able to tell a cooccurrence from a non-cooccurrence. Since we can model such connections as edges, requiring those connections to be stated explicitly is not a disadvantage – they need to be made explicit anyway. By requiring this explicitness, the model makes sure that the results are reproducible: It is clear how the two structures should cooccur and what is being counted. If we experience difficulties in making our queries that explicit, then this is not a problem of the model but a sure sign that we do not quite know what exactly we are looking for.

One thing that might not seem possible in the generalized cooccurrence model is to have “indirect” relations, where, for example, some word *A* governs some other word *B* either directly via a dependency relation from *A* to *B* or indirectly via more than one dependency relation with other vertices in between.

Another seemingly problematic issue might be the linear order of the words.

Finally, there is a conceivable use case where we are interested in, let’s say, monotransitive uses of the lemma *give*, i. e. instances of *give* with a direct object but without an indirect object. It might not be obvious how we can avoid finding ditransitive uses when we are looking for monotransitive uses.

All those issues can be dealt with by adding additional vertices and edges to the linguistic structures. Some of the ideas have already been implemented in CWB-treebank (Proisl and Uhrig, 2012), the graph searching backend of the Treebank.info project.¹⁹

4.7.1 Solution: Additional vertices and edges

When we were reconstructing segment-based cooccurrences in Section 4.6.1, we introduced new vertices representing segments and connected all words within a segment to the corresponding segment vertex (cf. Figure 4.42e). That strategy of introducing additional vertices and edges can also be applied to overcome the seeming limitations of the generalized cooccurrence model mentioned above.

We can introduce additional edges between existing vertices to explicitly model “indirect” relations as illustrated in Figure 4.45. That way, it becomes possible to look, for example, for instances where the lemma *undergo* governs the lemma *government* either directly or indirectly.

Additional edges between existing vertices can also be used to implement restrictions on the linear order of the vertices in a graph. Adding an edge

¹⁹ <http://treebank.info/>

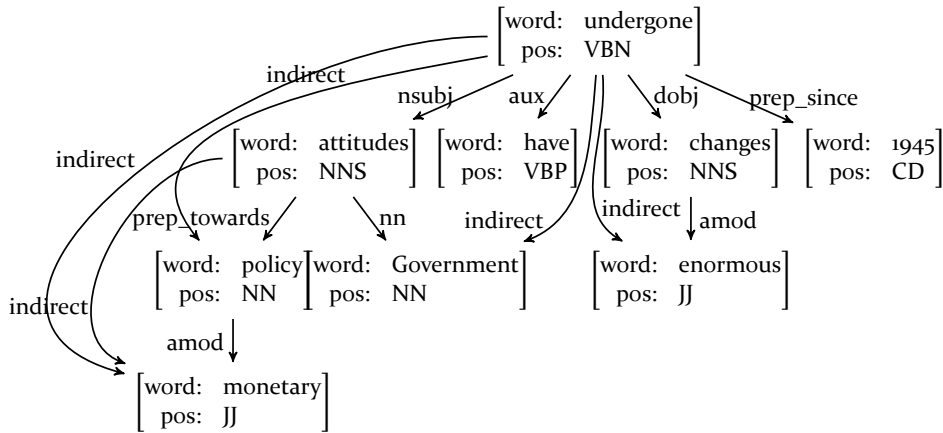


Figure 4.45: Representation of example sentence 6 with added edges for indirect relations

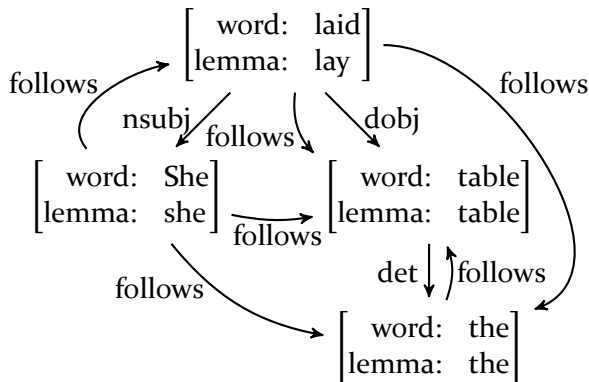


Figure 4.46: Representation of an example sentence with added edges for linear order

from every vertex in a sentence to all the vertices that follow it either directly or indirectly (cf. Figure 4.46) allows us to look, for example, for information-packaging constructions like preposing, postposing and inversion where the word order differs from the default (cf. Huddleston and Pullum, 2002: 1363–1447).

It is also possible to overcome the last seeming limitation mentioned above, i.e. to look for vertices that are not part of a certain relation. To do that, we need to make explicit the types of edges not incident to a vertex. One way to achieve this is to add “negated” relations from each vertex to some

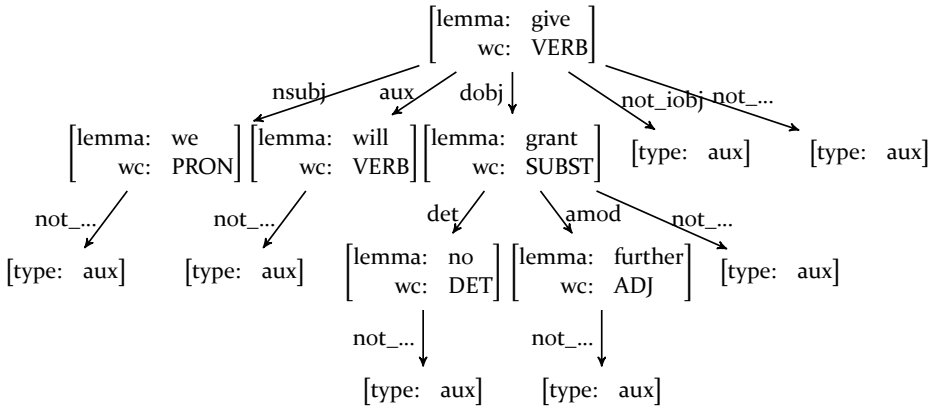


Figure 4.47: Representation of an example sentence with added “negated” edges for absent relations

auxiliary vertices.²⁰ For example, in a monotransitive use, the verb *give* would have an edge labeled *not_iobj* going to an auxiliary vertex (and of course a whole bunch of other such edges; cf. Figure 4.47; the label *not_...* is meant to indicate that there are many more negated relations). By converting a relation that was absent into another one that is present, we overcome the limitation.

That approach might not seem to be very elegant, but it is elegant in so far as the model itself remains unchanged, i. e. linguistic structures are still represented as labeled connected directed graphs. It is also important to note that the approach presented here serves predominantly the purposes of displaying what is possible within the formal model and of giving clear definitions – an implementation may do things differently. In an implementation we might not explicitly annotate things like “negated” edges but we might infer them instead from those edges that are present and the full set of relations.

If we are adding vertices and edges to a linguistic structure representing, for example, a dependency analysis of a sentence, it can happen that there are multiple edges from one vertex to another, i. e. that we have parallel edges. The resulting graphs are sometimes called multigraphs (Weisstein, n. d.: s. v. “Multigraph”²¹) or pseudographs (Weisstein, n. d.: s. v. “Pseudograph”²²).

²⁰ In the partially ordered set of labels, these “negated” relations are necessarily incomparable to the “regular” dependency relations.

²¹ <http://mathworld.wolfram.com/Multigraph.html>

²² <http://mathworld.wolfram.com/Pseudograph.html>

Note, however, that Definition 6 given on p. 59 is general enough to allow for multiple edges between vertices, i. e. the model is already well-equipped for such graphs.

4.8 Summary

In this chapter, we developed a generalized model for the cooccurrence of linguistic structures. To this end, we first introduced labeled connected directed graphs as a means for formally representing linguistic structures. The presentation included definitions of all necessary mathematical foundations, including the important concepts of subgraph isomorphism and subsumption. We went on to show that there is no straightforward solution to the problem of determining the frequency of non-trivial linguistic structures and presented four methods for counting linguistic structures as well as a generalized perspective on counting methods based on partitions of the set of embeddings.

Next, we developed a preliminary model that was an extension of relational cooccurrences and covarying collexeme analysis to arbitrary linguistic structures. Based on this model, we showed that counting and cross-classifying cooccurrences of linguistic structures can be problematic and that the different counting methods are – to varying degrees – prone to the problems of inflated frequencies (where the effects of combination and permutation lead to many non-independent instances) and inconsistency (where an instance is both an occurrence of A and of B but not a cooccurrence of the two).

By taking simple collexeme analysis into account and extending it to arbitrary linguistic structures, we developed the preliminary model into the generalized cooccurrence model. Based on the generalized perspective on counting methods developed earlier, we gave a complete formal definition of the generalized cooccurrence model.

We illustrated that relational and segment-based cooccurrences and simple and covarying collexeme analysis are merely special cases of the newly developed model. The final section showed how seeming limitations of the generalized cooccurrence model can be dealt with.

5 Evaluation of the methodology

Frameworks for the extraction of collocation candidates, i. e. word cooccurrences, can easily be evaluated against gold standard data derived from collocation dictionaries. Valency dictionaries might serve a similar purpose for the evaluation of approaches like simple collexeme analysis, i. e. cooccurrences of words with syntactic structures. For the cooccurrence of arbitrary structures, no gold standard is available, rendering a conventional evaluation unfeasible. What we can do instead is assess the robustness of the methodology developed in Chapter 4 based on theoretical considerations and empirical investigations.

The previous chapter introduced the generalized cooccurrence model and pointed out that there are different methods for obtaining frequency counts for linguistic structures. These counting methods are – to differing degrees – affected by two problems (cf. Section 4.4): 1) Under certain circumstances, a counting method can yield inflated frequencies by counting combinations and permutations of non-independent linguistic structures (inflated frequency problem). 11) Under certain circumstances, not all instances found by a counting method can be consistently classified into a 2×2 contingency table (inconsistency problem). The aim of this chapter is to find out whether those two problems play a role when performing actual queries against a corpus of real language data. To this end, we will first investigate how different association measures react to inflated frequencies by performing two thought experiments (Section 5.1). We will see that we can build groups of association measures that show similar behavior. Representatives of these groups will then be used as association measures in three experiments where we perform large numbers of queries in order to find out how problematic inflated frequencies and inconsistencies really are (Section 5.2).

5.1 Association measures and sample size

As has been shown in Section 4.4, some counting methods (counting embeddings or subgraphs) are prone to the problem of inflated frequencies. This means that the frequency counts for a given linguistic structure in a given

sentence can be drastically increased due to the effects of combination and permutation.

In this section, we conduct two thought experiments to estimate the effect of those inflated frequencies on different association measures. The thought experiments cover two scenarios: In the first, we assume that the inflated frequencies affect all cells of the contingency table equally (Section 5.1.1), in the second, we assume that, while the sample size is increased, the amount of positive evidence is kept stable, i. e. that the inflated frequencies only affect the cell O_{22} of the contingency table (Section 5.1.2).

5.1.1 Thought experiment 1

In this thought experiment, we assume that the problem of inflated frequencies affects all cells of the contingency table equally, i. e. that each cell O_{ij} is increased by a factor of n . Consequently, the sample size N and all expected frequencies E_{ij} are increased by the same factor. This thought experiment has a very intuitive interpretation, since it is equivalent to concatenating the corpus n times. Due to this equivalence to an increase in corpus size, the thought experiment is also relevant to the criticism of statistical hypothesis testing summarized in Section 3.4, in particular to the fact that, using for example the chi-squared test, even tiny deviations from the null hypothesis can become statistically significant if the sample is large enough.

In the remainder of this section, we look at the association measures introduced in Chapter 3 and analyze how they respond to the increased frequencies. Association scores for the n -fold increased frequencies will be marked with the subscript n .

Invariant measures

Some measures are completely insensitive to the frequency increase. Measures based on the two conditional probabilities $P(G_C|G_A)$ and $P(G_C|G_B)$ (cf. Section 3.3.4), for example, do not change at all, since the two conditional probabilities do not change:

$$P(G_C|G_A)_n = \frac{n \cdot O_{11}}{n \cdot R_1} = \frac{O_{11}}{R_1} = P(G_C|G_A)$$

$$P(G_C|G_B)_n = \frac{n \cdot O_{11}}{n \cdot C_1} = \frac{O_{11}}{C_1} = P(G_C|G_B)$$

Therefore, measures like the Dice or Jaccard coefficients, or other measures combining these two conditional probabilities, using for example the

geometric mean or the minimum as link function, are not affected at all if the frequencies in each cell of the contingency table are equally increased.

The same is true for phi or Cramér's V (cf. Section 3.4.1), another measure of effect size that is obtained by taking the square root of the chi-squared score divided by the sample size:

$$\begin{aligned}\phi_n = V_n &= \sqrt{\sum_{ij} \frac{(nO_{ij} - nE_{ij})^2}{nN \cdot nE_{ij}}} = \sqrt{\sum_{ij} \frac{n^2(O_{ij} - E_{ij})^2}{n^2 R_i C_j}} = \sqrt{\sum_{ij} \frac{(O_{ij} - E_{ij})^2}{R_i C_j}} \\ &= \phi = V\end{aligned}$$

Sublinearly scaling measures

The z-score and t-score measures (cf. Section 3.3.3) scale sublinearly with n . More precisely, if the frequencies in all cells of the contingency table are increased by a factor of n , then the z-score and t-score measures increase by a factor of $\sqrt{n}$:

$$\begin{aligned}\text{z-score}_n &= \frac{n \cdot O_{11} - n \cdot E_{11}}{\sqrt{n \cdot E_{11}}} = \frac{n \cdot (O_{11} - E_{11})}{\sqrt{n} \cdot \sqrt{E_{11}}} = \sqrt{n} \cdot \frac{O_{11} - E_{11}}{\sqrt{E_{11}}} \\ &= \sqrt{n} \cdot \text{z-score}\end{aligned}$$

$$\begin{aligned}\text{t-score}_n &= \frac{n \cdot O_{11} - n \cdot E_{11}}{\sqrt{n \cdot O_{11}}} = \frac{n \cdot (O_{11} - E_{11})}{\sqrt{n} \cdot \sqrt{O_{11}}} = \sqrt{n} \cdot \frac{O_{11} - E_{11}}{\sqrt{O_{11}}} \\ &= \sqrt{n} \cdot \text{t-score}\end{aligned}$$

Linearly scaling measures

The chi-squared and log-likelihood measures (cf. Section 3.3.2) scale linearly with n . If we double all the frequencies, then the chi-squared and log-likelihood scores will also double.

For chi-squared, this follows directly from the facts that phi is invariant to the frequency increase and that chi-squared = $N\phi^2$. Therefore, an n times bigger sample size N results in an n times bigger chi-squared score.

Here is a more detailed derivation for the log-likelihood measure:

$$\begin{aligned}\log\text{-likelihood}_n &= 2 \sum_{ij} n \cdot O_{ij} \log \frac{n \cdot O_{ij}}{n \cdot E_{ij}} = n \cdot 2 \sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}} \\ &= n \cdot \log\text{-likelihood}\end{aligned}$$

5.1.2 Thought experiment 2

In the second thought experiment, the amount of positive evidence, i. e. the frequencies in O_{11} , O_{12} and O_{21} , is kept stable and only the amount of seemingly irrelevant material, i. e. O_{22} , is increased. When O_{22} approaches infinity, the marginal frequencies R_2 and C_2 and the sample size N are dominated by O_{22} and approach the following limits:

$$\lim_{O_{22} \rightarrow \infty} R_2 \approx O_{22} = \infty$$

$$\lim_{O_{22} \rightarrow \infty} C_2 \approx O_{22} = \infty$$

$$\lim_{O_{22} \rightarrow \infty} N \approx O_{22} = \infty$$

Since N approaches infinity and $\frac{R_2}{N}$ and $\frac{C_2}{N}$ approach 1, the expected frequencies E_{11} to E_{22} approach the following limits:

$$\lim_{O_{22} \rightarrow \infty} E_{11} = \frac{R_1 C_1}{N} = 0$$

$$\lim_{O_{22} \rightarrow \infty} E_{12} = \frac{R_1 C_2}{N} = R_1$$

$$\lim_{O_{22} \rightarrow \infty} E_{21} = \frac{R_2 C_1}{N} = C_1$$

$$\lim_{O_{22} \rightarrow \infty} E_{22} = \frac{R_2 C_2}{N} \approx O_{22} = \infty$$

Invariant measures

The two conditional probabilities $P(G_C|G_A)$ and $P(G_C|G_B)$ do not depend on O_{22} :

$$P(G_C|G_A) = \frac{O_{11}}{R_1}$$

$$P(G_C|G_B) = \frac{O_{11}}{C_1}$$

Therefore, measures like the Dice or Jaccard coefficients, or other measures combining these two conditional probabilities, using for example the geometric mean or the minimum as link function (cf. Section 3.3.4), are not affected at all if the frequency counts in cell O_{22} are increased.

Measures with a finite limit

Since E_{11} approaches 0 when O_{22} approaches infinity, the t -score measure (cf. Section 3.3.3) approaches $\sqrt{O_{11}}$:

$$\lim_{O_{22} \rightarrow \infty} t\text{-score} = \frac{O_{11} - E_{11}}{\sqrt{O_{11}}} = \frac{O_{11}}{\sqrt{O_{11}}} = \sqrt{O_{11}}$$

By looking at the formula for phi or Cramér's V (cf. Section 3.4.1), we can see that if $i = 2$ and/or $j = 2$, then the summand has a limit of 0 since the numerator is finite and the denominator approaches infinity. The limit of ϕ is the gmean association measure, i. e. the geometric mean of the two conditional probabilities $P(G_C|G_A)$ and $P(G_C|G_B)$ that express the probability of observing a cooccurrence given one of the two elements (cf. Section 3.3.4):

$$\begin{aligned} \lim_{O_{22} \rightarrow \infty} \phi = V &= \sqrt{\sum_{ij} \frac{(O_{ij} - E_{ij})^2}{NE_{ij}}} = \sqrt{\sum_{ij} \frac{(O_{ij} - E_{ij})^2}{R_i C_j}} = \sqrt{\frac{O_{11}^2}{R_1 C_1}} \\ &= \frac{O_{11}}{\sqrt{R_1 C_1}} \end{aligned}$$

Measures without a finite limit

Finally, there are also measures that increase without a finite limit as O_{22} is increased. The z -score, for example, approaches infinity since its denominator approaches 0:

$$\lim_{O_{22} \rightarrow \infty} z\text{-score} = \frac{O_{11} - E_{11}}{\sqrt{E_{11}}} = \frac{O_{11}}{\sqrt{E_{11}}} = \infty$$

As shown above, phi approaches a finite limit. Therefore, chi-squared approaches infinity as O_{22} is increased:

$$\lim_{O_{22} \rightarrow \infty} \text{chi-squared} = N\phi^2 = \infty$$

When determining the behavior of the log-likelihood measure (cf. Section 3.3.2), it is best to look at each of the summands $O_{ij} \log \frac{O_{ij}}{E_{ij}}$ in turn:

$$\lim_{O_{22} \rightarrow \infty} O_{11} \log \frac{O_{11}}{E_{11}} = \infty$$

$$\begin{aligned}
\lim_{O_{22} \rightarrow \infty} O_{12} \log \frac{O_{12}}{E_{12}} &= O_{12} \log \frac{O_{12}}{R_1} \\
\lim_{O_{22} \rightarrow \infty} O_{21} \log \frac{O_{21}}{E_{21}} &= O_{21} \log \frac{O_{21}}{C_1} \\
\lim_{O_{22} \rightarrow \infty} O_{22} \log \frac{O_{22}}{E_{22}} &= O_{22} \log \frac{O_{22}(O_{11} + O_{12} + O_{21} + O_{22})}{(O_{21} + O_{22})(O_{12} + O_{22})} \\
&= \frac{\log \frac{O_{22}(O_{11} + O_{12} + O_{21} + O_{22})}{(O_{21} + O_{22})(O_{12} + O_{22})}}{\frac{1}{O_{22}}} \\
&\stackrel{\text{LH}}{=} \frac{-\frac{O_{11}O_{22}^2 - 2O_{12}O_{21}O_{22} - O_{12}O_{21}^2 - O_{12}^2O_{21} - O_{11}O_{12}O_{21}}{O_{22}(O_{22} + O_{12})(O_{22} + O_{21})(O_{11} + O_{12} + O_{21} + O_{22})}}{-\frac{1}{O_{22}^2}} \\
&= \frac{O_{22}(O_{11}O_{22}^2 - 2O_{12}O_{21}O_{22} - O_{12}O_{21}^2 - O_{12}^2O_{21} - O_{11}O_{12}O_{21})}{(O_{22} + O_{12})(O_{22} + O_{21})(O_{22} + O_{21} + O_{12} + O_{11})} \\
&= O_{11}
\end{aligned}$$

For $i = 1, j = 1$, the fraction, and therefore the logarithm and the whole summand, approach infinity. For $i = 1, j = 2$ and $i = 2, j = 1$, the summands approach a finite limit. For $i = 2, j = 2$, we set

$$E_{22} = \frac{R_2 C_2}{N} = \frac{(O_{21} + O_{22})(O_{12} + O_{22})}{O_{11} + O_{12} + O_{21} + O_{22}}$$

and apply L'Hôpital's rule (Weisstein, n. d.: s. v. "L'Hospital's Rule"¹), replacing the numerator and denominator with their derivatives. The resulting unwieldy fraction approaches a finite limit of O_{11} according to Wolfram Alpha.² Therefore, the sum of all four summands approaches infinity as O_{22} increases:

$$\lim_{O_{22} \rightarrow \infty} \log\text{-likelihood} = 2 \sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}} = \infty$$

5.1.3 Overview of results

The findings obtained through the two thought experiments are summarized in Table 5.1. The Dice and Jaccard coefficients stand out because they are not

¹ <http://mathworld.wolfram.com/LHopitalsRule.html>

² [https://www.wolframalpha.com/input/?i=limit+\(d*\(a*d%5E2-2*b*c*d-b*c%5E2-b%5E2*c-a*b*c\)%2F\(\(b%2Bd\)*\(c%2Bd\)*\(a%2Bb%2Bc%2Bd\)\)\)+as+d-%3Einfinity](https://www.wolframalpha.com/input/?i=limit+(d*(a*d%5E2-2*b*c*d-b*c%5E2-b%5E2*c-a*b*c)%2F((b%2Bd)*(c%2Bd)*(a%2Bb%2Bc%2Bd)))+as+d-%3Einfinity)

vulnerable to either of the two scenarios of inflated frequencies. Cramér’s V is not affected in the first scenario and has a finite limit that approaches the gmean association measure in the second. Both t -score and z -score scale sublinearly with corpus size in the first scenario, but while t -score approaches a finite limit in the second scenario, z -score does not. Both the chi-squared and the log-likelihood measures scale linearly with corpus size in the first scenario and do not have a finite limit in the second. The order in which the association measures are presented in Table 5.1 reflects how affected the association measures are by inflated frequencies, with the most robust measures at the top and the most vulnerable at the bottom.

measure	scenario 1	scenario 2
Dice	invariant	invariant
Jaccard	invariant	invariant
ϕ /Cramér’s V	invariant	finite limit
t -score	sublinear	finite limit
z -score	sublinear	no finite limit
chi-squared	linear	no finite limit
log-likelihood	linear	no finite limit

Table 5.1: Behavior of association measures in the scenarios discussed in Sections 5.1.1 and 5.1.2

5.2 Experiments with frequent subgraphs

The two scenarios for inflated frequencies discussed in the previous section have shown that the various association measures behave very differently. The question is whether the theoretical problem of inflated frequencies is also problematic when working with real language data. To answer this question, we generate cooccurrence queries based on frequent subgraphs and calculate association strengths using three different association measures for each of the four counting methods. Ideally, the four counting methods should produce the same ranking of cooccurrences when using the same association measure. If inflated frequencies are a problem that occurs in practice and if the association measure used reacts to inflated frequencies, then we should see a difference in the distribution of association scores. While performing the queries, we also register all cases of inconsistent frequency counts (cf. Section 4.4).

We repeat this experiment three times: For the first run, we use completely lexicalized queries. For the second, we modify the queries so that they are partly unlexicalized. For the third run, we delexicalize even greater parts of the queries. If inflated frequencies are indeed a problem in practice, we can expect that problem to become more severe the more abstract the queries are.

The three association measures are selected in such a way that they represent each of the groups identified in the previous section: I) The Dice coefficient is not affected in either of the two scenarios for inflated frequencies, II) the t -score measure is sublinearly scaling in scenario 1 and has a finite limit in scenario 2 and III) the log-likelihood measure is linearly scaling in scenario 1 and has no finite limit in scenario 2. Incidentally, the three chosen measures are probably also the most popular choices for collocation extraction.

For the cooccurrence queries, we want to identify frequent subgraphs that satisfy the following desiderata. The complete structures should have more than two vertices, otherwise we would be looking at relational cooccurrences. For each structure G_C , we have to assign elements to G_A and G_B in such a way that both are connected linguistic structures and that the overlapping part is also connected. The overlapping part should also have at least 2 vertices, because for a single vertex three of the four counting methods would yield identical results (for a single vertex, it makes no difference if we count embeddings, subgraphs or focus points). As a rule, to keep the queries as independent from each other as possible, we do not want to use frequent subgraphs that are subgraphs of other frequent subgraphs. By a “frequent subgraph” we mean one that occurs in at least n sentences. Given that the probability for multiple occurrences of a linguistic structure in a corpus will be the lower the larger the structure is, we will have to settle for relatively low values of n .

The exact procedure for extracting the frequent subgraphs and turning them into queries is described in the following sections.

The NetworkX library (Hagberg et al., 2008) has been invaluable for conducting the experiments.

5.2.1 Extraction of frequent subgraphs

The starting point for our extraction of frequent subgraphs are words that are likely to show “interesting” combinatorial behavior. As a source for such interesting words we use the Oxford Collocations Dictionary (McIntosh et al., 2009) that contains approximately 250 000 word combinations involving

adjectives, adverbs, nouns and verbs. For each of the four word classes we draw a sample of 125 words such that almost the whole frequency range of the word class (as determined on the basis of the BNC) is covered. To avoid extremely rare and extremely common words, the lowest and highest 5% of frequencies are excluded.

For each of the 500 words, we first collect all subgraphs from the parsed BNC that consist of 4 to 6 vertices and contain the word. We then filter out all subgraphs that occur in less than 5 sentences. This results in 97 221 distinct subgraphs for 358 words. For 142 words there were no subgraphs of sufficient size and frequency.

Many of these 97 221 subgraphs are very similar to each other because they are subgraphs of larger frequent structures. To get rid of these near-duplicates, we calculate an estimate for the similarity between subgraphs that, for the sake of simplicity, ignores the graph structure. This estimate is identical to the Jaccard coefficient under the assumption that vertex and edge labels are disjoint:

$$\text{sim}(G_1, G_2) = \frac{|\{\lambda(u)|u \in V_1\} \cap \{\lambda(v)|v \in V_2\}| + |\{\lambda(d)|d \in E_1\} \cap \{\lambda(e)|e \in E_2\}|}{|\{\lambda(u)|u \in V_1\} \cup \{\lambda(v)|v \in V_2\}| + |\{\lambda(d)|d \in E_1\} \cup \{\lambda(e)|e \in E_2\}|}$$

Since our similarity measure is the Jaccard coefficient, we can turn it into a distance metric by subtracting it from 1 (cf. Section 3.3.4). The resulting distances range from 0 to 1.

Now that we know the distances between subgraphs, we use the DBSCAN algorithm (Ester et al., 1996) as implemented in scikit-learn (Pedregosa et al., 2011) to group similar subgraphs together.³ The choice of DBSCAN is motivated by the fact that, as a density-based clustering algorithm, it does not need to be provided with a parameter for the desired number of clusters. For each resulting cluster, we construct the complete distance graph and select the item, i. e. the subgraph, with the highest closeness centrality (Freeman, 1978). This results in 3 440 subgraphs. The order distribution of these subgraphs, i. e. the number of subgraphs with 4, 5 or 6 vertices, is shown in Table 5.2.

³ We set the parameter `eps` that indicates “[t]he maximum distance between two samples for them to be considered as in the same neighborhood” to 0.35 and the parameter `min_samples` for “[t]he number of samples [...] in a neighborhood for a point to be considered as a core point” to 3 (<http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html>).

vertices	graphs
4	2381
5	686
6	373

Table 5.2: Distribution of extracted subgraphs

5.2.2 Query creation

For our purposes, we interpret each of the 3 440 subgraphs extracted above as the cooccurrence of two linguistic structures, i. e. as a linguistic structure G_C . The two linguistic structures G_A and G_B that cooccur in linguistic structure G_C are randomly determined according to the following procedure: First, we create V_l , the set of vertices that are part of both G_A and G_B and that make up G_N , by randomly choosing between 2 and $|V_C| - 1$ vertices from V_C such that the induced subgraph $G_C\langle V_l \rangle$ is connected. Then we create the set $V_{\alpha'}$ by randomly choosing between 0 and $|V_C| - |V_l|$ vertices from $V_C \setminus V_l$ such that the induced subgraphs $G_C\langle V_{\alpha} \rangle$ and $G_C\langle V_{\beta} \rangle$ based on the vertex sets $V_{\alpha} = V_{\alpha'} \cup V_l$ (the set of vertices that make up G_A) and $V_{\beta} = V_C \setminus V_{\alpha'}$ (the set of vertices that make up G_B) are connected. The resulting induced subgraphs constitute the two cooccurring linguistic structures: $G_A = G_C\langle V_{\alpha} \rangle$ and $G_B = G_C\langle V_{\beta} \rangle$.

As an example, consider the subgraph for the fragment “could not be more different” (Figure 5.1). Two of the vertices, *different* and *be* get assigned to V_l . Two of the remaining three vertices, *not* and *more*, get assigned to $V_{\alpha'}$. The remaining vertex, *could*, is denoted as belonging to $V_{\beta'}$ in the figure, i. e. $V_{\beta'} = V_C \setminus (V_l \cup V_{\alpha'})$. In this example, V_{α} , the union of $V_{\alpha'}$ and V_l , consists of four vertices, *not*, *be*, *more* and *different*. The induced subgraph G_A that constitutes the first of our two cooccurring linguistic structures is visualized in Figure 5.2a. The second structure, G_B (Figure 5.2b), is the subgraph induced by V_{β} , the union of $V_{\beta'}$ and V_l . This set contains three vertices, *could*, *be* and *different*. The cooccurrence of G_A and G_B is G_C (Figure 5.2c), the original graph from Figure 5.1. The overlapping part of G_A and G_B , *be different*, constitutes G_N (Figure 5.2d).

5.2.3 Evaluation based on completely lexicalized subgraphs

For the first evaluation experiment, we use the queries created above “as is”, i. e. we look at cooccurrences of completely lexicalized linguistic structures where every vertex has labels for word form, lemma, part-of-speech tag and word

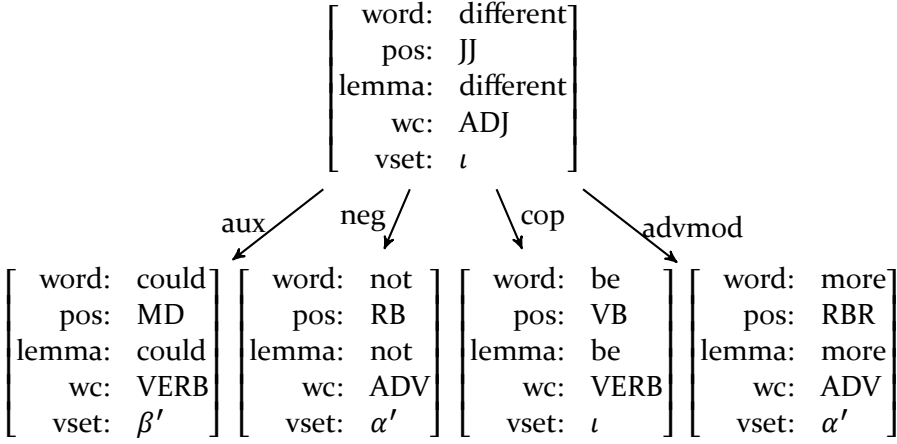


Figure 5.1: The vertices of the frequent subgraph have been randomly assigned to V_ι , $V_{\alpha'}$ and $V_{\beta'}$.

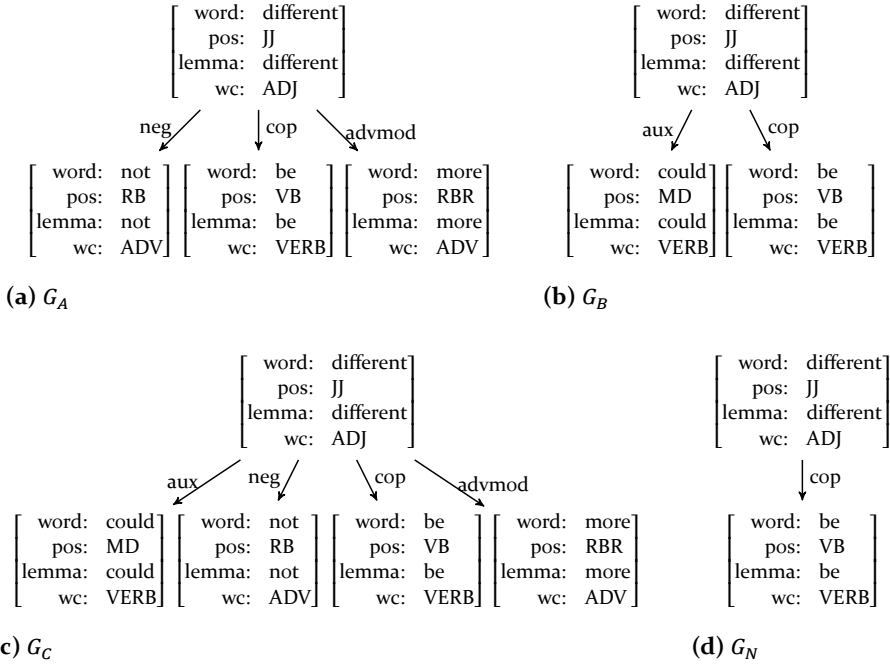


Figure 5.2: Query graphs for the structure "could not be more different" as a cooccurrence of "not be more different" and "could be different".

class. As an example, consider the query in Figure 5.2 where we look for the structure “could not be more different” (G_C) as a cooccurrence of “not be more different” (G_A) and “could be different” (G_B). The units of analysis that we need to classify are given by the structure G_N , “be different”. For the experiment, we determine the association strength for the cooccurrence of G_A and G_B for all 3 440 queries based on all four counting methods introduced in Chapter 4 using the three association measures Dice, t -score and log-likelihood. For every combination of counting method and association measure, we rank the queries according to the association strength.

Results

For this kind of query, the inconsistency problem is virtually non-existent (cf. Figure 5.3). The only counting method that is affected at all is the one that counts graphs. And even for that counting method, there are only three problematic queries and each of these queries yields only a single inconsistency, i. e. for each of the three queries there is only a single sentence that cannot be consistently classified. One of the affected queries, for example, asks for the association between “spent a lot” and “quite a lot” and there is one sentence that contains instances of both but not of the cooccurrence we are looking for, i. e. “spent quite a lot”.

We use the number of embeddings per focus point as a measure for inflated frequencies. Figure 5.4 indicates that inflated frequencies are also a non-issue. The only two queries that stand out are variants of a syntactic structure that originates from the sentence in Example 8:

- (8)_{BNC: B2C 1188} Numbers and letters as follows: 1 barn owl, 2 snowy owl, 3 long-eared owl, 4 short-eared owl, 5 Verreaux eagle owl, 6 spotted eagle owl, 7 European eagle owl, 8 great grey owl, 9 tawny owl, 10 little owl, 11 kestrel, 12 hen harrier, a arctic fox, b bat-eared fox, c coyote, f red fox, g small-spotted genet, m white tailed mongoose, p pine marten.

This sentence occurs six times in the BNC and its automatic syntactic analysis (making *barn owl* the direct object of *follows* and attaching four *owl* appositions to it) gave rise to a frequent subgraph consisting of the noun *owl* that is the direct object of the verb *follows* and that has two appositions: The noun *owl* and the noun *owl*. It should be clear from Section 4.4 that such a structure is susceptible to inflated frequencies when counting embeddings or subgraphs. In this particular case, there are $\binom{4}{2} = 6$ matching subgraphs and $\binom{4}{2}! = 12$ embeddings of the query in a single instance of this sentence – but we only have a single instance when we are counting focus points or graphs.

5.2 Experiments with frequent subgraphs

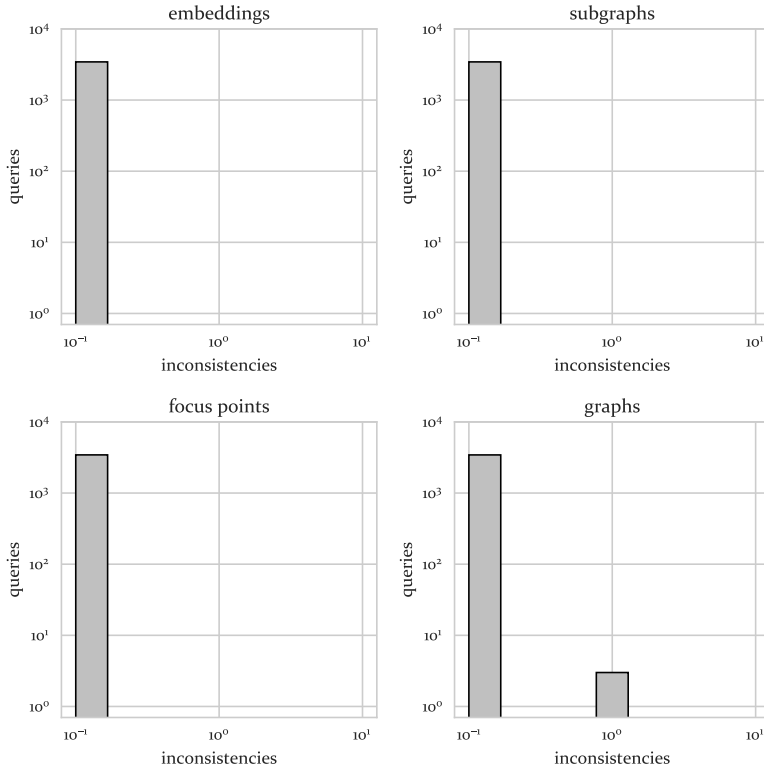


Figure 5.3: Inconsistencies for completely lexicalized queries. Note that both axes are logarithmic.

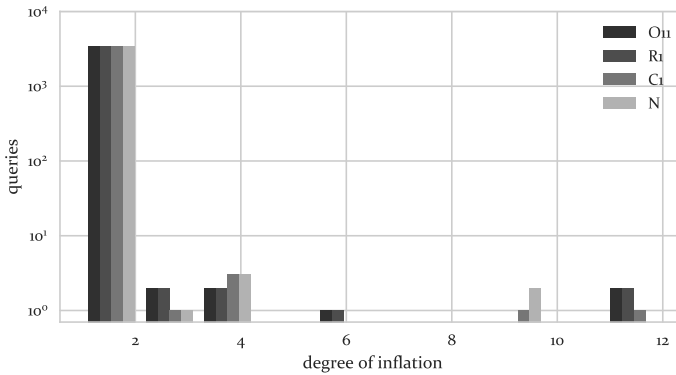


Figure 5.4: Degree of inflation for completely lexicalized queries (number of embeddings per focus point). Note that the y-axis is logarithmic.

As we have seen, inconsistencies and inflated frequencies are practically non-existent. The other major source for differences between the rankings produced by the different counting methods are the counting methods themselves. For example, if there are multiple matching subgraphs in a sentence, the counting methods will yield different frequencies, thereby leading to different rankings of the queries. Given that our queries are completely lexicalized, we do not expect that to happen very frequently and therefore we can expect the different counting methods to produce more or less the same rankings, irrespective of the association measures used.

Figure 5.5 shows that all counting methods produce virtually identical rankings of the queries (Spearman correlation coefficients ≈ 1.00 in all cases) when using the Dice coefficient. The only cases where we can observe noticeably differences in the rankings involve counting focus points or graphs. These differences are due to the fact that in some cases one instance according to those two counting methods corresponds to more than one instance when we are counting embeddings or subgraphs. Consequently, we have differences in the contingency tables and therefore in the rankings.

For t -score, the overall picture is the same (cf. Figure 5.6). All counting methods produce virtually identical rankings with Spearman correlation coefficients ≥ 0.99 . As we would expect from the theoretical considerations in Section 5.1, we can observe ranking differences for the two queries that are affected by inflated frequencies. The third query that stands out does so because the frequency of its overlapping part is lower when counting graphs than when using the other counting methods. The resulting differences in the frequency signatures ((7, 7, 10, 10) vs. (7, 7, 10, 12)) are responsible for the differences in the rankings.⁴

The plots for log-likelihood are very similar to those for t -score (cf. Figure 5.7). Again, the rankings produced by the counting methods are virtually identical with Spearman correlation coefficients ≥ 0.99 . The outlier queries are the same as for t -score and for the same reasons.

As we can see from Figure 5.8, the little amount of inconsistencies and inflated frequencies does not influence the overall distribution of association scores.

5.2.4 Evaluation based on partly unlexicalized subgraphs

In the second evaluation experiment, we want to find out how the presence of underspecified elements in the query influences the number of inconsis-

⁴ Note that this case is an instance of the scenario described in Section 5.1.2: The amount of positive evidence is the same but the amount of “irrelevant” material is increased.

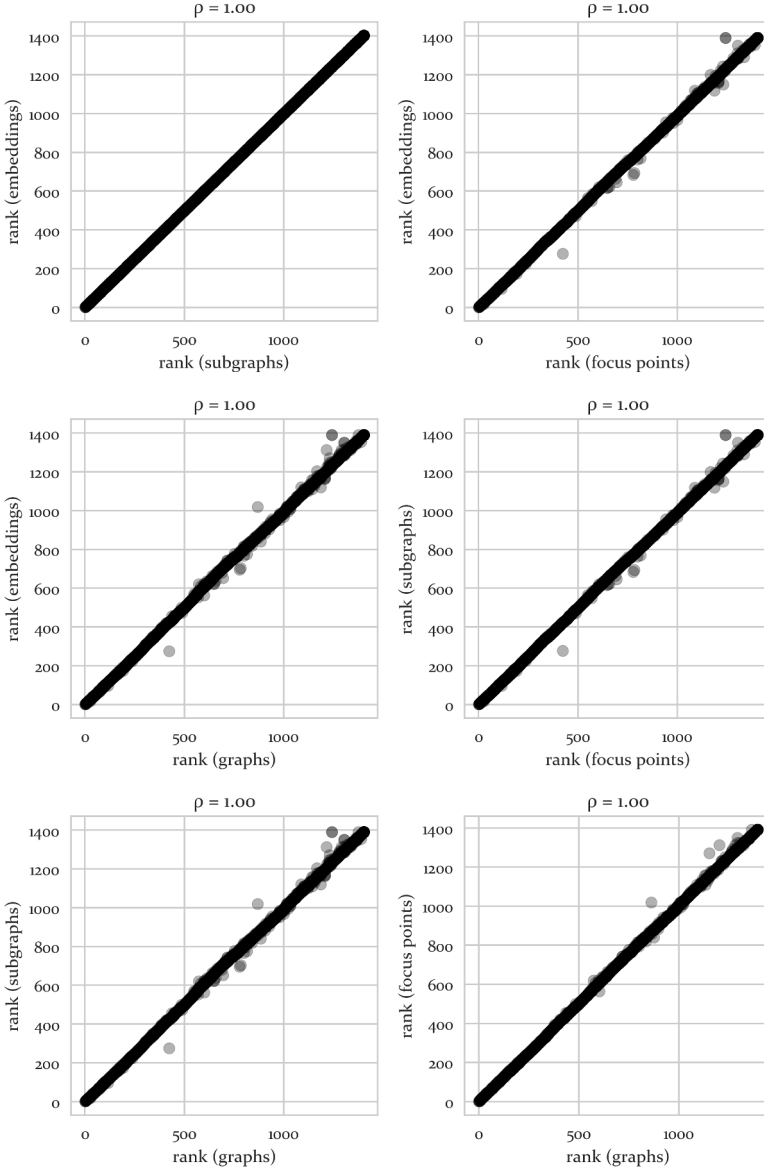


Figure 5.5: Scatterplots and Spearman correlation coefficients for different counting methods based on the Dice coefficient (completely lexicalized queries)

tendencies and the problem of inflated frequencies. To do that, we remove all word and lemma labels from the overlapping part of G_A and G_B . This means that both G_A and G_B are now partly unlexicalized and that G_N is completely

5 Evaluation of the methodology

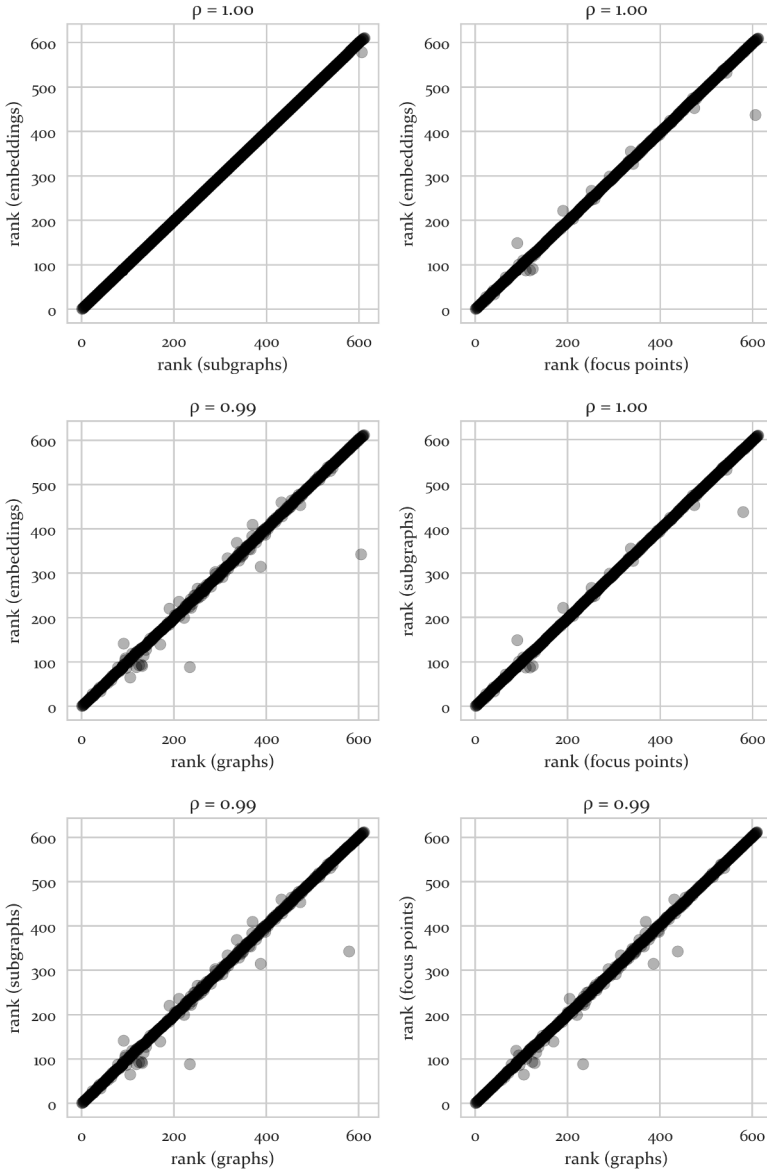


Figure 5.6: Scatterplots and Spearman correlation coefficients for different counting methods based on t -score (completely lexicalized queries)

unlexicalized, i. e. that all of its vertices have only part-of-speech tag and word class labels. For our example structure, “could not be more different”, this means that we remove the word and lemma labels from the *be* and *different*

5.2 Experiments with frequent subgraphs

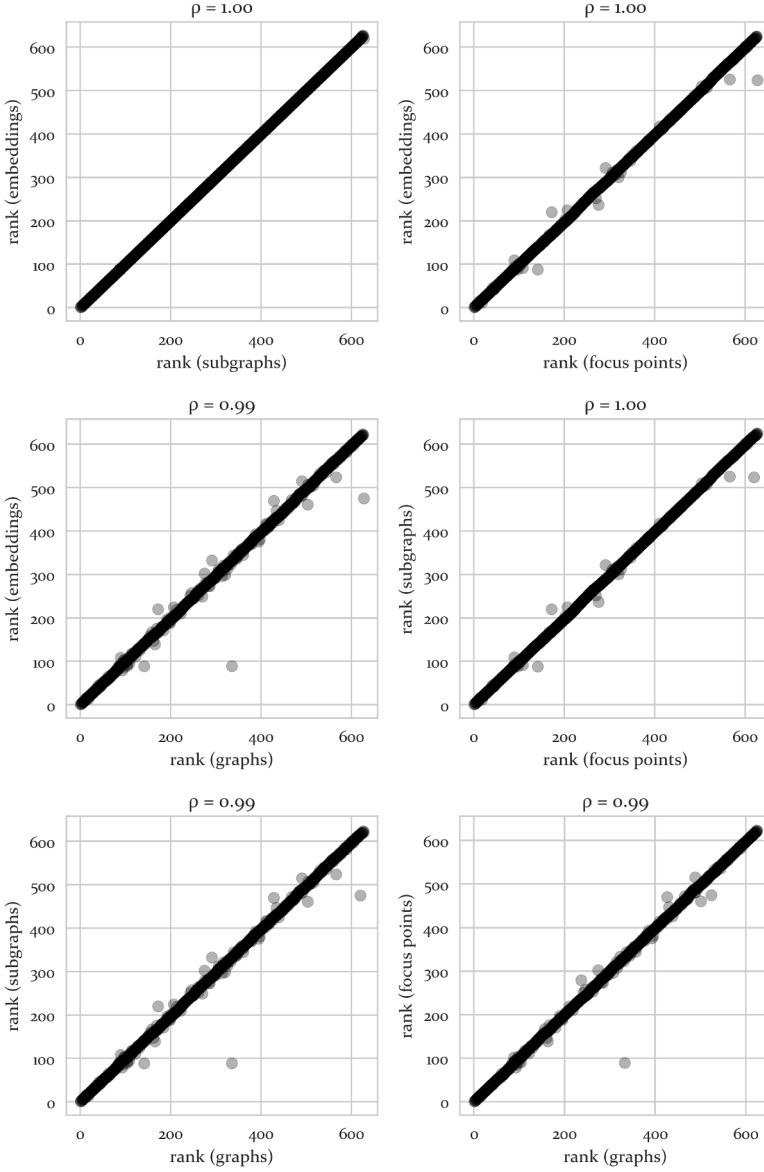


Figure 5.7: Scatterplots and Spearman correlation coefficients for different counting methods based on log-likelihood (completely lexicalized queries)

vertices, leading to the structure “could not VERB more ADJECTIVE”. The resulting query graphs are shown in Figure 5.9.

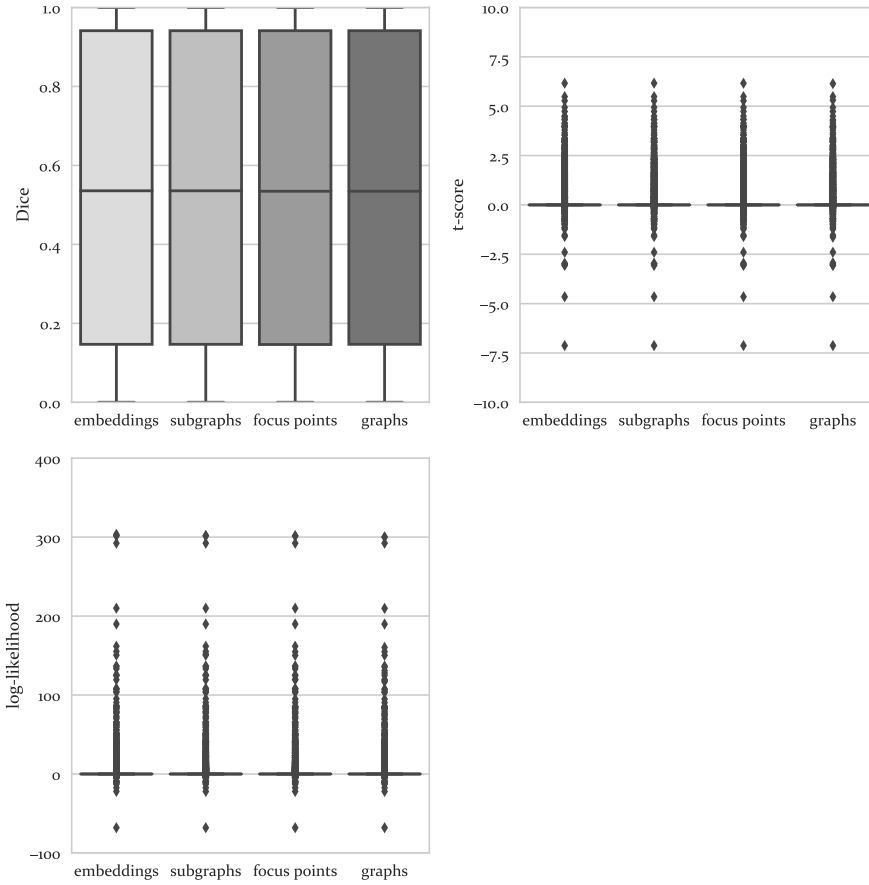


Figure 5.8: Distribution of association scores for different counting methods (completely lexicalized queries)

Apart from these changes in the queries, the experiment is identical to the previous one: We determine the association strength for all cooccurrences based on all four counting methods and the three association measures and rank the queries according to the association strength for every combination of counting method and association measure.

Results

Since the cooccurring linguistic structures are now less specific, we can observe more inconsistencies (cf. Figure 5.10). When counting focus points, there are 14 queries with inconsistencies, when counting graphs, there are

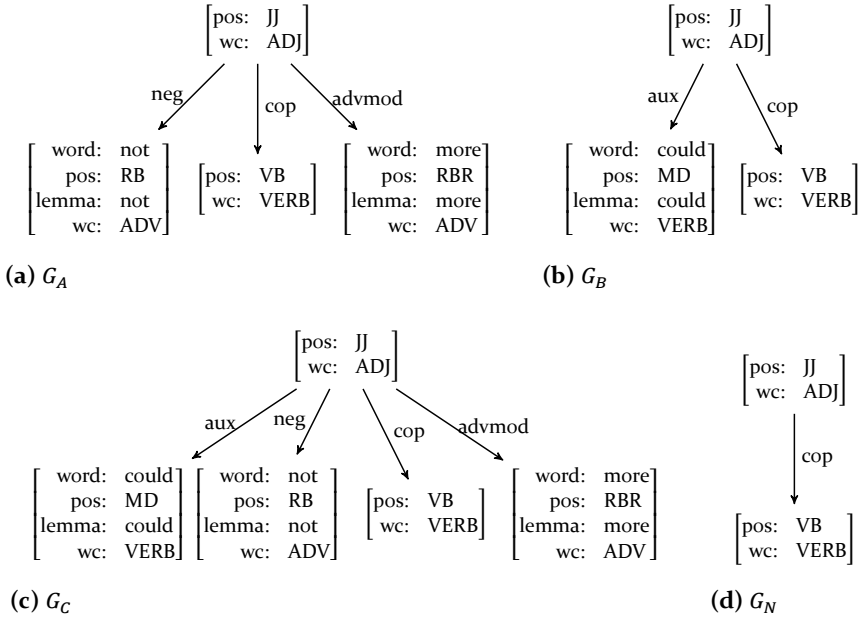


Figure 5.9: Query graphs for the structure “could not VERB more ADJECTIVE” as a cooccurrence of “not VERB more ADJECTIVE” and “could VERB ADJECTIVE”.

362. When counting embeddings or subgraphs, no inconsistent cases arise, even though they are theoretically possible. For most of the affected queries, there are only one or two cases where no consistent classification is possible. However, the plot shows that there are also some queries where many of the instances found in the corpus cannot be classified consistently.

The query with the highest number of inconsistencies when counting focus points is the one shown in Figure 5.11. For ease of reference, we will call it the “VERBed that he VERBed” structure (G_C , Figure 5.11c). What we want to determine is the association strength between “VERBed he VERBed” (G_A , Figure 5.11a) and “VERBed that VERBed” (G_B , Figure 5.11b) in this structure and to do that, we need to classify all instances of “VERBed VERBed” (G_N , Figure 5.11d). Since we are counting focus points, i. e. distinct root verbs of the construction, inconsistencies can arise whenever there are instances of G_A and G_B that share their root verbs. In Example 9, the verb *claimed* is part both of an instance of G_A (“claimed he suffered”) and an instance of G_B (“claimed that implied”) but there is no instance of G_C .

(9)_{BNC: A8D 322} [...] Donovan claimed he suffered serious injury [...] and that the article implied he was guilty [...].

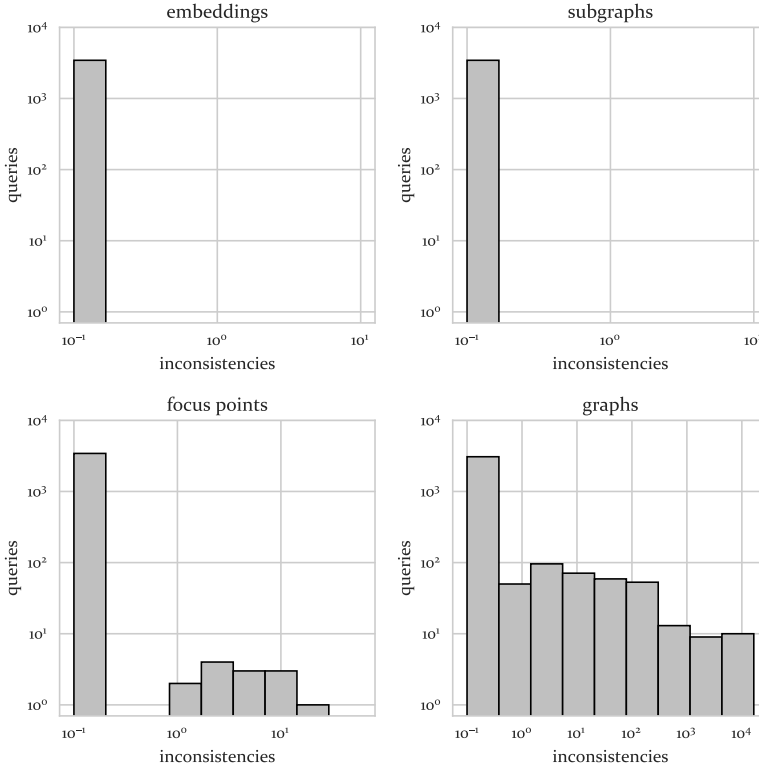


Figure 5.10: Inconsistencies for partly unlexicalized queries. Note that both axes are logarithmic.

The query with the highest number of inconsistencies when counting graphs could be referred to as the “the NOUN of the NOUN” structure and is shown in Figure 5.12. Since we are counting graphs, we are classifying all sentences that contain an instance of “NOUN of NOUN” (G_N , Figure 5.12d). A consistent classification will be impossible in all cases where a sentence contains independent instances of “NOUN of the NOUN” (G_A , Figure 5.12a) and “the NOUN of NOUN” (G_B , Figure 5.12b), which explains the high number of inconsistencies for this query. When counting focus points, this particular query is entirely unproblematic since there are no cooccurrences of G_A and G_B in the corpus that have the root vertex of this query in common and do not constitute an instance of G_C .

When looking at the inflation scores for the partly unlexicalized queries in Figure 5.13a, we see that there is one query with an extremely high degree of

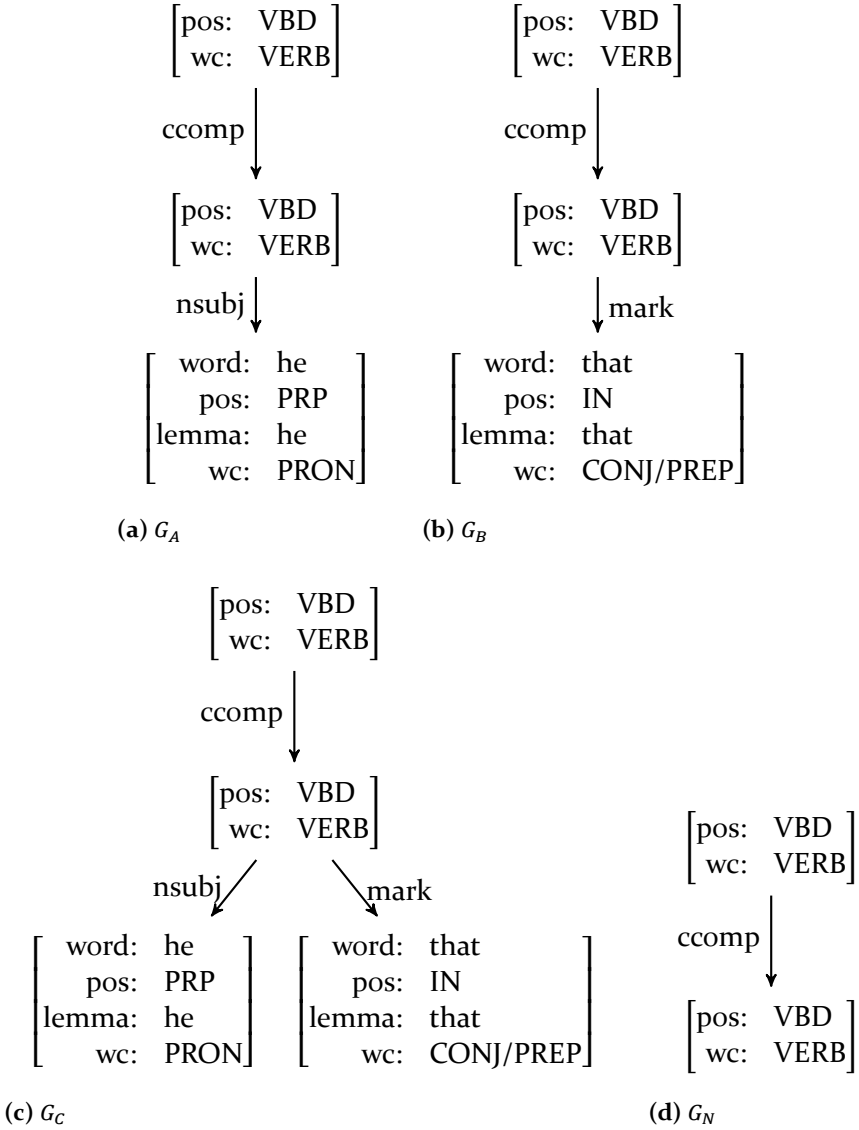


Figure 5.11: Query graphs for the “VERBed that he VERBed” structure yielding inconsistent results

inflation. This query looks for a noun with five outgoing nn relations. Four of these relations have the requirement that their dependent is a noun, the dependent of the fifth relation has to be the noun *urokinase*. This structure looks almost like the one used in Section 4.4.1 to illustrate the problem of

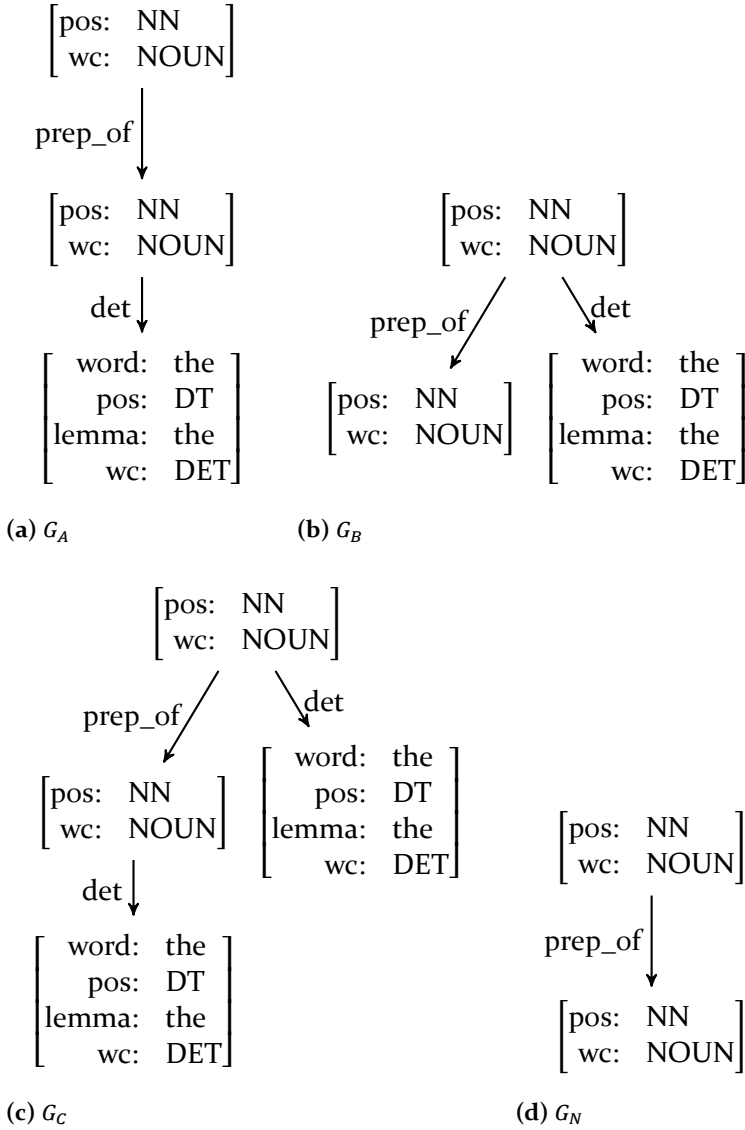


Figure 5.12: Query graphs for the “the NOUN of the NOUN” structure yielding inconsistent results

inflated frequencies and it should be clear that the method of counting embeddings will be highly affected by the effects of combination and permutation. The reason why this weird structure involving an obscure noun is a query in the first place is that there are seven instances of the phrase “urokinase

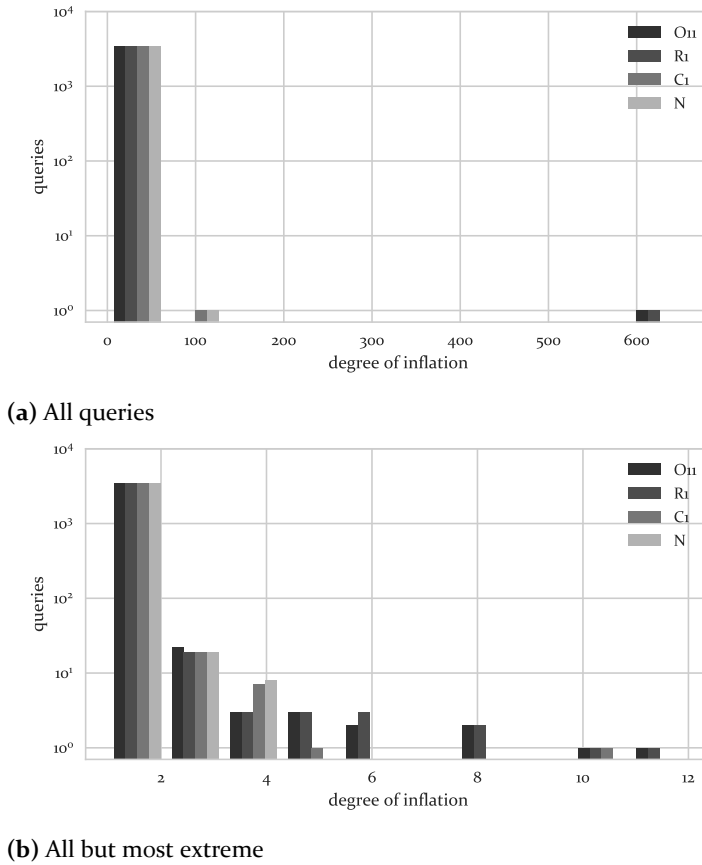


Figure 5.13: Degree of inflation for partly unlexicalized queries (number of embeddings per focus point). Note that the y -axis is logarithmic.

type plasminogen activator/tissue type plasminogen activator antigen ratio” in the BNC.

If we omit this query from the plot (cf. Figure 5.13b), we can see degrees of inflation quite similar to the completely lexicalized queries. The second most extreme query, i. e. the one with the highest degree of inflation in this plot, is the partly delexicalized version of the *owl* query mentioned in Section 5.2.3.

Figure 5.14 shows that, when using the Dice coefficient, all counting methods produce virtually identical rankings of the partly unlexicalized queries with Spearman correlation coefficients ≈ 1.00 in all cases. The point that is furthest away from diagonal corresponds to the *urokinase* query mentioned above.

5 Evaluation of the methodology

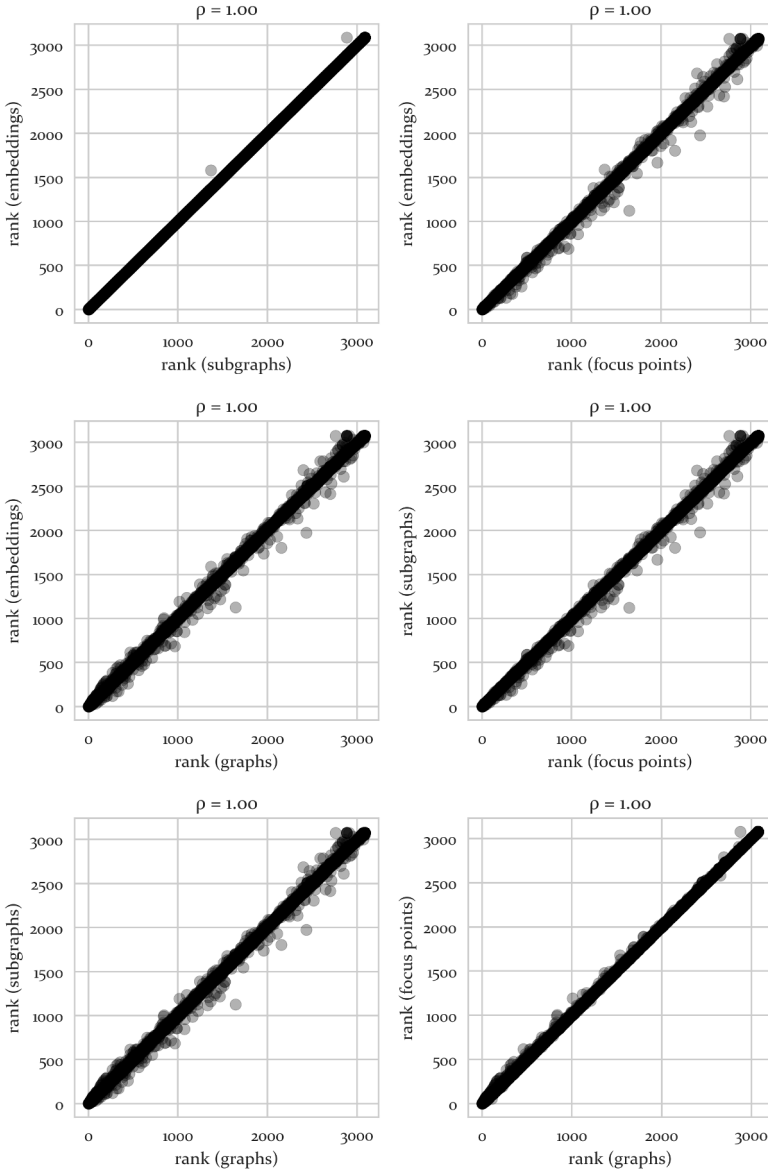


Figure 5.14: Scatterplots and Spearman correlation coefficients for different counting methods based on the Dice coefficient (partly unlexicalized queries)

The corresponding plots for t -score and log-likelihood (Figures 5.15 and 5.16) show more instances further away from the diagonal, indicating greater

5.2 Experiments with frequent subgraphs

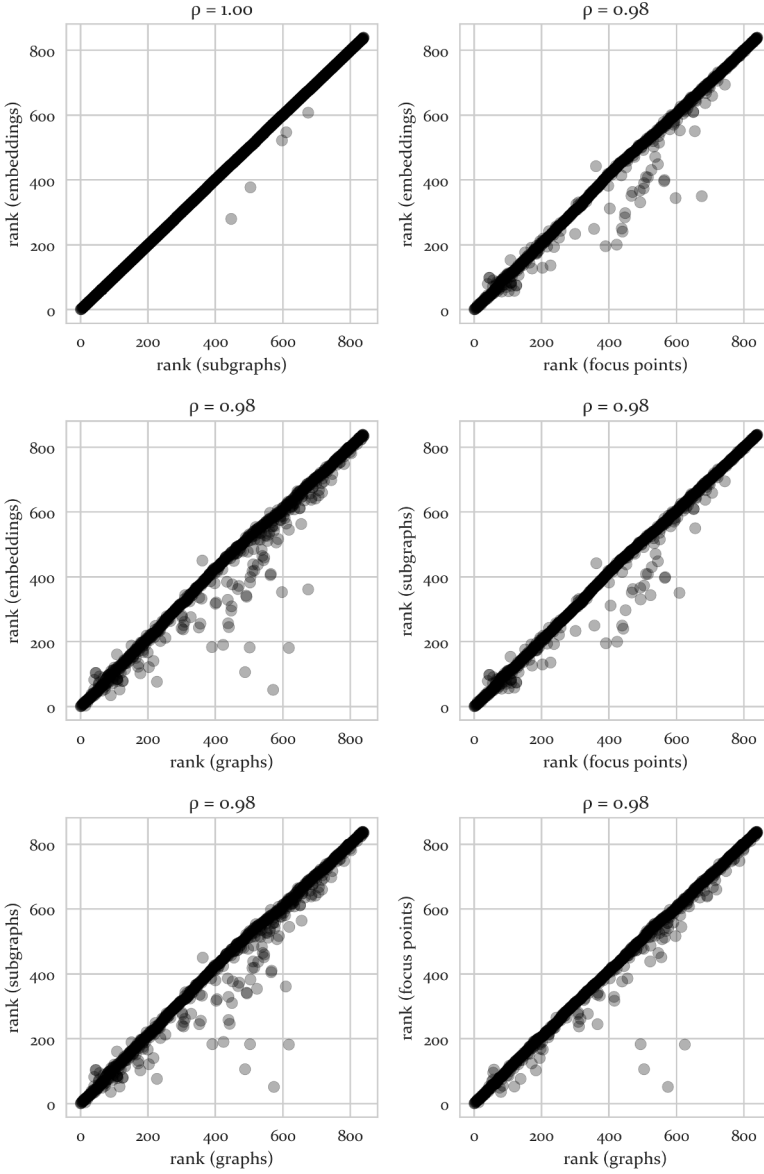


Figure 5.15: Scatterplots and Spearman correlation coefficients for different counting methods based on t -score (partly unlexicalized queries)

ranking differences. Nevertheless, the overall rankings are still nearly identical for all counting methods (Spearman correlation coefficients ≥ 0.98).⁵

⁵ When looking at the graphs, a couple of dozen data points off the diagonal might seem a lot at first, but we have to keep in mind that there are 3 440 data points.

5 Evaluation of the methodology

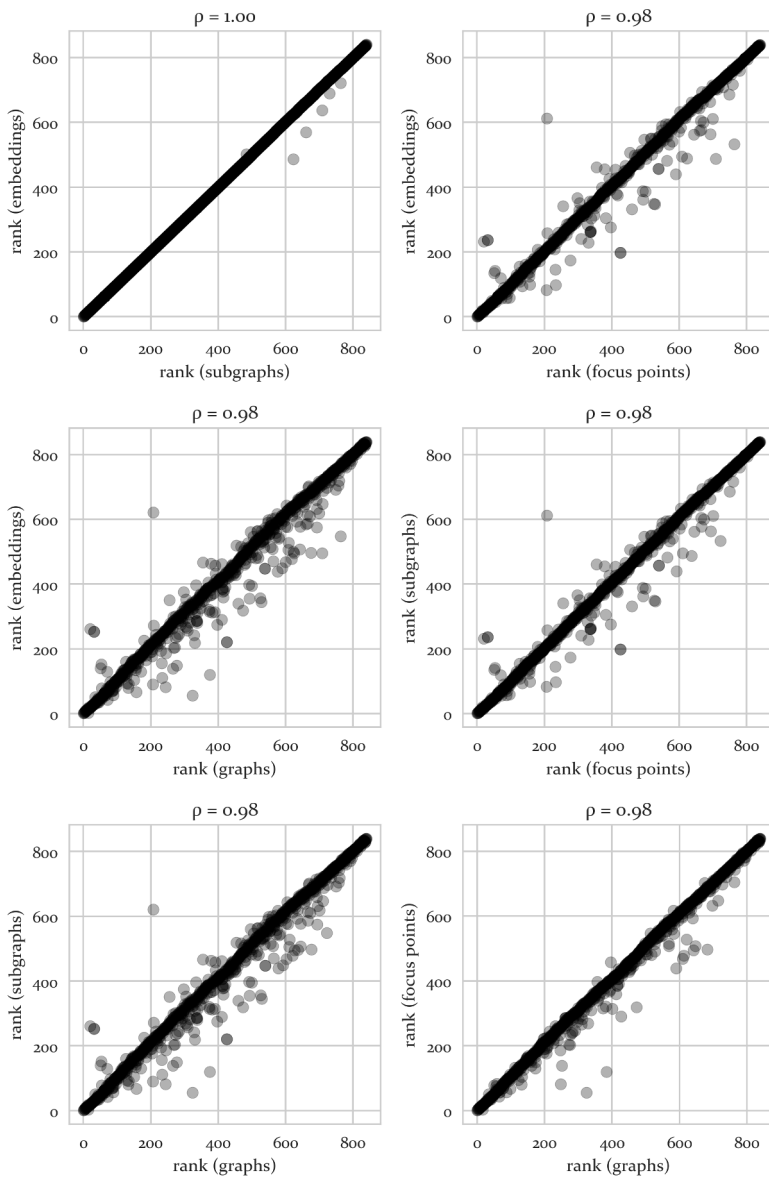


Figure 5.16: Scatterplots and Spearman correlation coefficients for different counting methods based on log-likelihood (partly unlexicalized queries)

The most notable differences can be observed when one of the counting methods involved is counting graphs. However, the queries with the greatest ranking differences look fairly unsuspicious. So how do the huge ranking

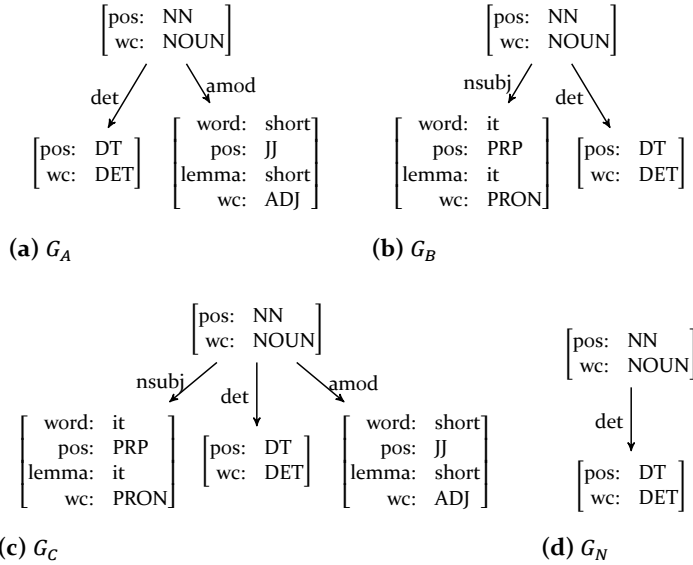


Figure 5.17: Query graphs for the “it (BE) DETERMINER short NOUN” structure

differences come about? Let us illustrate this with the most affected query (cf. Figure 5.17). The structure we are looking for could be called the “it (BE) DETERMINER short NOUN” structure (G_C) and matches things like “it was a short letter” or “it is a short step”. We are interested in the association between “it (BE) DETERMINER NOUN” (G_A) and “DETERMINER short NOUN” (G_B). In order to determine the association strength, we classify all instances of “DETERMINER NOUN” (G_N). The query looks unsuspecting because neither the whole structure nor its parts are susceptible to the problems of combination and permutation. In most cases, there is only a single instance of G_A , G_B or G_C in a sentence, regardless of the counting method involved. Therefore, O_{11} , R_1 and C_1 are roughly the same for all counting methods. The crucial point is the frequency of G_N . It is fairly common that there is more than one instance of “DETERMINER NOUN” per sentence. When we are counting embeddings, subgraphs or focus points, each of these instances constitutes a unit of analysis and is counted towards the sample size N . However, when we are counting graphs, we only classify sentences, i. e. each sentence only counts once towards the sample size, even if it contains several instances of “DETERMINER NOUN”. As a result, O_{11} , R_1 and C_1 are roughly the same for all counting methods, whereas the sample size N differs by a factor of

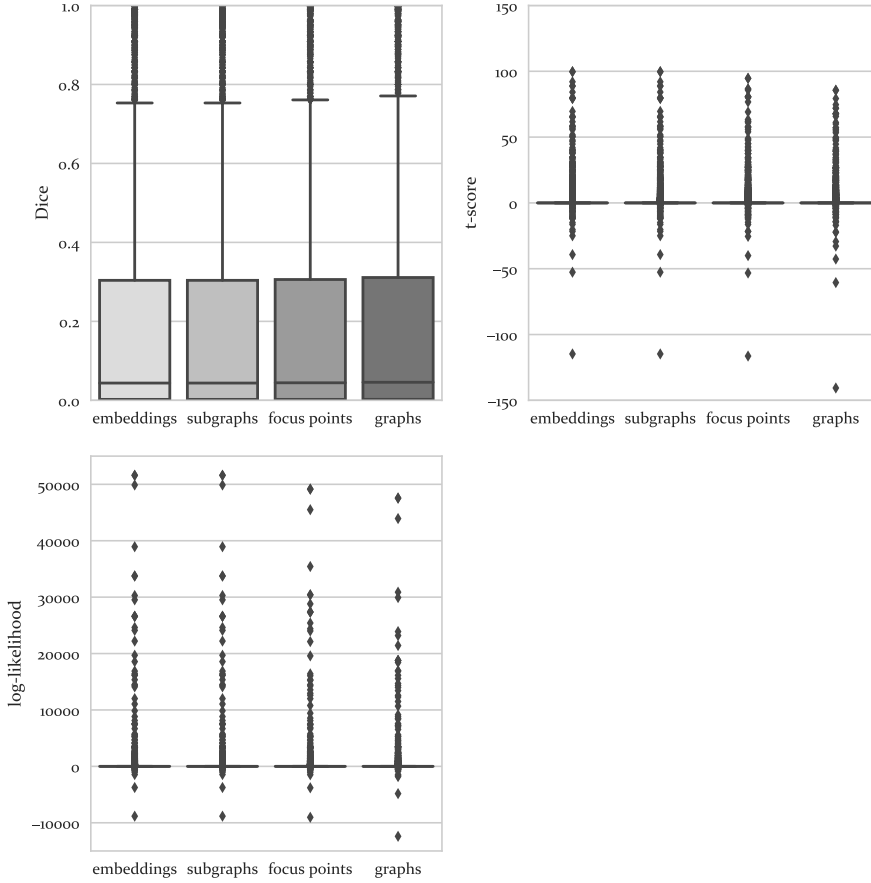


Figure 5.18: Distribution of association scores for different counting methods (partly unlexicalized queries)

two between counting graphs and the other counting methods – a perfect instance of the scenario discussed in the thought experiment in Section 5.1.2.

When counting focus points, the partly delexicalized version of the *owl* query is furthest from the diagonal. The reason for this is that the degree of inflation for O_{11} , R_1 and C_1 is three times that for N when compared to counting embeddings or subgraphs. The resulting differences in the contingency tables lead to the ranking differences.

Figure 5.18 shows that, on the whole, the problem of inflated frequencies does not play a role when analyzing partly unlexicalized queries. While both *t*-score and log-likelihood have more extreme values than in the previous

experiment, this is due to the fact that G_N is now completely unlexicalized, i. e. that we have much larger sample sizes. As shown in Section 5.1, such an increase affects both t -score and log-likelihood, with the former being less affected than the latter. Within this experiment, the distribution of association scores is virtually identical for all four counting methods and there is no evidence that the few cases of inflated frequencies lead to distortions.

5.2.5 Evaluation based on mostly unlexicalized subgraphs

For the third experiment, we further increase the number of underspecified elements in the queries. Instead of delexicalizing merely G_N , i. e. the overlapping part of G_A and G_B , we now additionally delexicalize all of G_A or G_B , whichever has less vertices. As a result, G_N and either G_A or G_B are now completely unlexicalized, i. e. all of their vertices have only part-of-speech tag and word class labels, and the remaining part, G_B or G_A , is partly unlexicalized. In our example structure, “could not be more different”, G_A has four vertices and G_B has three. This means that we remove the word and lemma labels from the vertices *could*, *be* and *different*, leading to the structure “MODAL not VERB more ADJECTIVE”. The resulting query graphs are shown in Figure 5.19.

Apart from these changes in the queries, the experiment is identical to those in the previous sections: We determine the association strength for all cooccurrences based on all four counting methods and the three association measures and rank the queries according to the association strength for every combination of counting method and association measure.

Results

The results are similar to those obtained from the partly unlexicalized queries in the previous section. The most extreme outliers are queries similar to those in the previous section or even (more strongly delexicalized versions of) the same queries.

The cooccurring linguistic structures are now even less specific than in the previous experiment, resulting in a greater number of inconsistent cases (cf. Figure 5.20). When counting embeddings or subgraphs, there are 6 queries with inconsistencies, when counting focus points, there are 27 and when counting graphs, there are 453 queries with inconsistencies.

When counting graphs, the query with the highest number of sentences that could not be consistently classified is looking for instances of the structure “NOUN of the ADJECTIVE NOUN” (G_C). We are interested in the association between “NOUN of ADJECTIVE NOUN” (G_A) and “the ADJECTIVE NOUN” (G_B).

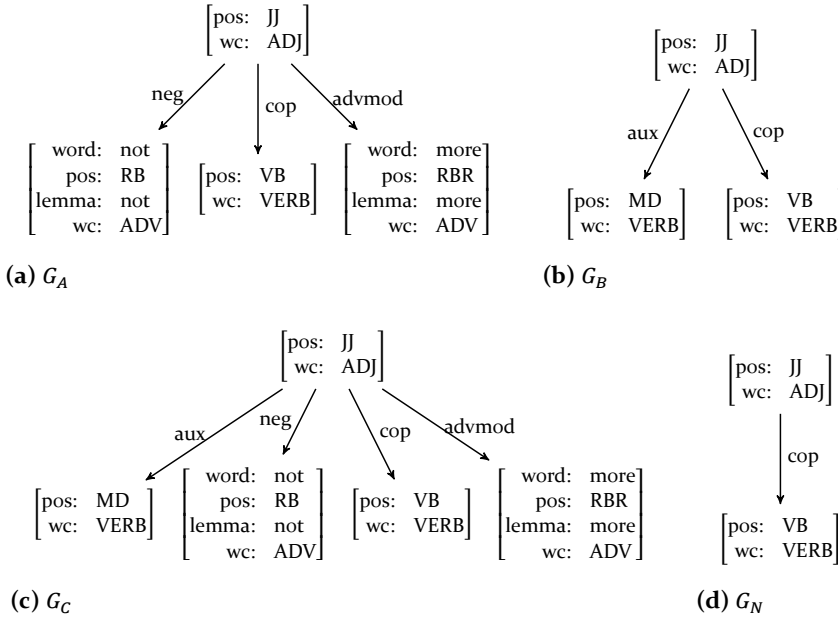


Figure 5.19: Query graphs for the structure “MODAL not VERB more ADJECTIVE” as a cooccurrence of “not VERB more ADJECTIVE” and “MODAL VERB ADJECTIVE”.

To measure the association strength, we need to classify all sentences that contain an instance of “ADJECTIVE NOUN” (G_N). Since both G_A and G_B readily cooccur in the same sentence independently of each other, this counting method leads to a high number of inconsistencies.

When counting focus points, the greatest number of inconsistencies arises for a query that looks for the structure “there VERB many ADJECTIVE NOUNS” (G_C). We want to measure the association strength between “there VERB ADJECTIVE NOUNS” (G_A) and “there VERB many NOUNS” (G_B). To do that, we classify all verbs that occur in the structure “there VERB NOUNS” (G_N). Example 10 contains an instance of G_N that, according to the method of counting focus points, can be classified as both an instance of G_A and G_B but not as an instance of G_C :

- (10)_{BNC: FT₂ 1867} Suitable explanations for any delay would be that there are too many patients or too few doctors and that the service is underfunded.

The focus point that matches in both cases is the verb *are*. Due to the way coordination is analyzed in the Stanford Dependencies scheme, it is part of both G_A (“there are many patients” and “there are few doctors”) and G_B

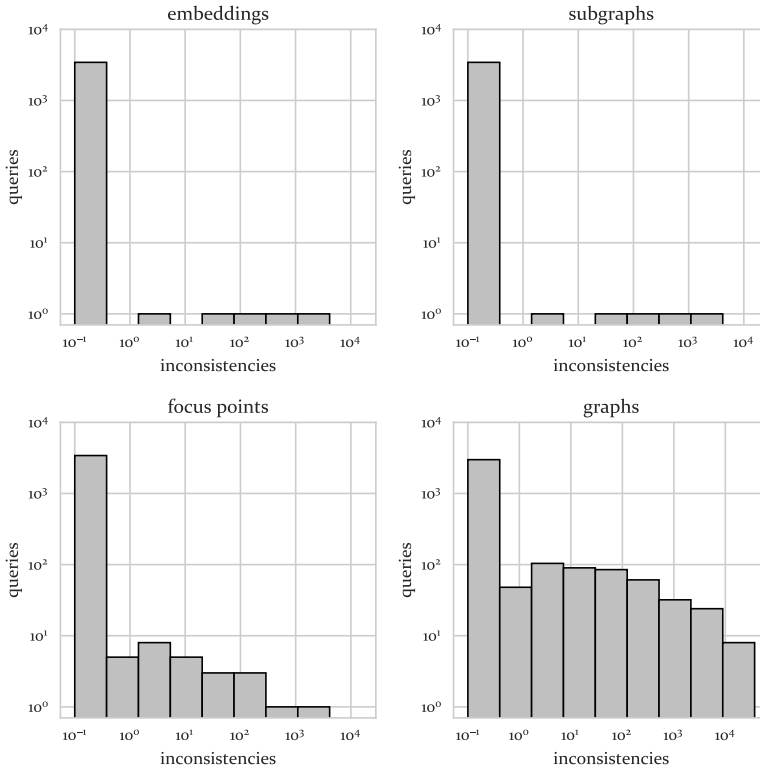


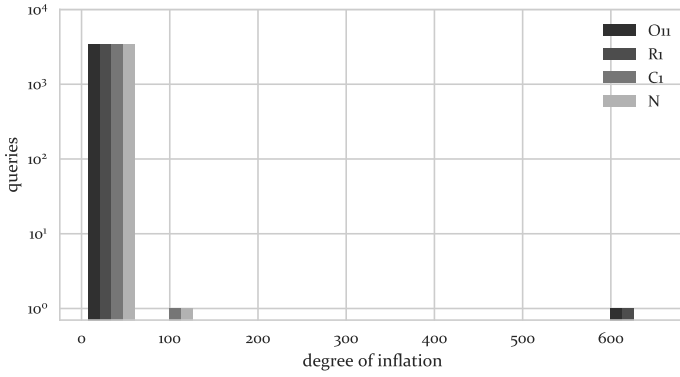
Figure 5.20: Inconsistencies for mostly unlexicalized queries. Note that both axes are logarithmic.

(there are many patients) but not of an instance of G_C (that would require an additional adjectival modifier for *patients*).

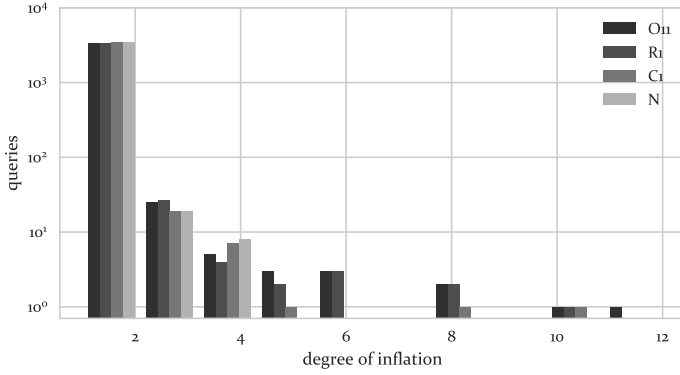
Similar to what we saw for the partly unlexicalized queries in the previous section, the plot in Figure 5.21a that shows all the inflation scores for the mostly unlexicalized queries is not very informative because of a single query with an extremely high degree of inflation. This query happens to be the mostly unlexicalized version of the *urokinase* query discussed above that was also the most extreme outlier in the previous experiment.

The query with the second highest degree of inflation that stands out most in Figure 5.21b (where the *urokinase* query has been removed) is another old acquaintance, the mostly unlexicalized version of the *owl* query discussed in Section 5.2.3.

5 Evaluation of the methodology



(a) All queries



(b) All but most extreme

Figure 5.21: Degree of inflation for mostly unlexicalized queries (number of embeddings per focus point). Note that the y-axis is logarithmic.

Figure 5.22 shows that, when using the Dice coefficient, all counting methods produce virtually identical rankings even for the mostly unlexicalized queries (Spearman correlation coefficients ≈ 1.00 in all cases). The two queries with the highest ranking differences are the *urokinase* query and the *owl* query.

When using the *t*-score or log-likelihood measures, there are more queries with greater ranking differences between the four counting methods (cf. Figures 5.23 and 5.24). This is particularly pronounced when the graph counting method is involved where we see Spearman correlation coefficients of 0.94–0.95 (in the other cases, we have correlation coefficients ≥ 0.98). This is the same phenomenon we could observe in the previous experiment (cf. Section 5.2.4) and the same explanation holds. The queries with the

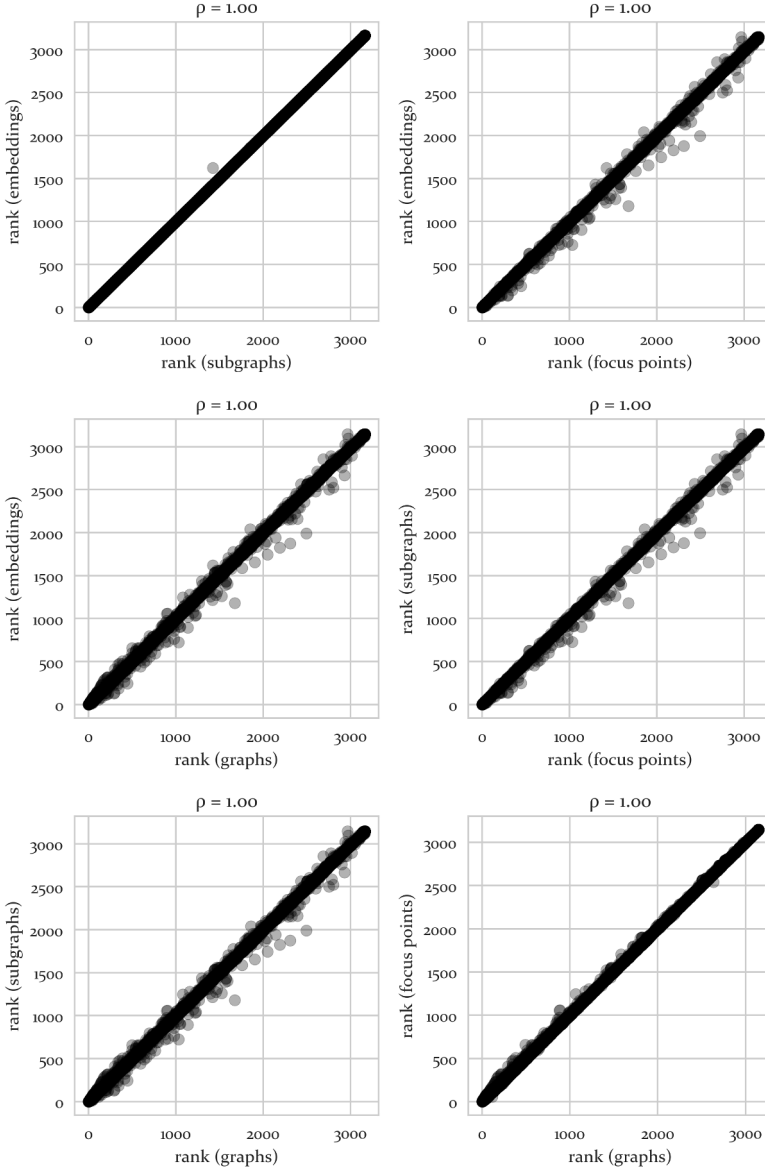


Figure 5.22: Scatterplots and Spearman correlation coefficients for different counting methods based on the Dice coefficient (mostly unlexicalized queries)

greatest ranking differences look quite harmless, searching for structures like “DETERMINER new NOUN NOUN” (G_C). These cooccurrences and their parts, e. g. “DETERMINER NOUN NOUN” (G_A) and “DETERMINER new NOUN”

5 Evaluation of the methodology

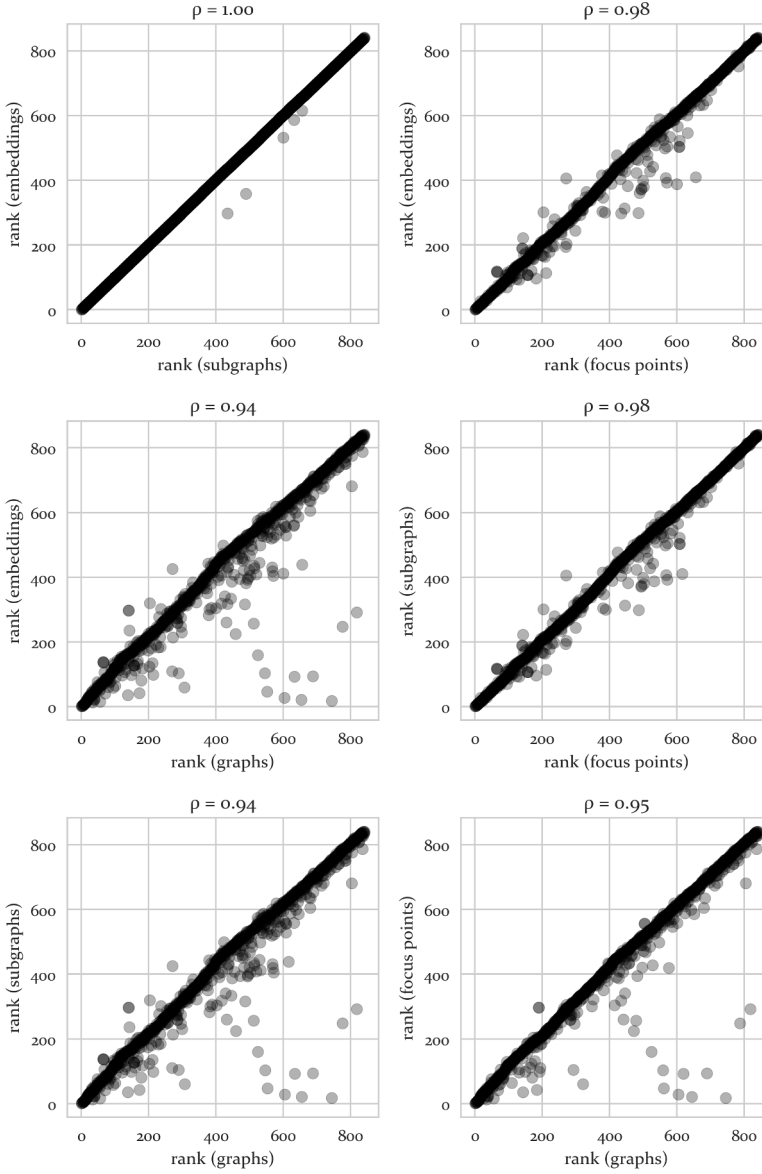


Figure 5.23: Scatterplots and Spearman correlation coefficients for different counting methods based on t -score (mostly unlexicalized queries)

(G_B), get roughly the same frequency counts with all four counting methods. However, the frequency of G_N , e. g. “DETERMINER NOUN”, is quite different

5.2 Experiments with frequent subgraphs

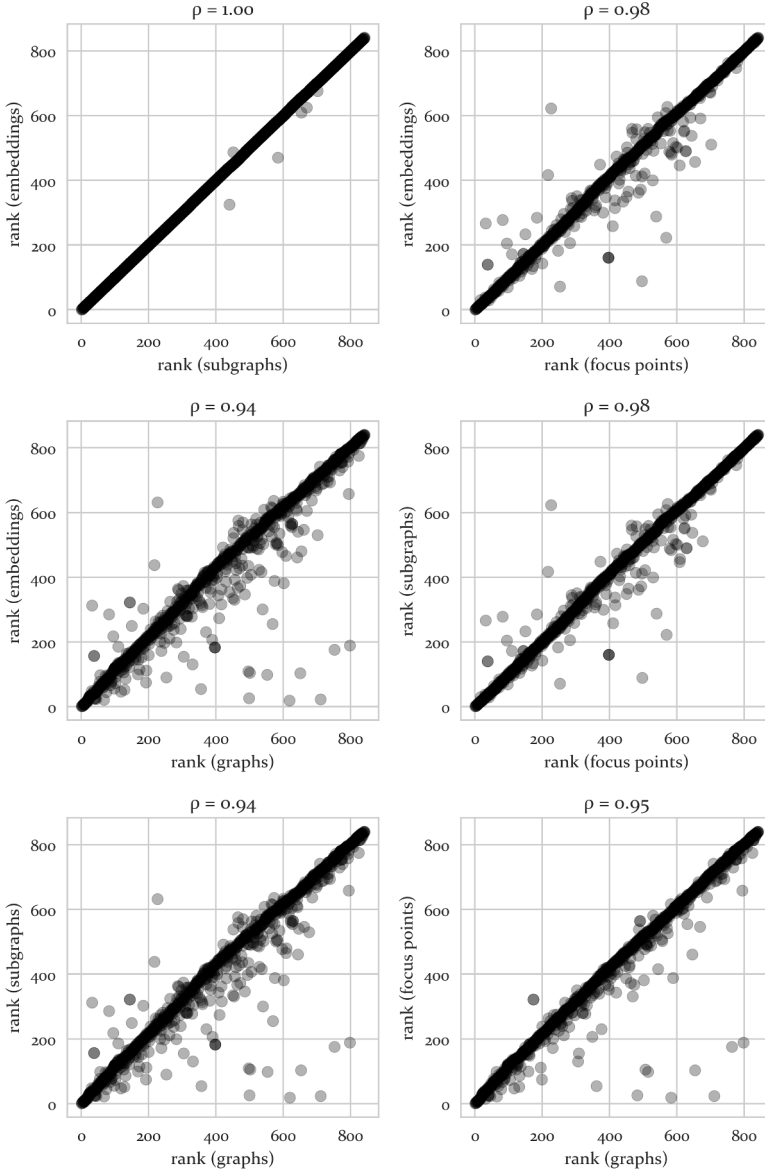


Figure 5.24: Scatterplots and Spearman correlation coefficients for different counting methods based on log-likelihood (mostly unlexicalized queries)

when counting graphs than when using the other counting methods. As a result, t -score and log-likelihood react as described in Section 5.1.2.

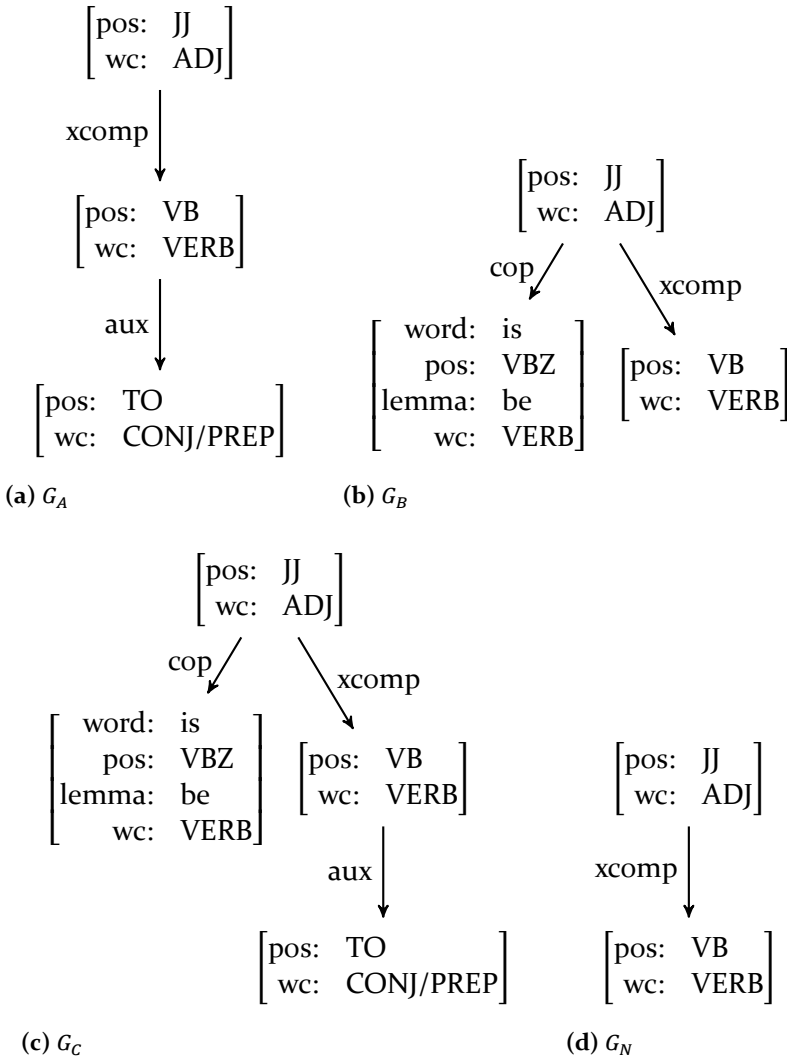


Figure 5.25: Query graphs for the “is ADJECTIVE to VERB” structure

When we count focus points, we see the greatest ranking differences for the *owl* query and the query depicted in Figure 5.25 that looks for the structure “is ADJECTIVE to VERB”.

While there are more queries with inconsistencies among the mostly unlexicalized queries, degrees of inflation are similar to partly unlexicalized queries and completely lexicalized queries. This is reflected in the overall

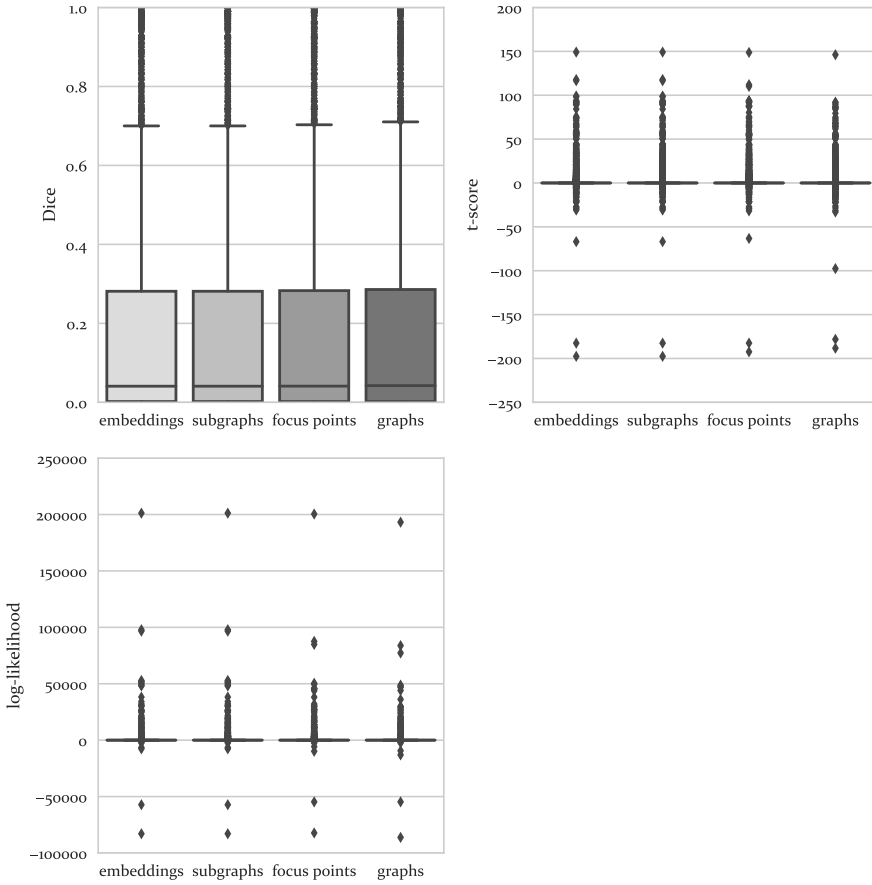


Figure 5.26: Distribution of association scores for different counting methods (mostly unlexicalized queries)

distribution of association scores visualized in Figure 5.26. The distribution of association scores is virtually identical for all four counting methods.

5.3 Summary

By formally analyzing the behavior of association measures in two theoretical scenarios for inflated frequencies, we showed that there are three groups of association measures in both scenarios. The groups exhibit different behaviors and can be represented by the Dice coefficient and the t -score and log-likelihood measures. We used those three measures in a three-part

evaluation experiment where we created queries based on frequent subgraphs extracted from the BNC. In the first experiment, the queries were completely lexicalized, for the second and third experiment, greater and greater parts of the queries were delexicalized, using only part-of-speech and word class information instead. We could observe an increase in the number of inconsistencies as the number of underspecified elements the queries increased, especially for the method of counting graphs. Inflated frequencies turned out not to be a major problem. While there were a number of problematic queries, the overall distribution of association scores was not affected and also the overall rankings produced by the different counting methods were virtually identical, with Spearman correlation coefficients in the high nineties. The insights from the theoretical investigation were confirmed by the empirical results: If the Dice coefficient is used, then it does not matter which counting method is chosen. The results obtained by counting embeddings and subgraphs are perfectly correlated, independently of the association measure used. The method of counting graphs leads to the most inconsistencies and is the one most different to the other counting methods.

6 Case studies

In this chapter, we present examples that illustrate some of the possibilities that the generalized cooccurrence model offers for performing queries. In these examples, we both reconstruct existing approaches and highlight some of the new possibilities. In some cases, we also explore alternative ways of putting the queries and the effect this has on the results.

Many of these case study focus on the verb *give*, mainly because *give* is a frequent verb that occurs in interesting syntactic structures. In the experiments that reconstruct relational cooccurrences, we count embeddings (cf. Section 4.6.1), in those that deal with larger structures, we count focus points.

6.1 Cooccurring words

Relational cooccurrence (Section 2.1.2), i. e. the cooccurrence of two items that are related via some relation, usually a dependency relation, has already been reconstructed in Section 4.6.1. In this section, we use relational cooccurrences to find collocation candidates for the verb *give*. First, we reconstruct two frequent scenarios: Finding collocation candidates by looking at all relations (Section 6.1.1) and finding collocation candidates by looking only at verbal relations (Section 6.1.2). After that, we highlight the flexibility of the generalized cooccurrence model by formulating the queries in such a way that we no longer classify relations but single tokens (Section 6.1.3).

The general approach, however, is always the same: First we have to determine the set of words (or lemmata, in our case) that cooccur with the verb *give*, then, for every cooccurring word, we determine the association strength between that word and the verb *give*. Finally, we rank the cooccurring words by their association strength.

In all experiments, we set a frequency threshold of $O_{11} \geq 5$.

6.1.1 Relational cooccurrences based on all relations

The relational cooccurrence between the verb *give* and another lemma, e. g. *rise* (linguistic structure G_C , Figure 6.1c), can be interpreted as the cooccurrence of linguistic structure G_A (Figure 6.1a), i. e. the verb *give* with an

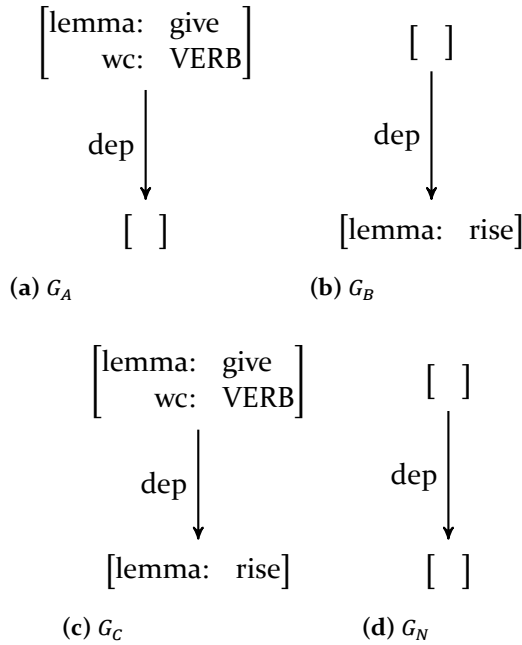


Figure 6.1: Query graphs for relational cooccurrences based on all relations

outgoing dependency relation, with linguistic structure G_B (Figure 6.1b), i. e. the lemma *rise* with an incoming dependency relation. This means we have to classify all instances of linguistic structure G_N (Figure 6.1d), i. e. all dependency relations in the corpus. Of course, we can also perform the same analysis for governors of the verb *give*. We simply have to reverse the direction of the dependency relation.

We performed those queries for all lemmata that cooccur at least five times with the lemma *give* in the BNC and calculated the log-likelihood scores. Table 6.1 shows the 20 most strongly associated dependents and governors obtained this way.

Table 6.1 contains a lot of lexicographically interesting information about the usage of the verb *give*. We can find *give rise*, *give up*, *give away* and *give birth*, we can see that *give* is frequently used in direct and indirect speech (*say*, *ask*, *reply*), and that it is used to modify nouns such as *reason*, *advice* and *information*.

However, from a lexicographer's perspective, there are also some uninteresting items in the lists, e. g. personal pronouns (*I*, *you*, *he*, *we*, *they*) and

#	dependent	O_{11}	G	#	governor	O_{11}	G
1	to	18210	11671.39	1	say	2372	2104.83
2	rise	1956	11303.66	2	reason	439	1249.48
3	chance	1711	10643.47	3	advice	214	1086.15
4	I	12482	10191.78	4	surprising	180	868.84
5	up	4325	8391.94	5	refuse	260	854.11
6	impression	1008	8272.58	6	ask	640	829.48
7	you	9573	7266.43	7	'	349	693.76
8	opportunity	1312	7086.41	8	information	268	641.38
9	advice	934	5309.58	9	reply	201	640.42
10	indication	586	4739.30	10	able	460	619.26
11	detail	1021	4411.97	11	give	1032	607.55
12	he	11445	4382.37	12	answer	266	571.30
13	away	1338	4209.36	13	want	761	545.99
14	birth	653	4090.47	14	prepare	256	518.70
15	we	5821	3988.52	15	decide	312	505.84
16	will	3642	3670.77	16	the	251	495.47
17	they	8255	3641.85	17	help	435	461.40
18	notice	742	3566.80	18	require	328	460.35
19	information	1213	3529.72	19	have	2361	438.54
20	evidence	937	3453.78	20	impression	110	399.47

Table 6.1: Dependents and governors of the verb *give*

other function words like *to* and *will*. The list of governors also contains items that are due to parsing errors, e. g. *the* and the quotation mark or apostrophe.

All in all, 11 out of the 20 highest ranking dependents are collocations listed in the Oxford Collocations Dictionary (OCD; McIntosh et al., 2009), corresponding to a precision at 20 (P@20) of 0.55. Only 7 of the governors from Table 6.1 are listed in the OCD, corresponding to a P@20 of 0.35.

6.1.2 Relational cooccurrences based on verbal relations

Studies on collocations that focus on a particular type of collocation, e. g. adjective-noun pairs or verb-noun pairs, usually do not base their analysis on all dependency relations in the corpus but only on the relevant subset, e. g. either all relations between adjectives and nouns or between verbs and nouns or only specific dependency relations (cf. also Section 6.2). We can perform our search for collocation candidates of *give* with a similar restriction. Since *give* is a verb, we are only interested in verb-based relations, i. e. in relations governed by a verb when we are looking for dependents of *give* and

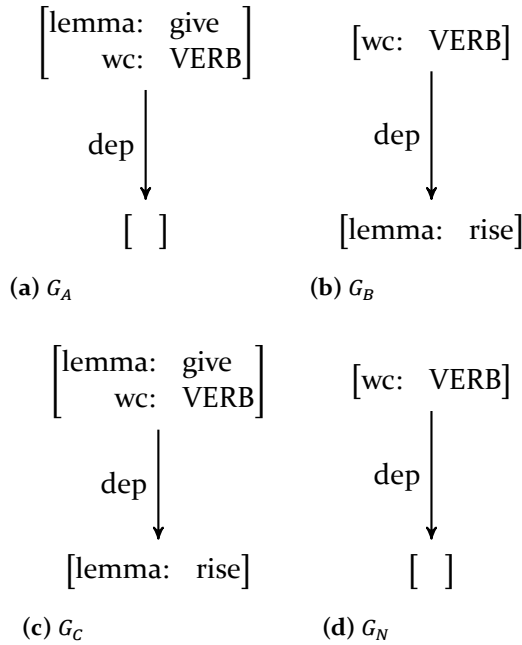


Figure 6.2: Query graphs for relational cooccurrences based on verbal relations

in relations with a verb as dependent when we are looking for governors of *give*. Figure 6.2 shows the corresponding query graphs, again for the collocation candidate *rise*. Linguistic structures G_A and G_C are the same as before, the only difference is that both G_B and G_N now have the restriction that the governor must be a verb.

The effects these changes in the queries have on the contingency tables are fairly obvious: While O_{11} and R_1 stay the same, C_1 can be expected to be smaller (unless the candidate is only ever governed by verbs, in which case it also stays the same) and N is definitely smaller. As can be expected, the differences in the frequency signatures lead to differences in the association measure and thereby to a different ranking of the collocation candidates. The lists of the 20 most strongly associated dependents and governors based on verbal relations (Table 6.2) do not only look different to the ones based on all relations (Table 6.1), they also look cleaner.

To assess the similarity between two lists A and B , we can use the Dice coefficient that has an intuitive interpretation since it corresponds to the probability with which an item from list A occurs in list B (or vice versa):

#	dependent	O_{11}	G	#	governor	O_{11}	G
1	rise	1956	9930.70	1	advice	214	1398.24
2	chance	1711	7979.84	2	reason	439	967.37
3	impression	1008	6560.82	3	information	268	766.82
4	opportunity	1312	5420.64	4	answer	266	584.43
5	birth	653	4568.20	5	reply	201	518.36
6	advice	934	4435.88	6	surprising	180	499.46
7	indication	586	3845.29	7	give	1032	490.05
8	detail	1021	3305.71	8	priority	95	430.27
9	information	1213	3140.80	9	impression	110	425.84
10	priority	555	2953.37	10	lecture	82	415.32
11	notice	742	2692.51	11	refuse	260	366.31
12	up	4325	2638.99	12	attention	75	364.40
13	consideration	600	2535.13	13	name	208	341.67
14	name	1116	2448.98	14	instruction	81	330.84
15	evidence	937	2343.61	15	say	2372	321.59
16	away	1338	2308.94	16	evidence	179	306.02
17	assurance	306	1991.58	17	example	165	290.01
18	support	977	1832.74	18	be	3351	255.81
19	permission	374	1692.12	19	ask	640	249.32
20	access	520	1637.09	20	publicity	44	242.06

Table 6.2: Dependents and governors of the verb *give*

$$\text{Dice} = \frac{2 \cdot |A \cap B|}{|A| + |B|}$$

The Dice coefficients for the two lists in Tables 6.1 and 6.2 are 0.65 for the dependents and 0.55 for the governors. The Spearman correlation coefficients for the complete rankings of all candidates are 0.83 for dependents and 0.61 for governors. This means that the lists do not only appear to be different, the difference is also measurable. Similarly, we can also quantify the quality of the 20 candidates. 17 of the dependents are listed in the OCD as collocations (P@20: 0.85) and 14 of the governors (P@20: 0.7). We can conclude that formulating more specific queries, in this case by a restriction to verbal relations, can lead to a different and markedly better ranking of the collocation candidates of *give*.

6.1.3 Alternative ways of formulating the queries

The conventional way of analyzing cooccurrences of pairs of words that are related via some relation is the relational cooccurrences approach where we classify the relations according to two criteria: “Does the first word occur in the first slot of the relation?” and “Does the second word occur in the second slot of the relation?”. However, this is not the only way in which we can tackle the problem.

In our example of looking for collocation candidates of the verb *give*, we can also choose to classify not all verb-based dependency relations but all verbs. The two criteria according to which we have to classify the verbs are: “Does the verb have the lemma *give*?” and “Does the verb have a dependency relation to the collocation candidate?”. This is illustrated in Figure 6.3 for the cooccurrence of *give* and *rise*.

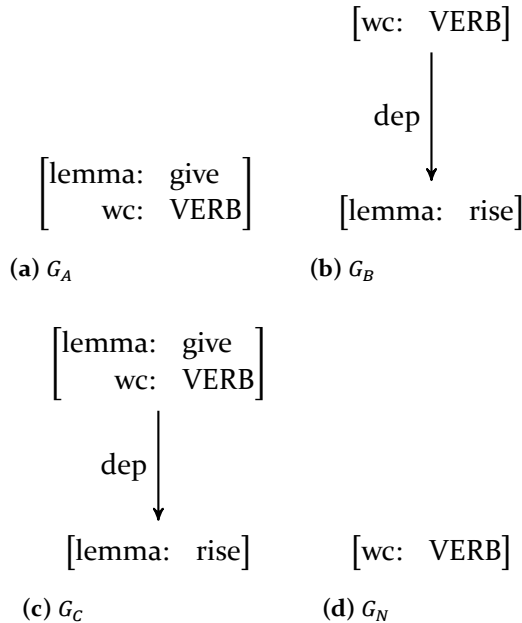


Figure 6.3: Query graphs for node-based cooccurrences

Note that by shifting the focus to *give*, the node of the collocation in Sinclairian terms, we are asking a different question. We are now interpreting relational cooccurrence as collocation (simple collexeme analysis). Instead of analyzing the association between *give* and *raise* within the simple linguistic structure “single dependency relation”, we are now analyzing the

#	dependent	O_{11}	G	#	governor	O_{11}	G
1	rise	1952	12742.67	1	advice	214	1341.46
2	chance	1708	10379.61	2	reason	439	872.62
3	be	21535	8104.74	3	information	268	705.06
4	he	11292	8024.50	4	answer	266	527.25
5	impression	1005	7997.70	5	reply	201	473.24
6	you	9473	7564.93	6	surprising	180	458.33
7	opportunity	1312	7249.36	7	priority	95	406.30
8	I	12319	7228.81	8	impression	110	398.84
9	to	18072	6851.21	9	lecture	82	394.27
10	up	4322	6755.97	10	give	1029	358.43
11	they	8085	6556.49	11	attention	75	345.27
12	advice	933	5750.77	12	refuse	260	317.50
13	birth	653	5531.93	13	instruction	81	310.77
14	indication	586	4699.32	14	name	208	300.74
15	information	1209	4696.42	15	evidence	179	270.33
16	detail	1019	4664.17	16	example	165	256.79
17	we	5772	4132.09	17	publicity	44	230.65
18	away	1338	3909.86	18	amount	139	201.83
19	name	1106	3808.01	19	undertaking	44	194.50
20	priority	554	3736.02	20	assistance	40	178.12

Table 6.3: Dependents and governors for node-based cooccurrences

association between *give* and the linguistic structure “arbitrary dependency relation with dependent *rise*”.

That also influences the contingency tables. While O_{11} and C_1 remain the same, we get very different values for R_1 (frequency of the verb *give* instead of number of dependency relations governed by the verb *give*) and N (number of verbs instead of number of verb-based dependency relations). Accordingly, there are some notable differences between the results obtained this way (Table 6.3) and those from a conventional analysis (Table 6.2), especially for dependents.

The Dice coefficients for the lists in Tables 6.2 and 6.3 are 0.65 for the dependents and 0.85 for the governors. The Spearman correlation coefficients for the complete rankings of all candidates are 0.88 for dependents and 0.99 for governors. This means that the change in methodology has little impact on the ranking of the governors. That is due to the fact that a word can have many dependents but only very rarely more than one governor, i. e. the counting difference is much more pronounced for dependents than for

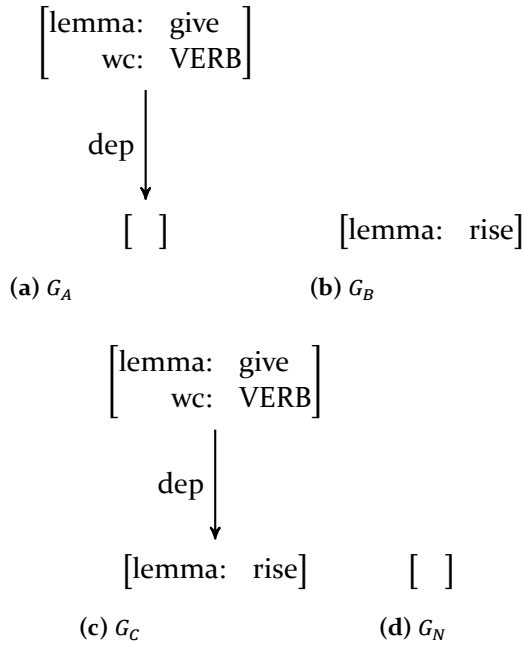


Figure 6.4: Query graphs for collocate-based cooccurrences

governors (NB: for relational cooccurrences, we are counting embeddings, i. e. dependency relations).

Precision against the OCD got notably worse for dependents (11 candidates in the dictionary, $P@20$: 0.55) and slightly better for governors (16 candidates in the dictionary, $P@20$: 0.8).

Instead of shifting the focus to the node of the collocation, we can also shift it to the Sinclairian collocate. Using our example of *give* and *rise*, this means we have to classify all tokens in the corpus according to the following two criteria: “Does the token have the lemma *rise*?” and “Is the token governed by the verb *give*?”. The corresponding queries are illustrated in Figure 6.4.

Again, this has an influence on the contingency tables. While O_{11} and R_1 are the same as when conventionally analyzing relational cooccurrences, both C_1 and N change considerably. Now, C_1 is the frequency of the collocate instead of the number of dependency relations connecting a verb to the collocate and N is the number of tokens in the corpus instead of the number of dependency relations.

The results of this kind of analysis are shown in Table 6.4.

#	dependent	O_{11}	G	#	governor	O_{11}	G
1	to	18208	12790.99	1	say	2360	7117.84
2	I	12475	11650.23	2	give	1004	3131.36
3	chance	1711	10668.06	3	want	760	2557.32
4	rise	1956	10327.90	4	ask	636	2538.13
5	up	4325	8722.05	5	able	456	2236.22
6	impression	1008	8329.08	6	reason	438	2120.97
7	you	9569	8085.21	7	go	1024	1875.58
8	opportunity	1312	7301.56	8	think	806	1841.99
9	he	11443	5577.41	9	refuse	260	1603.12
10	advice	934	5568.19	10	take	813	1560.93
11	indication	586	4650.00	11	need	577	1498.30
12	we	5820	4586.98	12	surprising	178	1466.29
13	detail	1021	4430.23	13	help	433	1426.95
14	they	8254	4397.47	14	decide	311	1377.19
15	away	1338	4317.38	15	try	428	1335.93
16	birth	653	4180.78	16	prepare	255	1318.65
17	will	3642	3881.94	17	require	325	1290.83
18	information	1213	3709.95	18	advice	214	1267.55
19	evidence	937	3518.04	19	tell	470	1234.34
20	priority	555	3417.83	20	mean	440	1151.91

Table 6.4: Dependents and governors for collocate-based cooccurrences

The Dice coefficients for the lists in Tables 6.2 and 6.4 are 0.65 for the dependents and 0.35 for the governors. The Spearman correlation coefficients for the complete rankings of all candidates are 0.84 for dependents and 0.27 for governors. Evaluation results against the OCD are also relatively poor: 11 dependents and 11 governors are listed in the dictionary as collocations, resulting in a $P@20$ of 0.55 for both.

To sum up, the generalized cooccurrence model is flexible enough to accommodate for different perspectives on the same phenomenon, i. e. cooccurrence of two words that are related via a dependency relation, that lead to (slightly) different results. In the present case study, the conventional approach of treating the phenomenon as a relational cooccurrence works better than the alternative approaches of treating it as a collocation.

6.2 Words and relations

In the examples so far, we have deliberately ignored the type of the dependency relation (direct or indirect object, adjectival modifier, etc.). Of course,

we can also incorporate this information in our analysis. In the following three sections, we present three case studies, each focusing on a different aspect of the association between words and relations.

As before, we set a frequency threshold of $O_{11} \geq 5$ in all experiments.

6.2.1 Cooccurrences of words and relations

Instead of looking at the association between two words, we can also look at the association between a word and its outgoing and incoming dependency relations. In this setting, we want to know via which types of dependency relations a word, in our example the verb *give*, governs its dependents or via which dependency relations it is itself governed.

One way to operationalize that research question is to classify all verbal dependency relations according to two criteria (in this example, we analyze the cooccurrence of the verb *give* and an outgoing indirect object relation): “Is *give* the governor of the relation?” and “Is the relation labeled iobj?”. The corresponding query graphs are depicted in Figure 6.5.¹

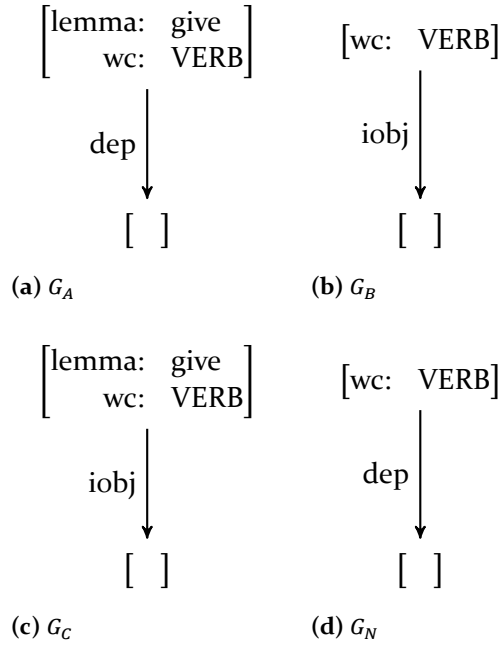
Table 6.5 lists the 20 most strongly associated outgoing and incoming dependency relations obtained this way. The most strongly associated outgoing relations are no big surprise: We would expect direct and indirect objects and prepositional phrases with *to* to be quite prominent.

The incoming dependency relations are probably a bit less straightforward, especially for a reader not familiar with the Stanford Dependencies. Since they will reappear in the next two case studies, let us illustrate some of them with example sentences (for easier reference, the governor of the relation in question will be set in italics). The xcomp relation (short for “open clausal complement”) indicates “a predicative or clausal complement without its own subject.” (de Marneffe and Manning, 2015: 10):

(11)_{BNC: K2W 225} A spokesman at the hospital *refused* to give any information as to Mr Keane’s condition.

The vmod relation (short for “reduced non-finite verbal modifier”) is used for “a participial or infinitive form of a verb heading a phrase (which may have some arguments, roughly like a VP).” (de Marneffe and Manning, 2015: 10):

¹ Alternatively, we could also apply a collocation approach and classify all verbs according to whether they are the verb *give* and whether they have an outgoing iobj relation. The main difference between the two approaches is that the former takes into account that verbs can have multiple outgoing dependency relations while the latter takes into account that verbs can also have no outgoing dependency relations at all.

**Figure 6.5:** Query graphs for cooccurrences of words and relations

#	outgoing	O_{11}	G	#	incoming	O_{11}	G
1	iobj	32036	299467.43	1	prep	4167	30121.38
2	dobj	102636	53163.30	2	xcomp	14840	7564.12
3	prep_to	18546	20770.41	3	vmod	12140	6534.42
4	pobj	5203	15983.02	4	rcmod	11937	5517.27
5	pcomp	944	3665.18	5	conj_and	11033	5178.72
6	auxpass	18120	1401.00	6	ccomp	12348	2551.21
7	vmod	3523	1370.38	7	conj_but	2175	748.40
8	nsubjpass	16805	1121.11	8	dep	6765	612.67
9	prepc_to	589	825.75	9	parataxis	2505	582.04
10	prt	6118	539.45	10	prepc_by	570	438.67
11	prepc_as_to	237	517.64	11	advcl	6578	428.60
12	csubj	807	234.23	12	prepc_without	173	146.39
13	prepc_for	349	75.74	13	conj_or	668	92.28
14	conj_x	9	36.39	14	csubj	634	91.28
15	csubjpass	138	35.84	15	prepc_for	412	80.58
16	conj_plus	25	19.25	16	conj	233	73.65
17	prepc_based_on	20	14.80	17	prepc_after	155	70.32
18	prepc_than	30	9.50	18	prepc_of	924	69.95
19	prep_together_with	58	7.81	19	prepc_into	63	69.10
20	preconj	212	7.74	20	pcomp	332	54.58

Table 6.5: Outgoing and incoming dependency relations of the verb *give*

- (12)_{BNC: GV2 2696} She saw no *reason* to give him any information he did not already have.

The rcmmod relation (short for “relative clause modifier”) should be straightforward:

- (13)_{BNC: CK0 508} That was an *answer* I could give in medical detail, and so I did.

The ccomp relation (short for “clausal complement”) is used for “a dependent clause with an internal subject which functions like an object of the verb, or adjective.” (de Marneffe and Manning, 2015: 4):

- (14)_{BNC: GXX 593} I *hope* it gives you some idea of what happened.

6.2.2 Cooccurrences of words and relation-word pairs

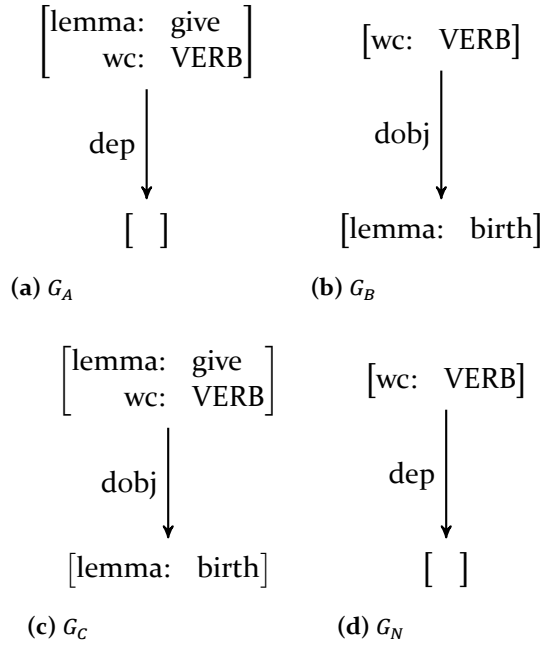
Knowing which dependency relations are strongly associated with a word can give valuable information about how the word is typically used. It would be even better, still, if we knew which words typically go with which dependency relation. One possible way of finding this out, is to determine the association strength between the word (in our example, this is still the verb *give*) and the cooccurring relation-word pairs. If we want to know how strongly *give* is associated with *birth* as direct object, we can classify all verbal dependency relations according to two criteria: “Is *give* the governor of the relation?” and “Is *birth* the dependent of the relation and is the relation labeled *dobj*?” (note how the second criterion checks for two different properties of the relation). The corresponding queries are illustrated in Figure 6.6.²

The results are shown in Table 6.6. The verb *give* is strongly associated with indirect objects that are pronouns (which is not particularly interesting from a lexicographer’s perspective) and with direct objects such as *chance*, *impression*, *birth*, *advice*, etc. *Give* is a dependent of nouns such as *time*, *reason*, *answer* and verbs such as *reply*, *say*, *prepare* and *refuse*.

6.2.3 Word sketches (cooccurrences of words by relation)

In the previous section, we established that it would be useful to know which words are typically used with which dependency relation. To that end, we measured the association between a word and the relation-word pairs it cooccurs with. A more popular alternative is to determine the association

² As in the previous case study, we could also phrase this as a simple collexeme analysis.

**Figure 6.6:** Query graphs for cooccurrences of words and word-relation pairs

#	relation	dependent	O_{11}	G	#	governor	relation	O_{11}	G
1	iobj	you	5016	44678.38	1	surprising	prep	125	1773.66
2	iobj	I	4832	42239.54	2	time	amod	219	1318.71
3	iobj	he	4159	36031.11	3	be	prep	169	1155.31
4	iobj	they	3250	29760.75	4	reason	vmod	217	1021.27
5	iobj	we	2701	24713.76	5	advice	vmod	109	998.84
6	dobj	rise	1908	17719.91	6	have	prep	108	876.27
7	iobj	it	1689	17648.50	7	answer	rcmod	113	851.87
8	iobj	she	994	9891.94	8	name	vmod	124	779.46
9	dobj	chance	1602	9112.65	9	priority	vmod	76	713.06
10	dobj	impression	975	7309.34	10	reply	ccomp	152	698.98
11	dobj	birth	629	6247.93	11	seem	prep	53	646.05
12	dobj	opportunity	1220	6209.66	12	advice	rcmod	78	632.10
13	dobj	indication	532	4999.77	13	possible	prep	48	625.20
14	dobj	advice	833	4918.04	14	have	xcomp	1096	616.90
15	dobj	evidence	810	4088.47	15	answer	vmod	80	607.39
16	dobj	detail	735	3846.81	16	say	ccomp	1688	570.32
17	prt	away	754	3804.41	17	give	parataxis	138	546.08
18	dobj	name	960	3612.73	18	prepare	xcomp	216	541.64
19	prt	up	4219	3498.47	19	refuse	xcomp	222	514.50
20	dobj	smile	544	3479.92	20	impression	vmod	58	496.17

Table 6.6: Dependents and governors of the verb *give* and their dependency relations

strength for cooccurring words separately for each dependency relation, i. e. to create separate lists for the most strongly associated direct objects, indirect objects, subjects, etc. Such lists of relational cooccurrences filtered by type of relation are often called word sketches (Kilgarrieff and Tugwell, 2002; Kilgarrieff et al., 2004, 2014a).

To arrive at those lists, it is not sufficient to simply take the results of the previous case study and group the relation-word pairs by relation. Since, for a single list of cooccurring words, we are only interested in a single type of dependency relation, we have to classify all instances of this single type of dependency relation. For the previous case study, we classified all verbal relations, i. e. not only those of a single type. This corresponds to an unnecessary inflation of O_{22} and has unfavorable effects on many association measures (cf. Section 5.1.2). As an example, if we want to determine the association strength between *give* and *birth* for the list of direct objects of *give*, we have to classify all verb-based dobj relations according to the following two criteria: “Is *give* the governor of the relation?” and “Is *birth* the dependent of the relation?”. The corresponding query graphs are shown in Figure 6.7.

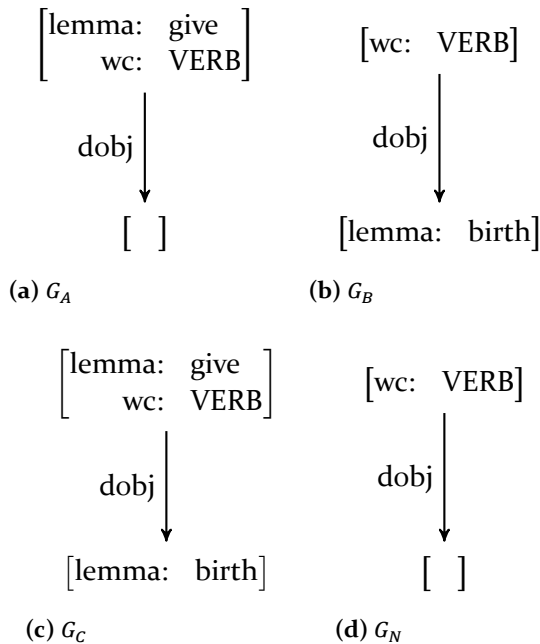


Figure 6.7: Query graphs for cooccurrences within a specific type of relation

Table 6.7 shows a word sketch for some of the outgoing dependency relations of *give*: Indirect objects, direct objects, prepositional phrases with *to* and passive nominal subjects. The most prominent difference between Table 6.6 and Table 6.7 is that (lexicographically less interesting) pronominal indirect objects rank very high in the former but (with two exceptions) not so in the latter. By constraining the sample size to the number of indirect object relations, the pronouns got ranked down. Together with the word sketch for some of the incoming dependency relations in Table 6.8, this gives a fairly comprehensive overview of words cooccurring with *give*, grouped by dependency relation.

6.3 Cooccurring structures

The cases studies above focus on cooccurrences of words or of words with relations. Here, we present further case studies that focus on larger structures. In Section 6.3.1, we revisit simple collexeme analysis (already reconstructed in Section 4.6.2), i. e. the cooccurrence of a word with a larger structure. In Section 6.3.2, we demonstrate how the generalized cooccurrence model allows for filtering relational cooccurrences by having additional relations on the words. In Section 6.3.3, we look at cooccurrences of larger structures, i. e. cooccurring structures where the smaller one is not just a single word.

6.3.1 Simple collexeme analysis

When we use a corpus to find out more about the usage of a word, it would be useful if we could go beyond simple collocations and word sketches and get an idea of the syntactic structures in which the word occurs. Luckily, simple collexeme analysis (cf. Section 2.2.1) enables us to determine the association strength between a word and any given linguistic structure. That means we “just” have to create a list of all structures that the word occurs in and can then proceed to performing simple collexeme analyses for all of them in order to rank them:

Since we do not know beforehand which of these constructions may be important, we should not constrain or predefine the productive language units beforehand, but *take all, arbitrarily large fragments of (previously experienced) utterance-analyses as possible units and let the statistics decide.*

Bod (1998: 5)

The main problem with this approach is the creation of the list of all cooccurring structures, of course. Using every single subgraph of every sentence the

6 Case studies

#	iobj	O_{11}	G
1	it	1689	421.72
2	child	168	126.80
3	she	994	89.54
4	people	256	81.83
5	everyone	44	54.89
6	side	49	41.78
7	pupil	54	40.80
8	parent	57	32.08
9	everybody	18	30.23
10	employee	39	28.68
11	team	38	27.42
12	bird	16	26.87
13	manager	30	24.29
14	someone	14	23.51
15	police	49	23.36
16	student	74	21.89
17	mother	36	21.87
18	matter	28	21.68
19	Labour	18	21.28
20	hair	12	20.15

#	prep_to	O_{11}	G
1	child	221	562.11
2	question	160	362.12
3	charity	73	295.47
4	people	248	236.22
5	patient	85	229.95
6	those	164	213.17
7	baby	42	172.82
8	candidate	41	146.81
9	poor	32	141.96
10	son	59	138.14
11	daughter	48	130.67
12	claim	51	121.03
13	authority	82	108.37
14	cause	42	95.23
15	police	66	93.20
16	person	82	88.16
17	pupil	37	87.98
18	student	54	86.01
19	boy	38	80.22
20	fear	24	75.82

#	dobj	O_{11}	G
1	rise	1908	14514.29
2	chance	1602	6560.41
3	impression	975	5691.76
4	birth	629	5178.68
5	opportunity	1220	4299.99
6	indication	532	4097.87
7	advice	833	3577.42
8	evidence	810	2819.92
9	detail	735	2687.38
10	smile	544	2592.56
11	notice	544	2465.60
12	priority	442	2295.25
13	name	960	2207.52
14	information	968	1985.67
15	answer	557	1844.06
16	laugh	328	1812.62
17	example	519	1768.18
18	support	763	1751.90
19	assurance	264	1610.78
20	lift	275	1508.79

#	nsubjpass	O_{11}	G
1	consideration	260	2287.47
2	detail	222	908.95
3	attention	174	776.68
4	example	182	715.29
5	priority	88	632.40
6	notice	102	581.26
7	reason	89	393.67
8	warning	56	359.66
9	instruction	68	339.39
10	Notice	41	313.85
11	thought	63	303.90
12	indication	41	277.26
13	permission	59	276.78
14	guidance	44	268.12
15	address	45	233.09
16	advice	52	231.73
17	answer	58	228.96
18	approval	46	224.67
19	information	133	216.44
20	preference	36	211.72

Table 6.7: Word sketch for some of the outgoing dependency relations of the verb *give*

word occurs in, without any constraints, is infeasible due to the enormous number of such subgraphs. Additionally, most of these subgraphs will be very rare and not particularly interesting (due to the little resemblance they bear to what linguists and lexicographers would typically call patterns of usage). There are two alternatives for coping with that situation: Either use a (probably manually curated) predetermined set of constructions or usage patterns or implement suitable constraints to reduce the number of subgraphs.

For the former alternative, we need an inventory of constructions or patterns (probably based on resources such as valency dictionaries, e. g. Herbst et al. (2004), or similar inventories of usage patterns, e. g. Hunston and Francis (2000)) and calculate the association between the word and each construction. The obvious advantage of this approach is that it will be much faster since we probably only have a few hundred candidates. Also, all of the candidates will be linguistically interesting. On the other hand, we miss out on the exploratory potential of the method if we limit ourselves to a predetermined set of constructions. If the word occurs in an interesting construction that is not in our inventory, we will not be able to find it. We can also expect to encounter practical difficulties. The inventory of constructions has to be stated in terms of the grammatical model used by the parser. That model will probably not provide all of the necessary information that is needed to adequately represent all of the constructions. An inventory based on a valency dictionary, for example, would require that the parser distinguishes between complements and adjuncts – a distinction which most parsers do not make.

For this case study, we implement the latter alternative, i. e. we introduce constraints to limit the amount of subgraphs that are taken into consideration. We use a structural constraint and a frequency threshold. Structurally, we limit the analysis to star-like subgraphs, i. e. subgraphs where all other vertices are adjacent to the center vertex. In our example using the verb *give*, this means that all words either govern or are governed by *give*. Note that there might be additional edges between the outer vertices. Since we are interested in syntactic patterns, we use delexicalized versions of those subgraphs, i. e. subgraphs that only contain labels for dependency relations and coarse-grained word classes.³ For further analysis, we only take into account subgraphs that occur in at least 10 different sentences. In the BNC,

³ Of course, we could also carry out an analysis for fully lexicalized subgraphs. Note, however, that this will be less rewarding due to sparseness problems since there are very few repetitions of lexicalized subgraphs that exceed a certain size (usually two or three words). If there are frequent subgraphs, they tend to be grammar-related, i. e. to consist of one or two lexical words plus auxiliaries, modals and articles. Another source for larger frequent subgraphs

6 Case studies

#	xcomp	O_{11}	G
1	prepare	216	256.30
2	refuse	222	229.82
3	require	215	186.38
4	combine	57	170.93
5	ask	284	144.52
6	willing	104	119.33
7	divide	19	91.14
8	integrate	15	90.77
9	design	142	88.34
10	charitable	8	79.99
11	decline	37	62.92
12	add	35	57.17
13	promise	55	52.89
14	offer	53	49.07
15	avoid	55	48.02
16	persuade	77	46.51
17	multiply	8	42.42
18	decide	175	40.82
19	call	72	37.89
20	please	30	31.38

#	rcmod	O_{11}	G
1	answer	113	666.63
2	advice	78	503.07
3	support	70	307.88
4	help	57	253.69
5	information	91	172.85
6	impression	41	165.74
7	reason	146	138.11
8	example	48	103.57
9	look	42	97.71
10	evidence	48	96.55
11	gift	22	88.26
12	p.	14	79.58
13	protection	17	72.27
14	lecture	13	70.44
15	account	28	68.05
16	goal	29	65.10
17	message	29	62.06
18	address	17	60.94
19	care	21	60.80
20	assistance	14	59.55

#	vmod	O_{11}	G
1	advice	109	801.01
2	reason	217	664.05
3	priority	76	574.66
4	name	124	563.49
5	answer	80	464.37
6	impression	58	391.18
7	support	76	296.00
8	example	83	294.35
9	attention	48	269.42
10	information	123	257.20
11	instruction	59	256.44
12	lecture	35	238.16
13	publicity	34	211.13
14	address	37	171.22
15	assurance	19	159.64
16	consideration	33	156.66
17	explanation	28	148.35
18	description	29	144.24
19	undertaking	29	143.29
20	account	39	132.21

#	ccomp	O_{11}	G
1	reply	152	550.05
2	give	89	122.14
3	recommend	46	66.15
4	hope	179	60.41
5	say	1688	53.00
6	demand	39	48.23
7	advice	8	44.16
8	lecture	8	31.74
9	attention	5	26.94
10	please	48	26.62
11	require	29	25.43
12	urge	14	23.55
13	reminder	9	23.28
14	information	9	22.29
15	answer	19	20.12
16	welcome	8	18.99
17	right	16	18.70
18	important	33	17.40
19	ensure	115	17.11
20	example	5	16.74

Table 6.8: Word sketch for some of the incoming dependency relations of the verb *give*

are parts of longer passages that occur several times in the corpus, cf. the *owl* and *urokinase* examples in Section 5.2.

give occurs in 26 495 different subgraphs, i. e. subgraph types, that satisfy those constraints.⁴

Even with the reduced number of subgraphs, the remaining collostructional analysis is still a compute-intensive task. We have to perform a simple collexeme analysis for each of the 26 495 subgraphs. The corresponding query graphs are shown in Figure 6.8. The query graphs G_A and G_N , corresponding to the frequencies R_1 and N , are always the same, i. e. we only have to perform those queries once. G_B and G_C , however, are always different. This means that, conceptually, we would have to perform 52 990 queries: 26 495 for G_B and 26 495 for G_C . Given that the two only differ in the one vertex label, we would have to check every single verb in the corpus 26 495 times. This is very inefficient. Instead, we can do a single pass through the corpus, looping over all sentences, and, for every sentence, extract all star-like subgraphs with verbs at the center. We delexicalize the subgraphs as described above and check for each subgraph if it is in the set of query graphs and count correspondingly.⁵

The results of this case study, i. e. the delexicalized subgraphs most strongly associated with the verb *give*, are shown in Table 6.9. As expected, the table shows that *give* is strongly associated with mono- and ditransitive uses⁶ as well as the prepositional dative. Indirect objects are usually pronouns, direct objects and subjects are either pronouns or nouns. Additional variation in the results table is due to auxiliaries and fragments of other associated structures.

6.3.2 Filtering relational cooccurrences

We can use the generalized cooccurrence model to filter relational cooccurrences by specifying additional incoming or outgoing dependency relations on the words. We can use this to create separate lists of collocation candidates for different syntactic functions of the word. In this case study, we look at the adjectival modifiers (amod relation) of the noun *bank* where *bank* is either a subject (nsubj relation) or a direct object (dobj relation).

⁴ Extracting those 26 495 subgraphs from the parsed version of the BNC took approximately 55 minutes using six 2.30 GHz cores.

⁵ For the analysis in the present case study, this still took approximately 660 minutes using twelve 2.30 GHz cores.

⁶ But note that, for example, the third entry (*give* plus direct object) does not necessarily correspond to monotransitive uses only. Among the 83 183 instances of *give* that have an outgoing dobj relation to a noun are, for example, 20 681 instances that have an additional outgoing iobj relation to a pronoun (cf. the second entry in the table).

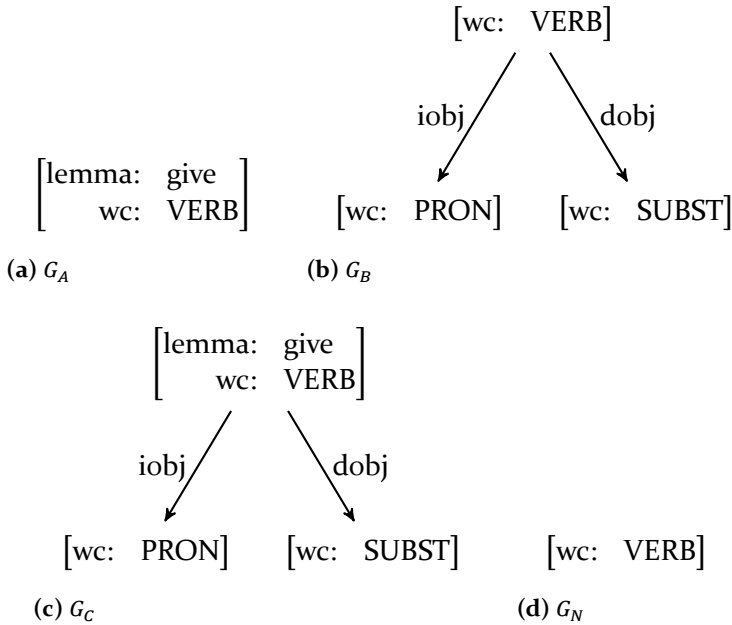


Figure 6.8: Query graphs for simple collexeme analysis

For a conventional analysis of adjectival modifiers of the noun *bank*, we would have to classify all noun-based amod relations according to whether *bank* is the governor and the collocation candidate in question, e.g. *central*, is the dependent. For the extended approach presented here, we still have to classify all noun-based amod relations, but we have to take into account the additional dependency relation, i.e. the incoming nsubj or dobj relation (in the example we use the nsubj relation). We have to classify all the noun-based amod relations according to the following two criteria: “Is *bank* the noun governing the relation and does it have an incoming nsubj relation?” and “Is *central* the dependent of the relation?” (note how the first criterion checks for two different properties of the relation). Figure 6.9 shows the corresponding queries. The additional dependency relation acts as a filter: We do no longer consider all instances of *bank* but only those governed by a nsubj relation.

The results are shown in Table 6.10 (note that we use a frequency threshold of $O_{11} \geq 3$ instead of $O_{11} \geq 5$ as in the other case studies). The adjectival modifiers of *bank* as nsubj and of *bank* as dobj show some very interesting differences. For *bank* as subject, all of the 20 most strongly associated adjectival modifiers refer to the “financial institute” sense of the word, whereas for *bank*

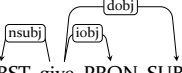
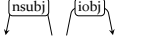
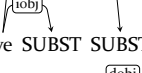
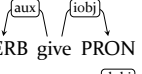
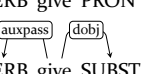
#	structure	O_{11}	G	#	structure	O_{11}	G
1		23185	247191.28	11		4861	50570.61
2		20681	221980.89	12		4420	46604.07
3		83183	189731.63	13		9147	46477.63
4		8590	102778.45	14		4440	44848.88
5		9712	100711.95	15		3826	38946.30
6		7987	97445.06	16		19767	37645.91
7		8579	89674.44	17		22844	34490.43
8		6657	67774.09	18		21069	34249.55
9		5793	59584.79	19		14595	33724.28
10		8082	55477.27	20		3125	32343.21

Table 6.9: Syntactic structures associated with the verb *give*

as direct object, there are also some modifiers that clearly refer to the “river bank” sense, e. g. *steep*, *grassy*, *far* or *muddy*.

6.3.3 Cooccurrences of larger structures

Structures associated with give damn

Now, let us take a look at cooccurrences of larger structures. In our first example, we look at larger structures that are associated with the structure “verb *give* with direct object *damn*” (Figure 6.10a). As in Section 6.3.1, we have to limit the amount of structures we take into account and we do it in a similar manner. Instead of allowing all arbitrary subgraphs of sentences in which *give damn* occurs, we only select those subgraphs that consist of

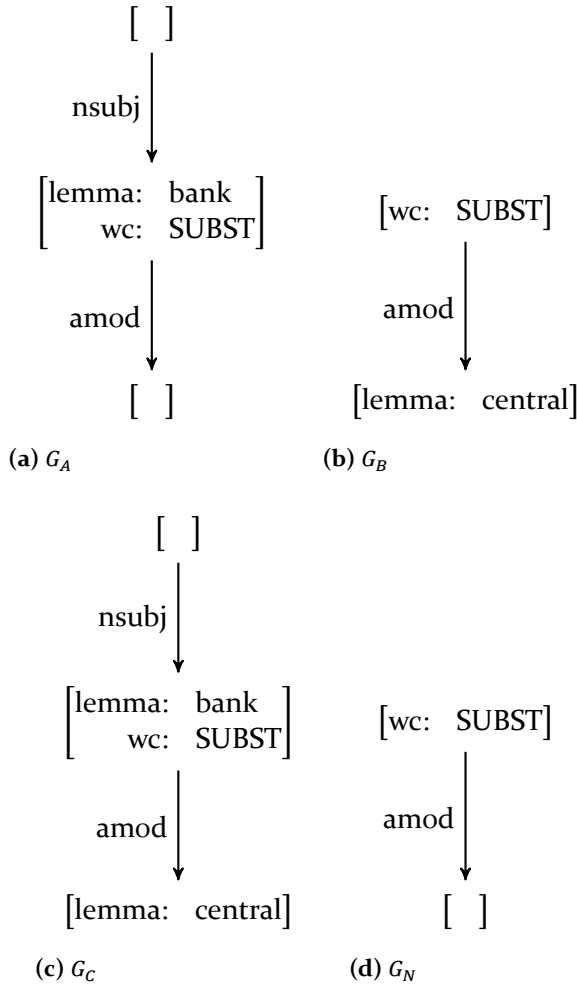


Figure 6.9: Query graphs for adjectival modifiers of *bank* as subject

vertices adjacent to either *give* or *damn*, i. e. a generalization of the concept of a star-like subgraph used in Section 6.3.1.

To determine the association between *give damn* and another larger structure, e. g. the one shown in Figure 6.10b, we have to classify all direct object relations where a verb governs a noun (Figure 6.10d) according to the following two (complex) criteria: “Is *give* the governor of the relation and is *damn* the dependent of the relation?” and “Does the governor of the relation have an outgoing *nsubj* relation governing a pronoun and an outgoing *neg* relation governing an adverb and does the dependent of the relation have an outgoing

#	amod	O_{11}	G	#	amod	O_{11}	G
1	central	162	1729.17	1	central	69	691.24
2	commercial	64	606.70	2	national	31	202.89
3	japanese	39	361.17	3	steep	11	123.29
4	acquire	15	190.47	4	foreign	17	106.63
5	french	31	185.65	5	commercial	14	97.23
6	european	33	179.01	6	european	17	91.52
7	pay	16	175.93	7	overseas	10	91.19
8	foreign	25	138.45	8	grassy	6	81.54
9	most	30	132.90	9	out-of-state	3	76.20
10	german	22	127.21	10	largest	9	60.45
11	lead	19	119.07	11	far	6	48.75
12	national	27	117.58	12	regional	8	48.51
13	big	27	110.25	13	big	12	43.90
14	american	21	97.93	14	independent	7	35.77
15	individual	18	76.06	15	computerised	3	32.02
16	international	18	75.85	16	muddy	3	31.36
17	swiss	8	69.86	17	local	14	30.29
18	largest	12	68.55	18	acquire	3	29.69
19	participate	6	66.39	19	weak	4	29.11
20	independent	13	65.40	20	worldwide	3	24.31

Table 6.10: Adjectival modifiers of *bank* as nsubj (left) and dobj (right)

det relation governing a determiner?”. All corresponding query graphs are shown in Figure 6.10.

For the analysis, we use a frequency threshold of $O_{11} \geq 2$. The results are shown in Table 6.11 and point toward a single highly associated structure (possibly with a few variants): Pronoun-auxiliary-negation marker-give-determiner-damn-about-noun, i. e. “someone does not give a damn about something”.

Adverb-verb-direct object triples

For the second example, we want to find strongly associated triples of adverbs, verbs and direct objects like *properly do job* or *slowly shake head*. As has been mentioned before, the generalized cooccurrence model presented here is a generalization of the traditional approach to the analysis of cooccurrences based on two-dimensional contingency tables. As such, we cannot directly analyze the association between three or more items and have to resort to other means of analysis. Here, we present two possible approaches.

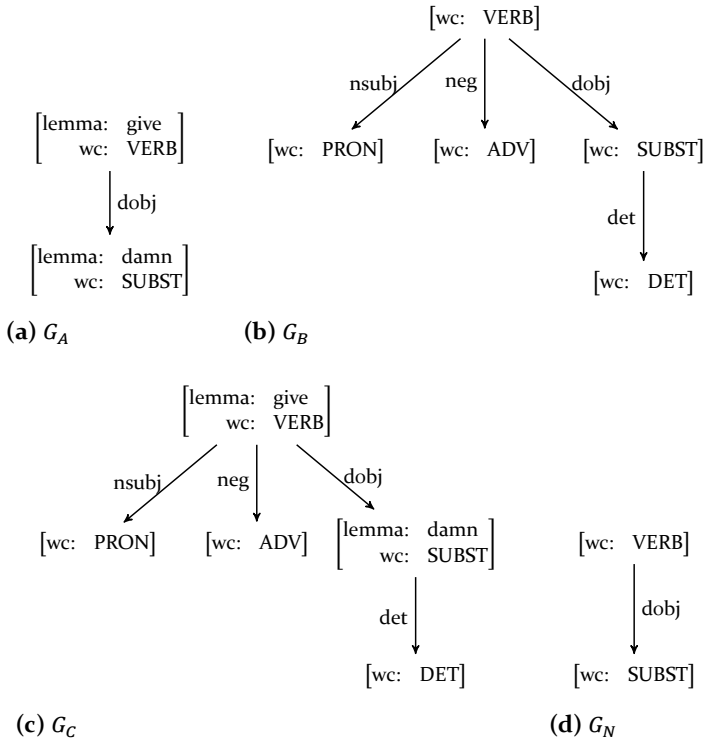


Figure 6.10: Query graphs for structures associated with *give damn*

The first one is to analyze the association between the direct object and adverb-verb pairs within the syntactic structure (Figure 6.11). This means we classify all instances of verbs with an outgoing *advmod* relation to an adverb and an outgoing *dobj* relation to a noun (Figure 6.11d) according to the following two criteria (the example uses the triple *slowly shake head*): “Is *head* the dependent of the *dobj* relation?” (Figure 6.11a) and “Is the verb slot filled by *shake* and is *slowly* the dependent of the *advmod* relation?” (Figure 6.11b).

The second approach analyzes the association between the structures “verb with direct object *head*” (Figure 6.12a) and “*shake* with adverbial modifier *slowly*” (Figure 6.12b). This is accomplished by classifying all verbs according to the following two criteria: “Does the verb have an outgoing *dobj* relation to the noun *head*?” and “Is the verb slot filled by *shake* and does it have an outgoing *advmod* relation to the adverb *slowly*?”. The queries are shown in Figure 6.12.

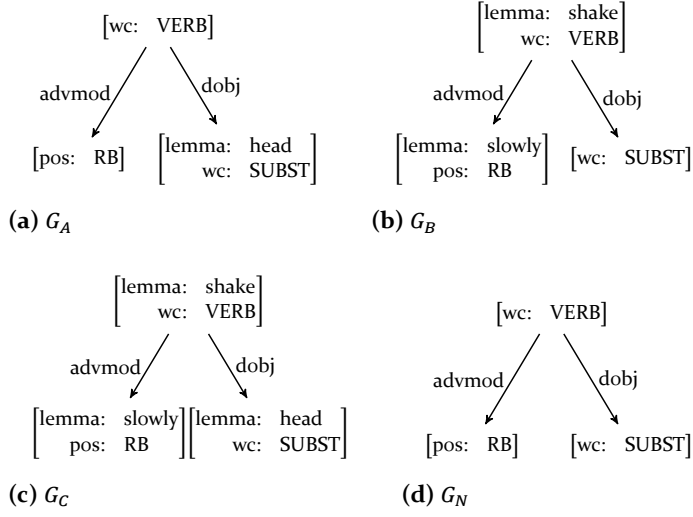


Figure 6.11: Query graphs for the first approach

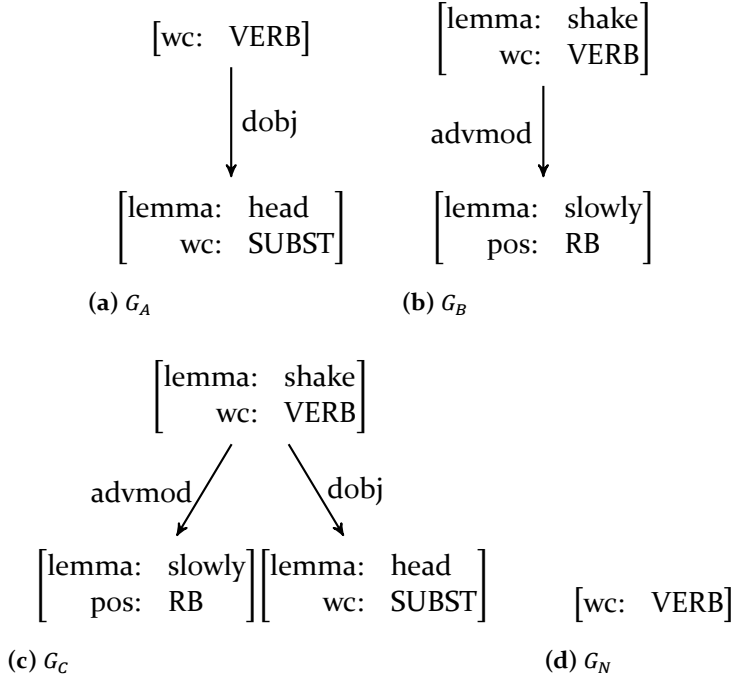


Figure 6.12: Query graphs for the second approach

6 Case studies

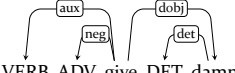
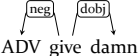
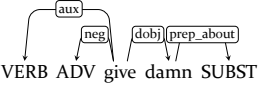
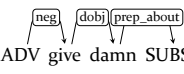
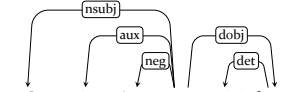
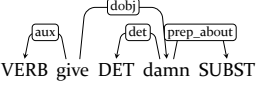
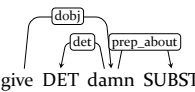
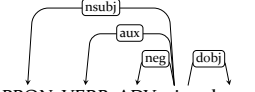
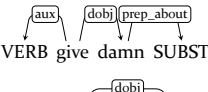
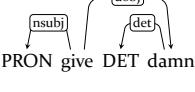
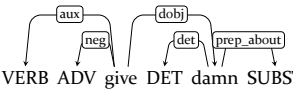
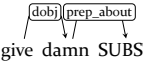
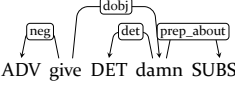
#	structure	O_{11}	G	#	structure	O_{11}	G
1		119	1165.15	11		117	564.75
2		111	1089.96	12		92	505.44
3		119	930.02	13		28	494.96
4		90	906.00	14		28	482.98
5		89	904.79	15		30	472.03
6		111	874.38	16		36	454.15
7		89	731.02	17		30	386.91
8		90	729.43	18		106	364.01
9		28	580.18	19		36	346.64
10		28	567.00	20		150	342.65

Table 6.11: Structures associated with *give damn*

The results for both approaches are shown in Table 6.12 (first approach to the left, second approach to the right). As we can see, there is a fairly high overlap between the twenty highest ranking triples obtained by the two approaches

#	adverb	verb	noun	O_{11}	G	#	adverb	verb	noun	O_{11}	G
1	together	get	act	120	1953.88	1	down	put	phone	138	2704.08
2	down	put	phone	138	1944.24	2	back	make	way	146	2209.41
3	back	make	way	146	1648.42	3	together	get	act	120	2189.75
4	also	play	part	120	1334.65	4	also	play	part	120	1775.89
5	slowly	shake	head	83	1253.92	5	slowly	shake	head	83	1737.18
6	properly	do	job	90	1252.29	6	then	shake	head	88	1609.88
7	then	shake	head	88	1153.50	7	properly	do	job	90	1391.90
8	away	take	breath	86	1137.54	8	also	play	role	90	1374.49
9	just	round	corner	55	1102.18	9	just	round	corner	55	1352.23
10	also	play	role	90	1021.29	10	away	take	breath	86	1316.20
11	so	do	i.	56	880.02	11	also	have	effect	143	1168.34
12	back	turn	clock	50	874.93	12	sadly	shake	head	50	1055.72
13	straight	set	record	47	847.60	13	straight	set	record	47	1022.38
14	back	get	money	109	763.03	14	now	take	place	100	995.07
15	sadly	shake	head	50	754.39	15	only	take	place	89	988.27
16	away	turn	head	71	703.76	16	also	take	place	93	914.05
17	also	express	concern	46	682.05	17	back	turn	clock	50	898.28
18	back	push	chair	48	677.93	18	back	push	chair	48	876.18
19	away	give	game	59	674.30	19	away	give	game	59	856.70
20	back	come	flooding	32	659.76	20	so	do	i.	56	855.85

Table 6.12: Adverb-verb-direct object triples obtained by both approaches

(Dice coefficient 0.80) and a Spearman correlation of 0.95 suggests that the two approaches could be used interchangeably.⁷

Table 6.12 contains a few triples with the noun *head*. In Table 6.13 we show more comprehensive lists of triples with that noun (as above, results for the first approach are shown on the left side, results for the second on the right). Again, both the top 20 lists and the overall ranking are very similar (Dice coefficient: 0.90, Spearman correlation: 0.93). Such a list gives a nice overview of what is usually done with a head and in which manner and could probably serve as the basis for a corresponding dictionary entry.

6.4 Summary

In this chapter, we presented a number of case studies that highlighted some of the possibilities of the generalized cooccurrence model. For cooccurring words, the examples included reconstructions of conventional relational cooccurrences based on all relations and based on verbal relations as well as reinterpretations of relational cooccurrence as node-based or collocate-based

⁷ The triple *so do I*. that occurs in both top 20 lists is due to a parser error that probably originated in a wrong segmentation and resulted in a wrong analysis.

#	adverb	verb	noun	O_{11}	G	#	adverb	verb	noun	O_{11}	G
1	slowly	shake	head	83	1253.92	1	slowly	shake	head	83	1737.18
2	then	shake	head	88	1153.50	2	then	shake	head	88	1609.88
3	sadly	shake	head	50	754.39	3	sadly	shake	head	50	1055.72
4	away	turn	head	71	703.76	4	away	turn	head	71	740.62
5	back	throw	head	40	425.65	5	down	keep	head	45	602.72
6	again	shake	head	34	413.16	6	again	shake	head	34	555.85
7	just	shake	head	28	363.00	7	back	throw	head	40	547.41
8	down	keep	head	45	347.63	8	just	shake	head	28	492.21
9	slightly	turn	head	22	274.38	9	slightly	turn	head	22	349.44
10	slowly	turn	head	26	270.81	10	down	put	head	37	344.47
11	slightly	shake	head	17	245.03	11	slowly	turn	head	26	331.65
12	violently	shake	head	19	239.10	12	violently	shake	head	19	301.38
13	vigorously	shake	head	16	194.09	13	slightly	shake	head	17	277.14
14	back	tilt	head	15	193.63	14	slowly	raise	head	18	275.35
15	quickly	shake	head	14	193.57	15	vigorously	shake	head	16	263.18
16	slowly	raise	head	18	190.48	16	quickly	shake	head	14	260.30
17	then	bow	head	11	165.71	17	back	tilt	head	15	254.73
18	back	tip	head	14	157.19	18	back	pull	head	25	225.61
19	down	get	head	22	155.38	19	back	tip	head	14	224.77
20	down	put	head	37	153.29	20	emphatically	shake	head	10	211.09

Table 6.13: Adverb-verb-direct object triples for the noun *head*

collostructional analyses. An evaluation against the Oxford Collocations Dictionary showed that for the verb *give*, the conventional approach based on verbal relations seems to work best.

In another three case studies, we explored cooccurrences that include a specific type of dependency relation. We looked at the association between a word and its incoming or outgoing dependency relations, at cooccurrences of a word with associated relation-word pairs and at word sketches, i. e. relational cooccurrences grouped by relation. We also argued why the last case study needs its own analysis and cannot simply filter the results of the second.

Finally, we looked at cooccurrences of larger structures. We discussed some of the practical challenges that a large-scale collostructional analysis looking for the syntactic structures associated with a word poses and presented results for the verb *give*. We showed how the possibilities of the generalized cooccurrence model can be used to filter the relational cooccurrences of a word by the syntactic function of that word and we gave examples for analyzing the cooccurrences of structures where both consist of more than one word.

7 Conclusion and future work

7.1 Conclusion

As argued in the introduction (Chapter 1), the study of cooccurrences has had a profound impact on our view of language. It contributed greatly to the insight that semi-preconstructed phrases and item-specific knowledge are central to how language works. This importance is mirrored by the fact that cooccurrences of various items have been studied under a variety of names, such as collocation, colligation or collostruction. The analysis of cooccurrences also plays a central part in the identification of multiword units, the extraction of valency patterns and in investigations into patterns of language use that are typically represented as hybrid n-grams involving both lexical and grammatical items.

For cooccurrences of words, there are several well-understood and fully worked out statistical models that are based on different notions of cooccurrence (Section 2.1): Cooccurrence within some segment of text, cooccurrence within some orthographically defined distance and cooccurrence via some grammatical relation. The predominant approach to the statistical analysis of cooccurrences – although some criticism has been leveled against it – is based on 2×2 contingency tables that represent the cross-classification of the units of analysis (Chapter 3). We can use the marginal frequencies of the contingency table to compute the frequencies that we would expect under a null hypothesis of independence. The observed frequencies in the contingency table and the expected frequencies derived from it can be used to calculate a wide range of association measures.

For cooccurrences of words with larger grammatical structures and for cooccurrences of grammatical structures with grammatical structures (Section 2.2), e. g. simple and covarying collexeme analysis, hybrid n-grams, valency pattern extraction and certain approaches to multiword collocations, there are typically no statistical models that are formalized to the same degree as for cooccurrences of words. It is not always clear how exactly an approach should be implemented, what the units of analysis are and how the frequencies are determined. The approaches are also usually tailored to specific needs, i. e. they are restricted in the kind of cooccurrences they can

analyze. None of the approaches offers a sound methodological framework for the analysis of cooccurrences of arbitrary linguistic structures.

This gap has been filled by the current work.

We formally represent linguistic structures as labeled connected directed graphs where the labels are organized as a partially ordered set (Section 4.1). Graph-based representations of linguistic structures are very versatile and flexible and allow for the linguistic information to be modeled in many different ways (cf. the examples in Section 4.1.7). We build on the concept of subgraph isomorphism and on the partial order of the labels to formally define the subsumption relation on linguistic structures. A more general linguistic structure is said to subsume a more specific one and we call the mapping from the former to the latter an embedding.

As pointed out in Section 4.2, determining the frequency of a linguistic structure in a corpus is a non-trivial task. We present and formally define four different counting methods and provide an additional generalized perspective on the counting methods where we show that all four can be formally represented as partitions of the set of embeddings.

In Section 4.3, we develop a preliminary cooccurrence model and show in Section 4.4 that two problems can arise when cooccurrences of linguistic structures are counted and cross-classified: The problem of inflated frequencies, where, due to the effects of combination and permutation, the same tokens are counted multiple times, and the inconsistency problem, where a unit of analysis is an instance of both linguistic structure G_A and linguistic structure G_B but not of their cooccurrence G_C . We show that the different counting methods are affected by those problems to different degrees.

The main contribution of the current work is the generalized cooccurrence model that is developed and formalized in Section 4.5 and that allows for the analysis of cooccurrences of two arbitrary linguistic structures. We require that two cooccurring linguistic structures overlap in at least one vertex, i. e. that the cooccurrence of the two is also a linguistic structure (and therefore connected). The key idea of the generalized cooccurrence model is to cross-classify the instances of that overlapping part. The formalization takes the generalized counting methods into account and suggests workarounds for coping with the inconsistency problem.

We show that relational cooccurrences, segment-based cooccurrences, simple collexeme analysis and covarying collexeme analysis are merely special cases of the generalized cooccurrence model (Section 4.6). We also show that some seeming limitations of the generalized cooccurrence model can be overcome by introducing additional vertices and edges to the graph structure

(Section 4.7). This further increases the range of possible analyses and allows, for example, to capture the linear order of the tokens in the text or to search for elements that are not there, e. g. for instances of *give* that do not have an outgoing indirect object relation, i. e. for monotransitive uses of the verb *give*.

The evaluation of the generalized cooccurrence model (Chapter 5) has a double focus. On the one hand, we perform thought experiments on popular association measures to analyze how they react to inflated frequencies. This shows that the association measures fall into three classes with quite different behaviors. On the other hand, empirical investigations show that inflated frequencies are not a major problem when working with authentic language data and that the overall impact of the different counting methods is limited.

In a series of case studies ranging from reconstructions of traditional collocational and collocation analyses to cooccurrences of structures that both consist of more than one word, we highlight the possibilities and the flexibility of the generalized cooccurrence model (Chapter 6).

The generalized cooccurrence model gives a formal definition of linguistic structures. It gives a formal definition of the cooccurrence of two linguistic structures. It gives formal definitions of several methods for counting linguistic structures. And it gives a formal definition of how to populate a 2×2 contingency table to which a wide range of association measures can be applied. Together, those definitions completely and unambiguously describe a methodology for analyzing cooccurrences of arbitrary linguistic structures. The generalized cooccurrence model operates on graphs. This opens up the possibility to have preprocessing steps that introduce additional vertices and edges and makes the model highly flexible.

7.2 Future work

The generalized cooccurrence model is a generalization of the standard approach to the study of cooccurrences that is based on the statistical analysis of 2×2 contingency tables. As such, it inherits the shortcomings discussed in Section 3.4.3. An obvious route for future work would be to address those shortcomings in the generalized cooccurrence model by adapting some of the solutions suggested so far in the literature. This could include implementing support for association measures that go beyond 2×2 contingency tables, such as lexical gravity, or extending the model to allow for cooccurrences of three or more linguistic structures.

In Section 4.7, we show that things that seem to be impossible with the generalized cooccurrence model become possible if we introduce additional

vertices or edges to the graph structure. The elegance of that approach is that we do not need to extend the formal model. We could, however, also explore alternative ways of dealing with those limitations that do not require additional preprocessing steps. If we extended the model to allow for additional constraints on vertices and edges that are not part of the graph structure, we could increase its expressiveness in a similar way. If we had a way to incorporate such constraints, we could, for example, disallow a vertex $v \in V$ to have an outgoing dependency relation labeled *iobj* by requiring all edges starting in v to not have a label equal to or more specific than *iobj*, i. e. $\forall(e \in E | s(e) = v), \lambda(e) \not\geq \text{iobj}$ or, equivalently, $\neg \exists(e \in E | s(e) = v), \lambda(e) \geq \text{iobj}$.

The availability of the generalized cooccurrence model opens up a wide range of applications. And while the examples in the current work are mainly concerned with cooccurrences of syntactic structures, the method itself is not limited to linguistic data but can be applied to anything that can be represented as graphs. Non-linguistic applications that come to mind include the analysis of associations in social graphs, e. g. Facebook or Twitter, or semantic networks and knowledge graphs, e. g. Freebase or Wikidata.

The generalized cooccurrence model could have applications in typical NLP pipelines. Wehrli (2014), for example, shows that knowledge of collocations improves the performance of a syntactic parser. If, thanks to the generalized cooccurrence model, the parser could not only leverage association strengths between words but also between larger syntactic structures, then this could further improve parser performance.

The methodological framework of the generalized cooccurrence model could also benefit psycholinguistic studies. Bel Enguix et al. (2014), for example, find that human word associations can be modeled by cooccurrence networks. Making use of the generalized cooccurrence model to quantify association strengths between more complex linguistic structures may allow for more extensive models of human associative behavior.

The most obvious applications of the generalized cooccurrence model are in corpus linguistics, terminology extraction and lexicography. Having a sound formal model for the analysis of cooccurrences of arbitrary linguistic structures is an excellent basis for extracting and analyzing phenomena like multiword expressions, valency patterns and formulaic language in general. The model could also be applied to cross-modal cooccurrences, e. g. to cooccurrences between verbal utterances and gestures.

The mechanics of lexicogrammatical interaction are not yet fully understood and explained. We hope that the generalized cooccurrence model

presented here will prove to be a useful tool in the endeavor to find, analyze and describe the relevant phenomena.

A Notes on the implementation

A collection of command line tools, called the Pareidoscope, that implements the generalized cooccurrence model is available from GitHub (<https://github.com/tsproisl/Pareidoscope>). With these tools, a wide range of analyses can be performed, including collocational and collostructional analyses and the analysis of larger linguistic structures.

A.1 Installation

The Pareidoscope is also available via PyPI, the Python Package Index (<https://pypi.python.org/pypi/Pareidoscope>), and can be installed using pip:

```
pip3 install Pareidoscope
```

Alternatively, you can download and decompress the latest release (<https://github.com/tsproisl/Pareidoscope/releases/latest>) or clone the git repository:

```
git clone https://github.com/tsproisl/Pareidoscope.git
```

In the new directory, run the following command:

```
python3 setup.py install
```

A.2 Usage

A.2.1 Input formats

Corpora

Currently, corpora can be provided in two different formats: In CoNLL-U format or in CWB-treebank format.

CoNLL-U is the format used for the treebanks of the Universal Dependencies project (Nivre et al., 2016). The format is specified in the UD documentation (<http://universaldependencies.org/format.html>). Here is an example that has been adapted from the documentation:

1	They	they	PRON	PRP	—	2	nsubj	2:nsubj 4:nsubj	—
2	buy	buy	VERB	VBP	—	0	root	0:root	—
3	and	and	CONJ	CC	—	4	cc	4:cc	—
4	sell	sell	VERB	VBP	—	2	conj	0:root 2:conj	—
5	books	book	NOUN	NNS	—	2	obj	2:obj 4:obj	—
6	.	.	PUNCT	.	—	2	punct	2:punct	—

There are ten tab-separated columns. The first five columns are for the word ID, the word form, the lemma, the universal part-of-speech tag and a language-specific part-of-speech tag. Columns six and ten, which are empty in this example, are for morphological features and miscellaneous annotation. In columns seven to nine, the dependency analysis of this sentence is encoded. Columns seven and eight encode the basic dependencies which are required to form a tree. Column seven indicates the ID of the governor, column eight the type of the dependency relation between the governor and the current word. In column nine, an enhanced dependency graph can be represented that does not need to be a tree.

For details on the CWB-treebank format, cf. Proisl and Uhrig (2012).

Queries

The query graphs can be provided as JSON serializations of the node-link format understood by NetworkX (Hagberg et al., 2008). All command-line tools can operate on multiple queries, therefore a list of queries has to be provided, even for a single query. Here is an example of a one-element list containing the query graph for finding associated larger structures of monotransitive uses of the verb *give* with `pareidoscope_associated_structures`:

```
[
  {
    "graph": {
      "description": "Monotransitive uses of the verb give"
    },
    "nodes": [
      {
        "id": 0,
        "wc": "VERB",
        "lemma": "give",
        "focus_point": true,
        "not_outdep": ["iobj", "obl"]
      },
      {
        "id": 1
      }
    ]
  }
]
```

```

    "links": [
      {
        "source": 0,
        "target": 1,
        "relation": "obj"
      }
    ]
  }
]

```

Queries are represented as dictionaries with two obligatory keys: `nodes` for the vertices and `links` for the edges. Under the key `graph`, additional information such as a description of the query can be stored. Both the vertices and the edges of the query graph are represented as lists of dictionaries. An edge is specified by the IDs of its source and target vertices and, optionally, by the kind of dependency relation. The vertices are required to have an ID and can have other, optional attributes.

The attributes that can be used for the vertices depend on the kind of query. The following attributes can always be used: `word`, `pos`, `lemma`, `wc`, `root`, `not_indep` (a list), `not_outdep` (a list). The first five attributes can also be negated by prefixing them with `not_`, e.g. `"not_wc": "NOUN"` for indicating that a vertex should not be a noun.

For determining the association strength between two structures with `pareidoscope_association_strength`, the following additional attributes can be used. The attribute `query` has to be used for every vertex and takes the values `A`, `B` or `AB`. This attribute indicates if the vertex belongs to G_A , G_B or to both, i.e. to G_C . For vertices marked as `"query": "AB"`, the optional attributes `only_A` and `only_B` can be used. These attributes are lists and indicate which other attributes only apply to G_A or to G_B . The focus point vertex of the graph can be marked by setting `"focus_point": true`. The attributes `only_A` and `only_B` can also be used for edges.

For simple collexeme analysis with `pareidoscope_collexeme_analysis`, the attribute `collo_item` has to be set to `true` for the collexeme vertex. This vertex is automatically the focus point.

For relational cooccurrences and covarying collexeme analysis with `pareidoscope_covarying_collexemes`, the attributes `collo_A`, `collo_B` have to be set to `true` for the two collexeme vertices. The attribute `focus_point` can be used to mark the focus point vertex.

For finding associated larger structures with `pareidoscope_associated_structures`, the focus point vertex can be marked by setting `"focus_point": true`.

A.2.2 Convert a corpus into an SQLite3 database

For most of the programs described below, it is necessary to convert your corpus into an SQLite3 database. This can considerably speed up highly selective queries; for very general queries that require that almost every sentence in the corpus is checked, this makes less of a difference.

Corpora in CoNLL-U or CWB-treebank format can be converted to an SQLite3 database using `pareidoscope_corpus_to_sqlite`. Running the program with the option `-h` outputs a help message with detailed usage information. Here is an example where we convert the training part of the English Universal Dependencies treebank (`en-ud-train.conllu`),¹ which is in CoNLL-U format, and create the database `en-ud-train.db`:

```
pareidoscope_corpus_to_sqlite --db en-ud-train.db \
    --format conllu en-ud-train.conllu
```

A.2.3 Association between two linguistic structures

The program `pareidoscope_association_strength` determines the association strength between two linguistic structures.

Here is a sample query for the cooccurrence of the ditransitive with direct objects that have a determiner (this query and other queries can be found in the query file `ex_association_two_structures.json`):

```
[
  {
    "graph": {
      "description": "cooccurrence of the ditransitive with direct
                     objects that have a determiner"
    },
    "nodes": [
      {
        "id": 0,
        "wc": "VERB",
        "query": "AB",
        "focus_point": true
      },
      {
        "id": 1,
        "query": "A"
      },
      {
        "id": 2,
```

¹https://github.com/UniversalDependencies/UD_English; we use the version included in the 2.0 release of the UD treebanks (<http://hdl.handle.net/11234/1-1983>).

```

        "wc": "NOUN",
        "query": "AB"
    },
    {
        "id": 3,
        "query": "B"
    }
],
"links": [
    {
        "source": 0,
        "target": 1,
        "relation": "iobj"
    },
    {
        "source": 0,
        "target": 2,
        "relation": "obj"
    },
    {
        "source": 2,
        "target": 3,
        "relation": "det"
    }
]
}
]

```

The verb *the* and direct object are part of both linguistic structures and are therefore marked as AB. The indirect object only belongs to the ditransitive and is marked as A, the determiner only belongs to the other linguistic structure and is marked as B. Additionally, the verb is marked as the focus point vertex.

Here is an example for invoking the program (use the option `-h` for detailed usage information):

```
pareidoscope_association_strength --format db -o associations \
    en-ud-train.db ex_association_two_structures.json
```

In this example, we run the queries specified in `ex_association_two_structures.json` on the corpus converted in Section A.2.2. Option `--format db` indicates that we operate on an SQLite3 database (this program can also operate directly on corpus files in CoNLL-U or CWB-treebank format). The results are written to `associations.tsv` in a tab-separated format and contain, for every query and every counting method, the frequencies O_{11} , R_1 , C_1 and N , the number of inconsistencies and three association measures (log-likelihood, t -score, Dice coefficient).

A.2.4 Simple collexeme analysis

The program `pareidoscope_collexeme_analysis` performs a simple collexeme analysis, i. e. it determines the association strength between a linguistic structure and the word forms or lemmata that occur in a given slot of that structure. To this end, the `collo` item vertex has to be marked with `"collo_item": true` in the query. Here is an example query (taken from the query file `ex_collexeme_analysis.json`) that finds verbs that are associated with the ditransitive:

```
[
  {
    "graph": {
      "description": "Verbs associated with the ditransitive"
    },
    "nodes": [
      {
        "id": 0,
        "wc": "VERB",
        "collo_item": true
      },
      {
        "id": 1
      },
      {
        "id": 2
      }
    ],
    "links": [
      {
        "source": 0,
        "target": 1,
        "relation": "iobj"
      },
      {
        "source": 0,
        "target": 2,
        "relation": "obj"
      }
    ]
  }
]
```

Here is an example for invoking the program (use the option `-h` for detailed usage information):

```
pareidoscope_collexeme_analysis -o collexemes en-ud-train.db \
  ex_collexeme_analysis.json
```

In this example, we run the collection of queries specified in `ex_collexeme_analysis.json` on the corpus converted in Section A.2.2. The program takes an optional option `-c` where we can specify if the `collo` items should be word forms or lemmata (the latter is the default).

The results are written to `collexemes.tsv` in a tab-separated format and contain, for every query and cooccurring lemma, the frequencies O_{11} , R_1 , C_1 and N and three association measures (log-likelihood, t -score, Dice coefficient). As pointed out in Section 4.6.2, three of the four counting methods are fully equivalent for simple collexeme analysis. Since counting sentences does not make much sense in this case because of the large number of inconsistencies that can be expected, we do not include that counting method. As a consequence, we do not need to distinguish between different counting methods and do not need to include a field for inconsistencies. The results are ordered by log-likelihood.

A.2.5 Relational cooccurrences and covarying collexeme analysis

The program `pareidoscope_covarying_collexemes` performs a covarying collexeme analysis which, for linguistic structures that consist of a single dependency relation, is equivalent to analyzing relational cooccurrences. The program determines the association between the word forms or lemmata that cooccur in two slots of a linguistic structure. To this end, the two slots have to be marked with `"collo_A": true` and `"collo_B": true` in the query. Here is an example query (taken from the query file `ex_covarying_collexemes.json`) that determines the association between the verbs in the *into*-causative:

```
[
  {
    "graph": {
      "description": "Into-causative, i.e. verb someone into verbing"
    },
    "nodes": [
      {
        "id": 0,
        "wc": "VERB",
        "collo_A": true
      },
      {
        "id": 1,
        "pos": "VBG",
        "collo_B": true
      }
    ]
  }
]
```

A Notes on the implementation

```
{
  {
    "id": 2
  },
  {
    "id": 3,
    "lemma": "into"
  }
],
"links": [
  {
    "source": 0,
    "target": 1,
    "relation": "advcl"
  },
  {
    "source": 0,
    "target": 2,
    "relation": "obj"
  },
  {
    "source": 1,
    "target": 3,
    "relation": "mark"
  }
]
}
```

Here is an example for invoking the program (use the option `-h` for detailed usage information):

```
pareidoscope_covarying_collexemes -o covarying en-ud-train.db \
  ex_covarying_collexemes.json
```

In this example, we run the collection of queries specified in `ex_covarying_collexemes.json` on the corpus converted in Section A.2.2. The program takes an optional option `-c` where we can specify if the cooccurring items should be word forms or lemmata (the latter is the default).

The results are written to `covarying.tsv` in a tab-separated format and contain, for every query, cooccurring pair of items and counting method, the frequencies O_{11} , R_1 , C_1 and N , the number of inconsistencies and three association measures (log-likelihood, t -score, Dice coefficient). The results are ordered by log-likelihood for counting focus points.

A.2.6 Associated larger structures

The program `pareidoscope_associated_structures` determines which larger delexicalized linguistic structures are associated with the query structure. It considers all star-like larger structures, i. e. structures where all new vertices have to be adjacent to a query vertex, that cooccur with the query structure in at least `--min-coocc` sentences (default: 5) and have a maximum of `--max-size` vertices (default: 7). The vertices of the larger structures are delexicalized and contain only word class information (the `wc` attribute). Here is an example query that looks for larger structures that are associated with monotransitive uses of the verb *give*:

```
[
  {
    "graph": {
      "description": "Monotransitive uses of the verb give"
    },
    "nodes": [
      {
        "id": 0,
        "wc": "VERB",
        "lemma": "give",
        "focus_point": true,
        "not_outdep": ["iobj", "obl"]
      },
      {
        "id": 1
      }
    ],
    "links": [
      {
        "source": 0,
        "target": 1,
        "relation": "obj"
      }
    ]
  }
]
```

Here is an example for invoking the program (use the option `-h` for detailed usage information):

```
pareidoscope_associated_structures -o assoc_struct \
  en-ud-train.db ex_associated_structures.json
```

In this example, we run the collection of queries specified in `ex_associated_structures.json` on the corpus converted in Section A.2.2.

The results are written to `assoc_struct.tsv` in a tab-separated format and contain, for every query, associated larger structure and counting method, the frequencies O_{11} , R_1 , C_1 and N , the number of inconsistencies and three association measures (log-likelihood, t -score, Dice coefficient). The results are ordered by log-likelihood for counting focus points.

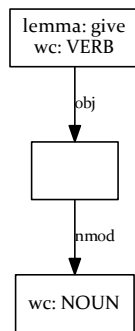
Visualizing associated structures

The associated larger structures output by `pareidoscope_associated_structures` are in the same node-link format as the query graphs and can be visualized with the program `pareidoscope_draw_graphs`. Note that this requires that Graphviz and the Python package PyDotPlus are installed on your computer.

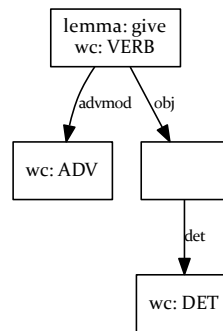
Here is an example for invoking the program (use the option `-h` for detailed usage information):

```
tail -n +2 assoc_struct.tsv | head | cut -f2 | \
  pareidoscope_draw_graphs -o draw -
```

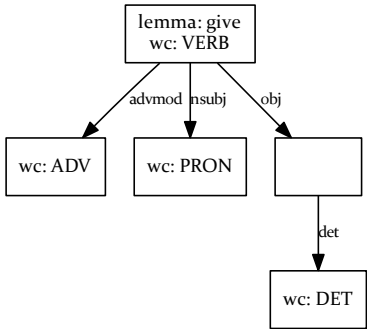
In this example, we use the output file created by the previous command, extract the ten most strongly associated larger structures (using GNU `coreutils`) and draw them. The images are written to the directory `draw`. Figure A.1 displays the visualizations created for the four larger structures that are most strongly associated with monotransitive *give*.



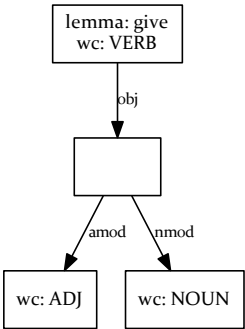
(a) Rank 1



(b) Rank 2



(c) Rank 3



(d) Rank 4

Figure A.1: Larger structures most strongly associated with monotransitive *give*

Bibliography

- Ágel, Vilmos, Ludwig M. Eichinger, Hans Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, Henning Lobin, eds. (2003): *Dependenz und Valenz. Ein internationales Handbuch der zeitgenössischen Forschung/Dependency and Valency. An International Handbook of Contemporary Research*. No. 29.1 in Handbücher zur Sprach- und Kommunikationswissenschaft. Berlin / New York, NY: de Gruyter.
- Ágel, Vilmos, Ludwig M. Eichinger, Hans Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, Henning Lobin, eds. (2006): *Dependenz und Valenz. Ein internationales Handbuch der zeitgenössischen Forschung/Dependency and Valency. An International Handbook of Contemporary Research*. No. 29.2 in Handbücher zur Sprach- und Kommunikationswissenschaft. Berlin / New York, NY: de Gruyter.
- Agresti, Alan (2013): *Categorical Data Analysis*, 3rd ed. Hoboken, NJ: Wiley.
- Agresti, Alan, David B. Hitchcock (2005): "Bayesian inference for categorical data analysis." *Statistical Methods and Applications* 14/3: 297–330. <https://doi.org/10.1007/s10260-005-0121-y>.
- Ambati, Bharat Ram, Siva Reddy, Adam Kilgarrieff (2012): "Word sketches for Turkish." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul: European Language Resources Association, 2945–2950. http://www.lrec-conf.org/proceedings/lrec2012/pdf/585_Paper.pdf.
- Anthony, Laurence (2014): "AntConc (version 3.4.3)." Computer Software. Tokyo, Japan: Waseda University. <http://www.laurenceanthony.net/software/antconc/>.
- Baayen, Harald R. (2001): *Word frequency distributions*. Dordrecht: Kluwer.
- Bahns, Jens (1996): *Kollokationen als lexikographisches Problem. Eine Analyse allgemeiner und spezieller Lernerwörterbücher des Englischen*. Tübingen: Niemeyer.

- Baroni, Marco, Stefan Evert (2009): "Statistical methods for corpus exploitation." In: Lüdeling and Kytö (2009), 777–803.
- Bartsch, Sabine (2004): *Structural and functional properties of collocations in English. A corpus study of lexical and pragmatic constraints on lexical co-occurrence*. Tübingen: Narr.
- Bartsch, Sabine, Stefan Evert (2014): "Towards a Firthian notion of collocation." *OPAL – Online publizierte Arbeiten zur Linguistik* 2/2014: 48–61. <http://pub.ids-mannheim.de/laufend/opal/pdf/opal2014-2.pdf>.
- Basili, Roberto, Maria Teresa Pazienza, Paola Velardi (1994): "A 'not-so-shallow' parser for collocational analysis." In: *Proceedings of the 15th Conference on Computational Linguistics (COLING'94)*. Tokyo: Association for Computational Linguistics, 447–453. <http://aclweb.org/anthology/C94-1074>.
- Bel Enguix, Gemma, Reinhard Rapp, Michael Zock (2014): "How well can a corpus-derived co-occurrence network simulate human associative behavior?" In: *Proceedings of the 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL'14)*. Gothenburg: Association for Computational Linguistics, 43–48. <http://aclweb.org/anthology/W14-0509>.
- Blaheta, Don, Mark Johnson (2001): "Unsupervised learning of multi-word verbs." In: *Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis and Exploitation*. Toulouse, 54–60. <http://web.science.mq.edu.au/~mjohnson/papers/2001/dpb-colloc01.pdf>.
- Bod, Rens (1992): "A computational model of language performance: Data oriented parsing." In: *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*. Nantes: Association for Computational Linguistics, 855–859. <http://aclweb.org/anthology/C92-3126>.
- Bod, Rens (1998): *Beyond Grammar. An Experience-Based Theory of Language*. No. 88 in CSLI Lecture Notes. Stanford, CA: CSLI Publications.
- Bod, Rens, Remko Scha, Khalil Sima'an, eds. (2003): *Data-Oriented Parsing*. Stanford, CA: CSLI Publications.
- Boslaugh, Sarah (2012): *Statistics in a Nutshell*, 2nd ed. Sebastopol, CA: O'Reilly.

- Bouma, Gosse, Gertjan van Noord, Robert Malouf (2001): "Alpino: Wide-coverage computational analysis of Dutch." In: *Computational Linguistics in The Netherlands 2000*. <http://www.let.rug.nl/~vannoord/papers/alpino.pdf>.
- Breidt, Elisabeth (1993): "Extraction of v-n-collocations from text corpora: A feasibility study for German." In: *Proceedings of the Workshop on Very Large Corpora: Academic and Industrial Perspectives*. Columbus, OH: Association for Computational Linguistics, 74–83. <http://aclweb.org/anthology/W93-0309>.
- Brent, Michael R. (1993): "From grammar to lexicon: Unsupervised learning of lexical syntax." *Computational Linguistics* 19/2: 243–262. <http://aclweb.org/anthology/J93-2002>.
- Bresnan, Joan, ed. (1982): *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.
- Briscoe, Ted, John Carroll (1997): "Automatic extraction of subcategorization from corpora." In: *Fifth Conference on Applied Natural Language Processing (ANLP'97)*. Washington, DC: Association of Computational Linguistics, 356–363. <http://aclweb.org/anthology/A97-1052>.
- British National Corpus (2007): "The British National Corpus." Distributed by Oxford University Computing Services on behalf of the BNC Consortium. Version 3 (BNC XML Edition), <http://www.natcorp.ox.ac.uk/>.
- Bunescu, Razvan, Raymond Mooney, Arun Ramani, Edward Marcotte (2006): "Integrating co-occurrence statistics with information extraction for robust retrieval of protein interactions from medline." In: *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*. New York, NY: Association for Computational Linguistics, 49–56. <http://aclweb.org/anthology/W06-3307>.
- Bybee, Joan (2010): *Language, Usage and Cognition*. Cambridge: Cambridge University Press.
- Carpenter, Bob (1992): *The Logic of Typed Feature Structures*. No. 32 in Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.
- Carroll, John, Guido Minnen, Ted Briscoe (1998): "Can subcategorisation probabilities help a statistical parser?" In: *Proceedings of the Sixth ACL/SIGDAT*

- Workshop on Very Large Corpora*. Montreal: Association for Computational Linguistics. <http://aclweb.org/anthology/W98-1114>.
- Carstensen, Kai-Uwe, Christian Ebert, Cornelia Ebert, Susanne Jekat, Ralf Klabunde, Hagen Langer, eds. (2010): *Computerlinguistik und Sprachtechnologie. Eine Einführung*, 3rd ed. Heidelberg: Spektrum.
- Charikar, Moses S. (2002): "Similarity estimation techniques from rounding algorithms." In: *Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing (STOC'02)*. New York, NY: ACM, 380–388. <https://doi.org/10.1145/509907.509965>.
- Charniak, Eugene, Don Blaheta, Niyu Ge, Keith Hall, John Hale, Mark Johnson (2000): "BLLIP 1987-89 WSJ Corpus Release 1." LDC corpus LDC2000T43. Philadelphia, PA: Linguistic Data Consortium, <https://catalog.ldc.upenn.edu/LDC2000T43>.
- Chaudhari, L. Dipak, P. Om Damani, Srivatsan Laxman (2011): "Lexical co-occurrence, statistical significance, and word association." In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. Edinburgh: Association for Computational Linguistics, 1058–1068. <http://aclweb.org/anthology/D11-1098>.
- Chen, Mei-hua, Chung-chi Huang, Shih-ting Huang, Jason S. Chang (2010): "GRASP: Grammar- and syntax-based pattern-finder for collocation and phrase learning." In: *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC 24)*. Sendai, 357–364. <http://aclweb.org/anthology/Y10-1040>.
- Cheng, Winnie, Chris Greaves, John Sinclair, Martin Warren (2009): "Uncovering the extent of the phraseological tendency: Towards a systematic analysis of congrams." *Applied Linguistics* 30/2: 236–252. <https://doi.org/10.1093/applin/amn039>.
- Cheng, Winnie, Chris Greaves, Martin Warren (2006): "From n-gram to skipgram to congram." *International Journal of Corpus Linguistics* 11/4: 411–433. <https://doi.org/10.1075/ijcl.11.4.04che>.
- Chesley, Paula, Susanne Salmon-Alt (2006): "Automatic extraction of subcategorization frames for French." In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa: European Language Resources Association, 253–258. http://www.lrec-conf.org/proceedings/lrec2006/pdf/101_pdf.pdf.

- Chomsky, Noam (2015): *The Minimalist Program. 20th Anniversary Edition*. Cambridge, MA: MIT Press.
- Choueka, Yaacov (1988): "Looking for needles in a haystack. Or: Locating interesting collocational expressions in large textual databases." In: *RIAO. Conference Proceedings 1988*. 609–623.
- Church, Kenneth, William Gale, Patrick Hanks, Donald Hindle (1989): "Parsing, word associations and typical predicate-argument relations." In: *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts, October 15-18, 1989*. Cape Cod, MA, 75–81. <http://aclweb.org/anthology/H89-2012>.
- Church, Kenneth W. (2000): "Empirical estimates of adaptation: The chance of two Noriegas is closer to $p/2$ than p^2 ." In: *Proceedings of the 18th Conference on Computational Linguistics (COLING'00)*, vol. 1. Stroudsburg, PA: Association for Computational Linguistics, vol. 1, 180–186. <http://aclweb.org/anthology/C00-1027>.
- Church, Kenneth Ward, Patrick Hanks (1990): "Word association norms, mutual information, and lexicography." *Computational Linguistics* 16/1: 22–29. <http://aclweb.org/anthology/J90-1003>.
- Clark, Stephen, James R. Curran (2004): "Parsing the WSJ using CCG and log-linear models." In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*. Barcelona: Association for Computational Linguistics, 104–111. <http://aclweb.org/anthology/P04-1014>.
- Cochran, Willian G. (1954): "Some methods for strengthening the common χ^2 tests." *Biometrics* 10/4: 417–451. <https://doi.org/10.2307/3001616>.
- Cohen, Jacob (1994): "The earth is round ($p < .05$)." *American Psychologist* 49/12: 997–1003. <https://doi.org/10.1037/0003-066X.49.12.997>.
- Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest (1990): *Introduction to Algorithms*. Cambridge, MA: MIT Press.
- Croft, William, D. Alan Cruse (2004): *Cognitive Linguistics*. Cambridge: Cambridge University Press.
- Dahl, Verónica (1989): "Discontinuous grammars." *Computational Intelligence* 5/3: 161–179. <https://doi.org/10.1111/j.1467-8640.1989.tb00326.x>.

- Daudaravičius, Vidas, Kaunas Rūta Marcinkevičienė (2004): "Gravity counts for the boundaries of collocations." *International Journal of Corpus Linguistics* 9/2: 321–348. <https://doi.org/10.1075/ijcl.9.2.o8dau>.
- Davies, Mark (2008–): "The Corpus of Contemporary American English (COCA): 520 million words, 1990–present." Available online at <http://corpus.byu.edu/coca/>. <http://corpus.byu.edu/coca/>.
- de Marneffe, Marie-Catherine, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, Christopher D. Manning (2014): "Universal Stanford Dependencies: A cross-linguistic typology." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik: European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062_Paper.pdf.
- de Marneffe, Marie-Catherine, Bill MacCartney, Christopher D. Manning (2006): "Generating typed dependency parses from phrase structure parses." In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa: European Language Resources Association, 449–454. http://www.lrec-conf.org/proceedings/lrec2006/pdf/440_.pdf.
- de Marneffe, Marie-Catherine, Christopher D. Manning (2008): "The Stanford typed dependencies representation." In: *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*. Manchester: Association for Computational Linguistics, 1–8. <http://aclweb.org/anthology/W08-1301>.
- de Marneffe, Marie-Catherine, Christopher D. Manning (2015): *Stanford typed dependencies manual*. Stanford University. http://nlp.stanford.edu/software/dependencies_manual.pdf.
- Dennis, Sally F. (1965): "The construction of a thesaurus automatically from a sample of text." In: Stevens et al. (1965), 61–148. <https://archive.org/details/statisticalassoc269stev>.
- Dias, Gaël (2003): "Multiword unit hybrid extraction." In: *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. Sapporo: Association for Computational Linguistics, 41–48. <http://aclweb.org/anthology/W03-1806>.

- Diestel, Reinhard (2010): *Graph Theory*, 4th ed. Heidelberg: Springer.
- Dobson, Annette J., Adrian Barnett (2008): *An Introduction to Generalized Linear Models*, 3rd ed. Boca Raton, FL: Chapman and Hall/CRC.
- Dodge, Yadolah (2008): *The Concise Encyclopedia of Statistics*. New York, NY: Springer.
- Dunning, Ted (1993): "Accurate methods for the statistics of surprise and coincidence." *Computational Linguistics* 19/1: 61–74. <http://aclweb.org/anthology/J93-1003>.
- Ersan, Murat, Eugene Charniak (1996): "A statistical syntactic disambiguation program and what it learns." In: Wermter, Stefan, Ellen Riloff, Gabriele Scheler (eds.) *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, Berlin / Heidelberg: Springer, 146–159. https://doi.org/10.1007/3-540-60925-3_44.
- Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu (1996): "A density-based algorithm for discovering clusters in large spatial databases with noise." In: Simoudis, Evangelos, Jiawei Han, Usama M. Fayyad (eds.) *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 226–231. <http://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf>.
- Everitt, Brian S. (1992): *The Analysis of Contingency Tables*. No. 45 in Monographs on Statistics and Applied Probability. London / New York, NY / Tokyo / Melbourne / Madras: Chapman & Hall/CRC.
- Evert, Stefan (2004): *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart. Published in 2005, <http://elib.uni-stuttgart.de/opus/volltexte/2005/2371/>.
- Evert, Stefan (2006): "How random is a corpus? The library metaphor." *Zeitschrift für Anglistik und Amerikanistik* 54/2: 177–190. <https://doi.org/10.1515/zaa-2006-0208>.
- Evert, Stefan (2009): "Corpora and collocations." In: Lüdeling and Kytö (2009), 1212–1248.
- Evert, Stefan, Brigitte Krenn (2001): "Methods for the qualitative evaluation of lexical association measures." In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01)*.

- Toulouse: Association for Computational Linguistics, 188–195. <http://www.aclweb.org/anthology/P01-1025>.
- Evert, Stefan, Peter Uhrig, Sabine Bartsch, Thomas Proisl (2017): “E-VIEW-alation – a large-scale evaluation study of association measures for collocation identification.” In: Kosem, Iztok, Carole Tiberius, Miloš Jakubiček, Jelena Kallas, Simon Krek, Vít Baisa (eds.) *Electronic lexicography in the 21st century. Proceedings of the eLex 2017 conference*. Leiden: Lexical Computing, 531–549. <https://elex.link/elex2017/wp-content/uploads/2017/09/paper32.pdf>.
- Farahmand, Meghdad, James Henderson (2016): “Modeling the non-substitutability of multiword expressions with distributional semantics and a log-linear model.” In: *Proceedings of the 12th Workshop on Multiword Expressions*. Berlin: Association for Computational Linguistics, 61–66. <https://aclweb.org/anthology/W16-1809>.
- Fillmore, Charles J., Paul Kay, Mary Catherine O'Connor (1988): “Regularity and idiomaticity in grammatical constructions: The case of let alone.” *Language* 64/3: 501–538. <https://doi.org/10.2307/414531>.
- Firth, John Rupert (1957): “A synopsis of linguistic theory, 1930–55.” In: *Studies in Linguistic Analysis*, Oxford: The Philological Society, 1–32.
- Forsberg, Markus, Richard Johansson, Linnéa Bäckström, Lars Borin, Benjamin Lyngfelt, Joel Olofsson, Julia Prentice (2014): “From construction candidates to constructicon entries: An experiment using semi-automatic methods for identifying constructions in corpora.” *Constructions and Frames* 6/1: 114–135. <https://doi.org/10.1075/cf.6.1.o7for>.
- Freeman, Linton C. (1978): “Centrality in social networks. Conceptual clarification.” *Social Networks* 1/3: 215–239. [https://doi.org/10.1016/0378-8733\(78\)90021-7](https://doi.org/10.1016/0378-8733(78)90021-7).
- Gallier, Jean (2011): *Discrete Mathematics*. New York, NY / Dordrecht / Heidelberg / London: Springer.
- Garey, Michael R., David S. Johnson (1979): *Computers and Intractability. A Guide to the Theory of NP-Completeness*. New York, NY: W. H. Freeman.
- Gazdar, Gerald, Ewan H. Klein, Geoffrey K. Pullum, Ivan A. Sag (1985): *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

- Gerdes, Kim, Eva Hajičová, Leo Wanner, eds. (2014): *Dependency Linguistics. Recent Advances in Linguistic Theory Using Dependency Structures*. Amsterdam / Philadelphia, PA: John Benjamins.
- Giuliano, Vincent E. (1965): "The interpretation of word associations." In: Stevens et al. (1965), 25–32. <https://archive.org/details/statisticalassoc269stev>.
- Goldberg, Adele E. (1995): *Constructions. A Construction Grammar Approach to Argument Structure*. Chicago, IL/London: University of Chicago Press.
- Goldberg, Adele E. (2006): *Constructions at Work. The Nature of Generalization in Language*. Oxford: Oxford University Press.
- Grefenstette, Gregory (1994): *Explorations in Automatic Thesaurus Discovery*. Boston, MA: Kluwer Academic Publishers.
- Gries, Stefan Th. (2005): "Null-hypothesis significance testing of word frequencies: a follow-up on Kilgarriff." *Corpus Linguistics and Linguistic Theory* 1/2: 277–294. <https://doi.org/10.1515/cllt.2005.1.2.277>.
- Gries, Stefan Th. (2008): "Dispersions and adjusted frequencies in corpora." *International Journal of Corpus Linguistics* 13/4: 403–437. <https://doi.org/10.1075/ijcl.13.4.02gri>.
- Gries, Stefan Th. (2012): "Frequencies, probabilities, and association measures in usage-/exemplar-based linguistics: Some necessary clarifications." *Studies in Language* 36/3: 477–510. <https://doi.org/10.1075/sl.36.3.02gri>.
- Gries, Stefan Th. (2013): "50-something years of work on collocations. What is or should be next" *International Journal of Corpus Linguistics* 18/1: 137–165. <https://doi.org/10.1075/ijcl.18.1.09gri>.
- Gries, Stefan Th. (2015): "More (old and new) misunderstandings of collocation analysis: On Schmid and Küchenhoff (2013)." *Cognitive Linguistics* 26/3: 505–536. <https://doi.org/10.1515/cog-2014-0092>.
- Gries, Stefan Th., Anatol Stefanowitsch (2004a): "Covarying collexemes in the Into-causative." In: Achard, Michel, Suzanne Kemmer (eds.) *Language, Culture, and Mind*, Stanford, CA: CSLI, 225–236.
- Gries, Stefan Th., Anatol Stefanowitsch (2004b): "Extending collocation analysis: A corpus-based perspective on 'alternations'." *International Journal of Corpus Linguistics* 9/1: 97–129. <https://doi.org/10.1075/ijcl.9.1.06gri>.

- Groß, Thomas, Timothy Osborne (2009): "Toward a practical dependency grammar theory of discontinuities." *SKY Journal of Linguistics* 22: 43–90. http://www.linguistics.fi/julkaisut/SKY2009/Gross_Osborne_NETTI.pdf.
- Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie, Yorick Wilks (2006): "A closer look at skip-gram modelling." In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa: European Language Resources Association, 1222–1225. http://www.lrec-conf.org/proceedings/lrec2006/pdf/357_pdf.pdf.
- Hagberg, Aric A., Daniel A. Schult, Pieter J. Swart (2008): "Exploring network structure, dynamics, and function using NetworkX." In: *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA, 11–15. https://conference.scipy.org/proceedings/scipy2008/paper_2/full_text.pdf.
- Halliday, Michael A. K., Ruqaiya Hasan (1976): *Cohesion in English*. London: Longman.
- Han, Xiwu, Tiejun Zhao, Haoliang Qi, Hao Yu (2004): "Subcategorization acquisition and evaluation for Chinese verbs." In: *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*. Geneva: Association for Computational Linguistics. <http://aclweb.org/anthology/C04-1104>.
- Hanks, Patrick (2010): "Nine issues in metaphor theory and analysis." *International Journal of Corpus Linguistics* 15/1: 133–150. <https://doi.org/10.1075/ijcl.15.1.o6han>.
- Hardie, Andrew (2012): "CQPweb – combining power, flexibility and usability in a corpus analysis tool." *International Journal of Corpus Linguistics* 17/33: 380–409. <https://doi.org/10.1075/ijcl.17.3.o4har>.
- Hasan, Ruqaiya (1984): "Coherence and cohesive harmony." In: Flood, James (ed.) *Understanding Reading Comprehension. Cognition, Language and the Structure of Prose*, Newark: International Reading Association, 181–219.
- Hausmann, Franz Joseph (1984): "Wortschatzlernen ist Kollokationslernen. Zum Lehren und Lernen französischer Wortverbindungen." *Praxis des neusprachlichen Unterrichts* 84/4: 395–406.
- Heid, Ulrich (1994): "On ways words work together – topics in lexical combinatorics." In: Martin, Willy, Willem Meijs, Margreet Moerland, Elsemiek ten

- Pas, Piet van Sterkenburg, Piek Vossen (eds.) *Euralex 1994 Proceedings*. Amsterdam, 226–257. http://www.euralex.org/elx_proceedings/Euralex1994/27_Euralex_Ulrich%20Heid%20-%20On%20Ways%20Words%20Work%20Together%20-%20Topics%20in%20Lexical%20Combinatorics.pdf.
- Heid, Ulrich (1998): “Towards a corpus-based dictionary of German noun-verb collocations.” In: Fontenelle, Thierry, Philippe Hiligsmann, Archibald Michiels, André Moulin, Siegfried Theissen (eds.) *Proceedings of the 8th EURALEX International Congress*. Liège: Euralex, 301–312. <http://euralex.org/wp-content/themes/euralex/proceedings/Euralex%201998%20Part%201/Ulrich%20HEID%20Towards%20a%20corpus-based%20dictionary%20of%20German%20noun-verb%20collocations.pdf>.
- Heid, Ulrich, Fabienne Fritzinger, Susanne Hauptmann, Julia Weidenkaff, Marion Weller (2008): “Providing corpus data for a dictionary for German juridical phraseology.” In: Storrer, Angelika, Alexander Geyken, Alexander Siebert, Kay-Michael Würzner (eds.) *Text Resources and Lexical Knowledge. Selected Papers from the 9th Conference on Natural Language Processing, KONVENS 2008, Berlin, Germany*. Berlin / Boston, MA: Mouton de Gruyter, 131–144. <https://doi.org/10.1515/9783110211818.2.131>.
- Heidorn, George E. (2000): “Intelligent writing assistance.” In: Dale, Robert, Hermann Moisl, Harold Somers (eds.) *Handbook of Natural Language Processing*, New York, NY / Basel: Marcel Dekker, 181–207.
- Herbst, Thomas (1996): “What are collocations: Sandy beaches or false teeth?” *English Studies* 77/4: 379–393. <https://doi.org/10.1080/00138389608599038>.
- Herbst, Thomas (2007): “Valency complements or valency patterns?” In: Herbst, Thomas, Kathrin Götz-Votteler (eds.) *Valency. Theoretical, Descriptive and Cognitive Issues*, Berlin / New York, NY: Mouton de Gruyter, 15–35.
- Herbst, Thomas (forthcoming 2019): “Dependency and valency in the description of English grammar.” In: Aarts, Bas (ed.) *The Oxford Handbook of English Grammar*, Oxford: Oxford University Press.
- Herbst, Thomas, David Heath, Ian F. Roe, Dieter Götz (2004): *A Valency Dictionary of English*. No. 40 in Topics in English Linguistics (TiEL). Berlin / New York, NY: Mouton de Gruyter.
- Herbst, Thomas, Susen Schüller (2008): *Introduction to Syntactic Analysis. A Valency Approach*. Tübingen: Narr.

- Hindle, Donald (1983): *User manual for Fidditch, a deterministic parser*. Naval Research Laboratory. Technical Memorandum 7590-142.
- Hoey, Michael (2004): "Textual colligation: a special kind of lexical priming." In: Aijmer, Karin, Bengt Altenberg (eds.) *Advances in corpus linguistics. Papers from the 23rd International Conference on English Language Research on Computerized Corpora (ICAME 23)*. Göteborg 22–26 May 2002, Amsterdam / New York, NY: Rodopi, no. 49 in Language and Computers – Studies in Practical Linguistics, 171–194.
- Hoffmann, Sebastian, Stefan Evert (2006): "BNCweb (CQP-edition): The marriage of two corpus tools." In: Braun, Sabine, Kurt Kohn, Joybrato Mukherjee (eds.) *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, Frankfurt am Main: Peter Lang, 177–195.
- Hoffmann, Sebastian, Stefan Evert, Nicholas Smith, David Lee, Ylva Berglund Prytz (2008): *Corpus Linguistics with BNCweb – a Practical Guide*. Frankfurt am Main: Peter Lang.
- Huddleston, Rodney, Geoffrey K. Pullum (2002): *The Cambridge Grammar of the English Language*. Cambridge: Cambridge University Press.
- Hudson, Richard (2010): *An Introduction to Word Grammar*. Cambridge: Cambridge University Press.
- Hunston, Susan, Gill Francis (2000): *Pattern Grammar. A corpus-driven approach to the lexical grammar of English*. No. 4 in Studies in Corpus Linguistics (SCL). Amsterdam / Philadelphia, PA: John Benjamins.
- Ienco, Dino, Serena Villata, Cristina Bosco (2008): "Automatic extraction of subcategorization frames for Italian." In: Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Daniel Tapias (eds.) *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'o8)*. Marrakech: European Language Resources Association, 2094–2100. http://www.lrec-conf.org/proceedings/lrec2008/pdf/536_paper.pdf.
- Ivanova, Kremena, Ulrich Heid, Sabine Schulte im Walde, Adam Kilgarriff, Jan Pomikalek (2008): "Evaluating a German sketch grammar: A case study on noun phrase case." In: Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Daniel Tapias (eds.) *Proceedings of the Sixth International Conference on Language Resources*

- and Evaluation (LREC'08). Marrakech: European Language Resources Association, 2101–2107. http://www.lrec-conf.org/proceedings/lrec2008/pdf/537_paper.pdf.
- Jones, Susan, John Sinclair (1974): “English lexical collocations.” *Cahiers de lexicologie* 24: 15–61.
- Jurafsky, Daniel, James H. Martin (2009): *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Upper Saddle River, NJ: Pearson.
- Justeson, John S., Slava M. Katz (1991): “Co-occurrences of antonymous adjectives and their contexts.” *Computational Linguistics* 17/1: 1–19. <http://aclweb.org/anthology/J91-1001>.
- Justeson, John S., Slava M. Katz (1995): “Technical terminology: some linguistic properties and an algorithm for identification in text.” *Natural Language Engineering* 1/1: 9–27. <https://doi.org/10.1017/S1351324900000048>.
- Katz, Graham, Eugenie Giesbrecht (2006): “Automatic identification of non-compositional multi-word expressions using latent semantic analysis.” In: *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties (MWE'06)*. Sydney: Association for Computational Linguistics, 12–19. <http://aclweb.org/anthology/W06-1203>.
- Katz, Slava M. (1996): “Distribution of content words and phrases in text and language modelling.” *Natural Language Engineering* 2/1: 15–59.
- Kermes, Hannah (2003): *Off-line (and On-line) Text Analysis for Computational Lexicography*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS) 9/3, http://www.ims.uni-stuttgart.de/~kermes/diss/03_diss_Kermes.pdf.
- Kermes, Hannah, Ulrich Heid (2003): “Using chunked corpora for the acquisition of collocations and idiomatic expressions.” In: Kiefer, F., J. Pajzs (eds.) *Proceedings of 7th Conference on Computational Lexicography and Corpus Research*. Budapest: Research Institute for Linguistics, Hungarian Academy of Sciences.
- Kiela, Douwe, Stephen Clark (2013): “Detecting compositionality of multi-word expressions using nearest neighbours in vector space models.” In:

- Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. Seattle, WA: Association for Computational Linguistics, 1427–1432. <http://www.aclweb.org/anthology/D13-1147>.
- Kilgarrieff, Adam (2005): “Language is never, ever, ever, random.” *Corpus Linguistics and Linguistic Theory* 1/2: 263–276. <https://doi.org/10.1515/cllt.2005.1.2.263>.
- Kilgarrieff, Adam, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý, Vít Suchomel (2014a): “The Sketch Engine: ten years on.” *Lexicography* 1/1: 7–36. <https://doi.org/10.1007/s40607-014-0009-9>.
- Kilgarrieff, Adam, Pavel Rychlý, Milos Jakubicek, Vojtěch Kovář, Vít Baisa, Lucia Kocincová (2014b): “Extrinsic corpus evaluation with a collocation dictionary task.” In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik: European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2014/pdf/52_Paper.pdf.
- Kilgarrieff, Adam, Pavel Rychlý, Pavel Smrz, David Tugwell (2004): “The Sketch Engine.” In: Williams, Geoffrey, Sandra Vessier (eds.) *Proceedings of the 11th EURALEX International Congress*. Lorient: Université de Bretagne-Sud, Faculté des lettres et des sciences humaines, 105–115. http://euralex.org/wp-content/themes/euralex/proceedings/Euralex%202004/011_2004_V1_Adam%20KILGARRIFF,%20Pavel%20RYCHLY,%20Pavel%20SMRZ,%20David%20TUGWELL_The%20%20Sketch%20Engine.pdf.
- Kilgarrieff, Adam, David Tugwell (2002): “Sketching words.” In: Corréard, Marie-Hélène (ed.) *Lexicography and Natural Language Processing: A Festschrift in Honour of B. T. S. Atkins*, EURALEX, 125–137.
- Kjellmer, Göran (1994): *A Dictionary of English Collocations*. Oxford: Clarendon.
- Klein, Dan, Christopher D. Manning (2003): “Accurate unlexicalized parsing.” In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*. Sapporo: Association for Computational Linguistics, 423–430. <http://aclweb.org/anthology/P03-1054>.

- Korhonen, Anna (2002): *Subcategorization acquisition*. Ph.D. thesis, Computer Laboratory, University of Cambridge. <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-530.pdf>.
- Korhonen, Anna, Yuval Krymolowski, Ted Briscoe (2006): "A large subcategorization lexicon for natural language processing applications." In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa: European Language Resources Association, 1015–1020. http://lrec-conf.org/proceedings/lrec2006/pdf/558_pdf.pdf.
- Küchenhoff, Helmut, Hans-Jörg Schmid (2015): "Reply to "More (old and new) misunderstandings of collostructional analysis: On Schmid & Küchenhoff" by Stefan Th. Gries." *Cognitive Linguistics* 26/3: 537–547. <https://doi.org/10.1515/cog-2015-0053>.
- Lapata, Maria (1999): "Acquiring lexical generalizations from corpora: A case study for diathesis alternations." In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*. College Park, MD: Association for Computational Linguistics, 397–404. <http://aclweb.org/anthology/P99-1051>.
- Lea, Diana (2007): "Making a collocations dictionary." *Zeitschrift für Anglistik und Amerikanistik* 55/3: 261–271. <https://doi.org/10.1515/zaa.2007.55.3.261>.
- Lehr, Andrea (1996): *Kollokationen und maschinenlesbare Korpora*. Tübingen: Niemeyer.
- Lenci, Alessandro, Gabriella Lapesa, Giulia Bonansinga (2012): "LexIt: A computational resource on Italian argument structure." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul: European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2012/pdf/622_Paper.pdf.
- Leydesdorff, Loet, Liwen Vaughan (2006): "Co-occurrence matrices and their applications in information science: Extending ACA to the web environment." *Journal of the American Society for Information Science and Technology* 57/12: 1616–1628. <https://doi.org/10.1002/asi.20335>.
- Li, Jianguo, Chris Brew, Eric Fosler-Lussier (2005): "Robust extraction of subcategorization data from spoken language." In: *Proceedings of the*

- Ninth International Workshop on Parsing Technology (IWPT)*. Vancouver: Association for Computational Linguistics, 204–205. <http://aclweb.org/anthology/W05-1529>.
- Liddell, Douglas (1976): “Practical tests of 2×2 contingency tables.” *Journal of the Royal Statistical Society. Series D (The Statistician)* 25/4: 295–304. <https://doi.org/10.2307/2988087>.
- Lienert, Gustav A. (1969): “Die ‘Konfigurationsfrequenzanalyse’ als Klassifikationsmethode in der klinischen Psychologie.” In: Irle, Martin (ed.) *Bericht über den 26. Kongreß der Deutschen Gesellschaft für Psychologie in Tübingen 1968*. Göttingen: Hogrefe, 244–253.
- Lijffijt, Jefrey, Terttu Nevalainen, Tanja Säily, Panagiotis Papapetrou, Kai Puolamäki, Heikki Mannila (2016): “Significance testing of word frequencies in corpora.” *Digital Scholarship in the Humanities* 31/2: 374–397. <https://doi.org/10.1093/llc/fquo64>.
- Lin, Dekang (1994): “PRINCIPAR – an efficient, broad-coverage, principle-based parser.” In: *Proceedings of the 15th International Conference on Computational Linguistics (COLING’94)*. 482–488. <http://aclweb.org/anthology/C94-1079>.
- Lin, Dekang (1998): “Extracting collocations from text corpora.” In: *Proceedings of the First Workshop on Computational Terminology*. Montreal, 57–63.
- Lin, Dekang (1999): “Automatic identification of non-compositional phrases.” In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL’99)*. Morristown, NJ: Association for Computational Linguistics, 317–324. <http://aclweb.org/anthology/P99-1041>.
- Loftus, Geoffrey R. (1991): “On the tyranny of hypothesis testing in the social sciences.” *Contemporary Psychology* 36/2: 102–105. <https://doi.org/10.1037/029395>.
- Loftus, Geoffrey R. (1996): “Psychology will be a much better science when we change the way we analyze data.” *Current Directions in Psychological Science* 5/6: 161–171. <https://doi.org/10.1111/1467-8721.ep11512376>.
- Lyons, John (1977): *Semantics*, vol. 2. Cambridge: Cambridge University Press.

- Lü, Yajuan, Ming Zhou (2004): "Collocation translation acquisition using monolingual corpora." In: *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*. Barcelona: Association for Computational Linguistics, 167–174. <http://aclweb.org/anthology/P04-1022>.
- Lüdeling, Anke, Merja Kytö, eds. (2009): *Corpus Linguistics: An International Handbook*. No. 29.2 in Handbücher zur Sprach- und Kommunikationswissenschaft. Berlin / New York, NY: Walter de Gruyter.
- Manning, Christopher D. (1993): "Automatic acquisition of a large subcategorization dictionary from corpora." In: *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL'93)*. Columbus, OH: Association for Computational Linguistics, 235–242. <http://aclweb.org/anthology/P93-1032>.
- Maron, Melvin Earl, John Lary Kuhns (1960): "On relevance, probabilistic indexing and information retrieval." *Journal of the ACM* 7/3: 216–244. <https://doi.org/10.1145/321033.321035>.
- Martens, Scott, Vincent Vandeghinste (2010): "An efficient, generic approach to extracting multi-word expressions from dependency trees." In: *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications*. Beijing: Coling 2010 Organizing Committee, 85–88. <http://aclweb.org/anthology/W10-3713>.
- McIntosh, Colin, Ben Francis, Richard Poole, eds. (2009): *Oxford Collocations Dictionary for students of English*, 2nd ed. Oxford University Press.
- Messiant, Cédric, Thierry Poibeau, Anna Korhonen (2008): "LexSchem: a large subcategorization lexicon for French verbs." In: Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Daniel Tapias (eds.) *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech: European Language Resources Association, 533–538. http://www.lrec-conf.org/proceedings/lrec2008/pdf/142_paper.pdf.
- Michelbacher, Lukas, Stefan Evert, Hinrich Schütze (2007): "Asymmetric association measures." In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007)*. Borovets. <http://lml.bas.bg/ranlp2007/DOCS/RANLP2007.pdf>.

- Michelbacher, Lukas, Stefan Evert, Hinrich Schütze (2011): "Asymmetry in corpus-derived and human word associations." *Corpus Linguistics and Linguistic Theory* 7/2: 245–276. <https://doi.org/10.1515/cllt.2011.012>.
- Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, Katherine J. Miller (1990): "Introduction to WordNet: An on-line lexical database." *International Journal of Lexicography* 3/4: 235–244. <https://doi.org/10.1093/ijl/3.4.235>.
- Mitchell, Terence Frederick (1975): *Principles of Firthian Linguistics*. London: Longman.
- Nastase, Vivi, Rada Mihalcea, Dragomir R. Radev (2015): "A survey of graphs in natural language processing." *Natural Language Engineering* 21/5: 665–698. <https://doi.org/10.1017/S1351324915000340>.
- Nerima, Luka, Violeta Seretan, Eric Wehrli (2003): "Creating a multilingual collocations dictionary from large text corpora." In: *Companion Volume to the Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*. Budapest: Association for Computational Linguistics, 131–134. <http://aclweb.org/anthology/E03-1022>.
- Nissim, Malvina, Andrea Zaninello (2013): "Modeling the internal variability of multiword expressions through a pattern-based method." *ACM Transactions on Speech and Language Processing (TSLP)* 10/2: 7:1–7:26. <https://doi.org/10.1145/2483691.2483696>.
- Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, Daniel Zeman (2016): "Universal Dependencies v1: A multilingual treebank collection." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož: European Language Resources Association, 1659–1666. http://www.lrec-conf.org/proceedings/lrec2016/pdf/348_Paper.pdf.
- Nivre, Joakim, Johan Hall, Jens Nilsson (2006): "MaltParser: A data-driven parser-generator for dependency parsing." In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa: European Language Resources Association, 2216–2219. http://www.lrec-conf.org/proceedings/lrec2006/pdf/162_pdf.pdf.

- Pearce, Darren (2001): "Synonymy in collocation extraction." In: *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*. Pittsburgh, PA: Association for Computational Linguistics, 41–46.
- Pearce, Darren (2002): "A comparative evaluation of collocation extraction techniques." In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*. Las Palmas: European Language Resources Association, 1530–1536. <http://www.lrec-conf.org/proceedings/lrec2002/pdf/169.pdf>.
- Pecina, Pavel (2005): "An extensive empirical study of collocation extraction methods." In: *Proceedings of the ACL Student Research Workshop*. Ann Arbor, MI: Association for Computational Linguistics, 13–18. <http://aclweb.org/anthology/P05-2003>.
- Pecina, Pavel (2010): "Lexical association measures and collocation extraction." *Language Resources and Evaluation* 44/1: 137–158. <https://doi.org/10.1007/s10579-009-9101-4>.
- Pecina, Pavel, Pavel Schlesinger (2006): "Combining association measures for collocation extraction." In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney: Association for Computational Linguistics, 651–658. <http://aclweb.org/anthology/P06-2084>.
- Pedersen, Ted (1996): "Fishing for exactness." In: *Proceedings of the South-Central SAS Users Group Conference*. Austin, TX, 188–200.
- Pedersen, Ted, Satandeep Banerjee, Bridget McInnes, Saiyam Kohli, Mahesh Joshi, Ying Liu (2011): "The ngram statistics package (Text::NSP) – a flexible tool for identifying ngrams, collocations, and word associations." In: *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011)*. Portland, OR: Association for Computational Linguistics, 131–133. <http://aclweb.org/anthology/W11-0821>.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay (2011): "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12: 2825–2830. <http://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>.

- Petrov, Slav, Dipanjan Das, Ryan McDonald (2012): "A universal part-of-speech tagset." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul: European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf.
- Pollard, Carl, Ivan A. Sag (1994): *Head-Driven Phrase Structure Grammar*. Chicago, IL: University of Chicago Press.
- Preiss, Judita, Ted Briscoe, Anna Korhonen (2007): "A system for large-scale acquisition of verbal, nominal and adjectival subcategorization frames from corpora." In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*. Prague: Association for Computational Linguistics, 912–919. <http://aclweb.org/anthology/P07-1115>.
- Proisl, Thomas (2012): "Automatically exploring lexical tendencies in English." In: Mukherjee, Joybrato, Magnus Huber (eds.) *Corpus Linguistics and Variation in English: Theory and Description*, no. 75 in Language and Computers – Studies in Practical Linguistics, Amsterdam / New York, NY: Rodopi, 143–154.
- Proisl, Thomas, Peter Uhrig (2012): "Efficient dependency graph matching with the IMS Open Corpus Workbench." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul: European Language Resources Association, 2750–2756. http://www.lrec-conf.org/proceedings/lrec2012/pdf/709_Paper.pdf.
- Ramisch, Carlos (2015): *Multiword Expressions Acquisition. A Generic and Open Framework*. Cham / Heidelberg / New York, NY / Dordrecht / London: Springer.
- Rayson, Paul, Damon Berridge, Brian Francis (2004): "Extending the Cochran rule for the comparison of word frequencies between corpora." In: Purnelle, Gérald, Cédric Fairon, Anne Dister (eds.) *Le poids des mots: Proceedings of the 7th International Conference on Statistical analysis of textual data (JADT 2004)*, vol. 2. Louvain-la-Neuve: Presses universitaires de Louvain, vol. 2, 926–936.

- Rodríguez-Fernández, Sara, Luis Espinosa Anke, Roberto Carlini, Leo Wanner (2016): "Semantics-driven recognition of collocations using word embeddings." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin: Association for Computational Linguistics, 499–505. <https://doi.org/10.18653/v1/P16-2081>.
- Salton, Gerard (1965): "An evaluation program for associative indexing." In: Stevens et al. (1965), 201–210. <https://archive.org/details/statisticalassoc269stev>.
- Sangati, Federico, Andreas van Cranenburgh (2015): "Multiword expression identification with recurring tree fragments and association measures." In: *Proceedings of the 11th Workshop on Multiword Expressions*. Denver, CO: Association for Computational Linguistics, 10–18. <http://aclweb.org/anthology/W15-0902>.
- Sarkar, Anoop, Daniel Zeman (2000): "Automatic extraction of subcategorization frames for Czech." In: *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*. Saarbrücken: Association for Computational Linguistics. <http://aclweb.org/anthology/C00-2100>.
- Schiehlen, Michael (2003): "Combining deep and shallow approaches in parsing german." In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*. Sapporo: Association for Computational Linguistics, 112–119. <http://aclweb.org/anthology/P03-1015>.
- Schmid, Hans-Jörg, Helmut Küchenhoff (2013): "Collostructional analysis and other ways of measuring lexicogrammatical attraction: Theoretical premises, practical problems and cognitive underpinnings." *Cognitive Linguistics* 24/3: 531–577. <https://doi.org/10.1515/cog-2013-0018>.
- Schmid, Helmut (2000): "LoPar: Design and implementation." Arbeitspapiere des Sonderforschungsbereiches 340, no. 149. Stuttgart: Institut für maschinelle Sprachverarbeitung. <http://www.cis.uni-muenchen.de/~schmid/papers/lopar.pdf>.
- Schulte im Walde, Sabine (2003): "A collocation database for German verbs and nouns." In: *Proceedings of the 7th Conference on Computational Lexicography and Text Research (COMPLEX'03)*. Budapest, 73–81. <http://www.ims.uni-stuttgart.de/institut/mitarbeiter/schulte/publications/workshop/complex-03.pdf>.

- Schulte im Walde, Sabine (2003): *Experiments on the Automatic Induction of German Semantic Verb Classes*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS) 9/2, <http://www.ims.uni-stuttgart.de/institut/mitarbeiter/schulte/theses/phd/phd-thesis.pdf>.
- Schuster, Sebastian, Christopher D. Manning (2016): "Enhanced English universal dependencies: An improved representation for natural language understanding tasks." In: Calzolari, Nicoletta, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis (eds.) *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož: European Language Resources Association, 2371–2378. http://www.lrec-conf.org/proceedings/lrec2016/pdf/779_Paper.pdf.
- Seretan, Violeta (2008): *Collocation Extraction Based on Syntactic Parsing*. Ph.D. thesis, Faculté des lettres, Université de Genève. <http://www.issco.unige.ch/en/staff/seretan/publ/PhDThesis-VioletaSeretan.pdf>.
- Seretan, Violeta, Luka Nerima, Eric Wehrli (2003): "Extraction of multi-word collocations using syntactic bigram composition." In: *Proceedings of the Fourth International Conference on Recent Advances in NLP (RANLP-2003)*. 424–431. <https://archive-ouverte.unige.ch/unige:17034>.
- Seretan, Violeta, Luka Nerima, Eric Wehrli (2004): "Multi-word collocation extraction by syntactic composition of collocation bigrams." In: Nicolov, Nicolas, Kalina Bontcheva, Galia Angelova, Ruslan Mitkov (eds.) *Recent Advances in Natural Language Processing III. Selected Papers from RANLP 2003*, no. 260 in Current Issues in Linguistic Theory, Amsterdam / Philadelphia, PA: John Benjamins, 91–100. <https://doi.org/10.1075/cilt.260.10ser>.
- Seretan, Violeta, Eric Wehrli (2006): "Accurate collocation extraction using a multilingual parser." In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney: Association for Computational Linguistics, 953–960. <http://aclweb.org/anthology/P06-1120>.
- Shieber, Stuart M. (1986): *An Introduction to Unification-based Approaches to Grammar*. No. 4 in CSLI Lecture Notes. Stanford, CA: CSLI Publications.
- Sinclair, John (1991): *Corpus, Concordance, Collocation*. Oxford: Oxford University Press.

- Sinclair, John (2004): *Trust the Text. Language, corpus and discourse*. London / New York, NY: Routledge. Ed. with Ronald Carter.
- Smadja, Frank A. (1993): "Retrieving collocations from text: Xtract." *Computational Linguistics* 19/1: 143–177. <http://aclweb.org/anthology/J93-1007>.
- Squillante, Luigi (2014): "Towards an empirical subcategorization of multiword expressions." In: *Proceedings of the 10th Workshop on Multiword Expressions (MWE 2014)*. Gothenburg: Association for Computational Linguistics, 77–81. <http://www.aclweb.org/anthology/W14-0813>.
- Stefanowitsch, Anatol, Stefan Th. Gries (2003): "Collostructions: Investigating the interaction of words and constructions." *International Journal of Corpus Linguistics* 8/2: 209–243. <https://doi.org/10.1075/ijcl.8.2.03ste>.
- Stefanowitsch, Anatol, Stefan Th. Gries (2005): "Covarying collexemes." *Corpus Linguistics and Linguistic Theory* 1/1: 1–43. <https://doi.org/10.1515/cllt.2005.1.1.1>.
- Stefanowitsch, Anatol, Stefan Th. Gries (2009): "Corpora and grammar." In: Lüdeling and Kytö (2009), 933–952.
- Stevens, Mary Elizabeth, Vincent E. Giuliano, Laurence B. Heilprin, eds. (1965): *Statistical Association Methods For Mechanized Documentation. Symposium Proceedings. Washington 1964*, no. 269 in National Bureau of Standards Miscellaneous Publication. National Bureau of Standards (U.S.). <https://archive.org/details/statisticalassoc269stev>.
- Stubbs, Michael (2009): "Technology and phraseology: With notes on the history of corpus linguistics." In: Römer, Ute, Rainer Schulze (eds.) *Exploring the Lexis-Grammar Interface*, no. 35 in Studies in Corpus Linguistics, Amsterdam / Philadelphia, PA: John Benjamins, 15–31.
- Terra, Egidio, Charles L. A. Clarke (2003): "Frequency estimates for statistical word similarity measures." In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'03)*. Edmonton, CA: Association for Computational Linguistics, 165–172. <http://aclweb.org/anthology/No3-1032>.
- Terra, Egidio, Charles L. A. Clarke (2004): "Fast computation of lexical affinity models." In: *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*. Geneva: Association for Computational Linguistics. <http://aclweb.org/anthology/Co4-1147>.

- Tesnière, Lucien (2015): *Elements of Structural Syntax*. Amsterdam / Philadelphia, PA: John Benjamins. Translated by Timothy Osborne and Sylvain Kahane.
- Teufel, Simone, Gregory Grefenstette (1995): "Corpus-based method for automatic identification of support verbs for nominalizations." In: *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL'95)*. Dublin: Association for Computational Linguistics, 98–103. <http://aclweb.org/anthology/E95-1014>.
- Tognini-Bonelli, Elena (2001): *Corpus linguistics at work*. Amsterdam / Philadelphia, PA: John Benjamins.
- Tsao, Nai-Lung, David Wible (2009): "A method for unsupervised broad-coverage lexical error detection and correction." In: *Proceedings of the NAACL HLT Workshop on Innovative Use of NLP for Building Educational Applications*. Boulder, CO: Association for Computational Linguistics, 51–54. www.aclweb.org/anthology/W09-21.
- Tsao, Nai-Lung, David Wible (2013): "Word similarity using constructions as contextual features." In: *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*. Trento: Association for Computational Linguistics, 51–59. <http://aclweb.org/anthology/W13-3818>.
- Tsvetkov, Yulia, Shuly Wintner (2014): "Identification of multiword expressions by combining multiple linguistic information sources." *Computational Linguistics* 40/2: 449–468. https://doi.org/10.1162/COLI_a_00177.
- Uhrig, Peter, Stefan Evert, Thomas Proisl (2018): "Collocation candidate extraction from dependency-annotated corpora: Exploring differences across parsers and dependency annotation schemes." In: Cantos-Gómez, Pascual, Moisés Almela-Sánchez (eds.) *Lexical Collocation Analysis: Advances and Applications*, Cham: Springer, 111–140. https://doi.org/10.1007/978-3-319-92582-0_6.
- Uhrig, Peter, Thomas Proisl (2012): "Less hay, more needles – using dependency-annotated corpora to provide lexicographers with more accurate lists of collocation candidates." *Lexicographica* 28: 141–180. <https://doi.org/10.1515/lexi.2012-0009>.

- Villada Moirón, María Begoña (2005): *Data-driven identification of fixed expressions and their modifiability*. Ph.D. thesis, University of Groningen. <http://www.rug.nl/research/portal/files/9790774/thesis.pdf>.
- von Eye, Alexander (1990): *Introduction to Configural Frequency Analysis. The Search for Types and Antitypes in Cross-Classification*. Cambridge: Cambridge University Press.
- Wehrli, Eric (2007): “Fips, a ‘deep’ linguistic multilingual parser.” In: *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*. Prague: Association for Computational Linguistics, 120–127. <http://aclweb.org/anthology/W07-1216>.
- Wehrli, Eric (2014): “The relevance of collocations for parsing.” In: *Proceedings of the 10th Workshop on Multiword Expressions (MWE)*. Gothenburg: Association for Computational Linguistics, 26–32. <http://aclweb.org/anthology/W14-0804>.
- Weisstein, Eric W. (n. d.): “MathWorld—A Wolfram Web Resource.” Website. <http://mathworld.wolfram.com/>.
- Weller, Marion, Ulrich Heid (2010): “Extraction of German multiword expressions from parsed corpora using context features.” In: Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, Daniel Tapias (eds.) *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*. Valletta: European Language Resources Association, 3195–3201. http://lrec-conf.org/proceedings/lrec2010/pdf/428_Paper.pdf.
- Wermter, Joachim, Udo Hahn (2006): “You can’t beat frequency (unless you use linguistic knowledge) – a qualitative evaluation of association measures for collocation and term extraction.” In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL (ACL’06)*. Sydney: Association for Computational Linguistics, 785–792. <http://aclweb.org/anthology/P06-1099>.
- Wible, David, Nai-Lung Tsao (2010): “StringNet as a computational resource for discovering and investigating linguistic constructions.” In: *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*. Los Angeles, CA: Association for Computational Linguistics, 25–31. <http://www.aclweb.org/anthology/W10-0804>.

- Wiechmann, Daniel (2008): "On the computation of collocation strength: Testing measures of association as expressions of lexical bias." *Corpus Linguistics and Linguistic Theory* 4/2: 253–290. <https://doi.org/10.1515/CLLT.2008.011>.
- Williams, Ken (1976): "The failure of Pearson's goodness of fit statistic." *Journal of the Royal Statistical Society. Series D (The Statistician)* 25/1: 49. <https://doi.org/10.2307/2988131>.
- Yates, F. (1984): "Tests of significance for 2×2 contingency tables." *Journal of the Royal Statistical Society, Series A (General)* 147/3: 426–463. <https://doi.org/10.2307/2981577>.
- Yazdani, Majid, Meghdad Farahmand, James Henderson (2015): "Learning semantic composition to detect non-compositionality of multiword expressions." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*. Lisbon: Association for Computational Linguistics, 1733–1742. <http://www.aclweb.org/anthology/D15-1201>.
- Yoshinaga, Naoki (2004): "Improving the accuracy of subcategorizations acquired from corpora." In: *Proceedings of the ACL 2004 Workshop on Student Research (ACLstudent'04)*. Barcelona: Association for Computational Linguistics. <http://aclweb.org/anthology/P04-2008>.
- Yoshinaga, Naoki, Jun'ichi Tsujii (2004): "Generalizing subcategorization frames acquired from corpora using lexicalized grammars." In: Rambow, Owen, Matthew Stone (eds.) *Proceedings of the 7th International Workshop on Tree Adjoining Grammar and Related Formalisms (TAG+7)*. Vancouver, 104–110. <http://aclweb.org/anthology/W04-3314>.
- Zaiu Inkpen, Diana, Graeme Hirst (2002): "Acquiring collocations for lexical choice between near-synonyms." In: *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*. Philadelphia, PA: Association for Computational Linguistics, 67–76. <http://aclweb.org/anthology/W02-0909>.
- Zeman, Daniel (2008): "Reusable tagset conversion using tagset drivers." In: Calzolari, Nicoletta, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Daniel Tapias (eds.) *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech: European Language Resources Association. http://www.lrec-conf.org/proceedings/lrec2008/pdf/66_paper.pdf.

- Zinsmeister, Heike, Ulrich Heid (2003): "Significant triples: Adjective+noun+verb combinations." In: *Proceedings of the 7th Conference on Computational Lexicography and Text Research (Complex 2003)*. Budapest. <http://www.ims.uni-stuttgart.de/%7Ezinsmeis/pubs/SigColl-paper.pdf>.
- Zinsmeister, Heike, Ulrich Heid (2004): "Collocations of complex nouns: Evidence for lexicalisation." In: *Proceedings of KONVENS 2004*. Vienna. http://www.ims.uni-stuttgart.de/%7Ezinsmeis/pubs/ZinsmeisterHeid_Konvo4.pdf.
- Zipser, Florian, Laurent Romary (2010): "A model oriented approach to the mapping of annotation formats using standards." In: *Proceedings of the Workshop on Language Resource and Language Technology Standards, LREC 2010*. Valetta. https://hal.inria.fr/inria-00527799/file/ZipserRomary_final.pdf.

The study of cooccurrences, i. e. the analysis of linguistic units that occur together, has had a profound impact on our view of language. Not only has it contributed greatly to the insight that semi-preconstructed phrases and item-specific knowledge are central to how language works, but it has also led to improved dictionaries and teaching materials. Cooccurrences of various linguistic items have been studied under a variety of names, e.g. collocation, colligation or collostruction. While there are well-understood and fully worked out statistical models for the analysis of cooccurrences of pairs of words, no such model exists for cooccurrences of larger linguistic structures. This situation is remedied by the current work.

Building on the well-understood 2×2 contingency tables and a graph-based representation of linguistic structures, we develop the generalized cooccurrence model, an explicit formal model for the statistical analysis of cooccurrences of arbitrary linguistic structures. Existing methods for the analysis of two-word cooccurrences and for collostructional analysis are shown to be simply special cases of the generalized cooccurrence model.

