

TRISTAN MILLER

ADJUSTING SENSE
REPRESENTATIONS FOR WORD
SENSE DISAMBIGUATION AND
AUTOMATIC PUN INTERPRETATION



Vom Fachbereich Informatik der Technische Universität Darmstadt genehmigte Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.), vorgelegt von Tristan Miller, M.Sc., B.Sc. (Hons.) aus Regina

Einreichung: 4. Januar 2016

Disputation: 22. März 2016

Referenten: Prof. Dr. Iryna Gurevych
Prof. Dr. Rada Mihalcea
Prof. Dr. Wolf-Tilo Balke

Darmstadt, 2016
D 17

© 2016 Tristan Miller

Please cite this document as

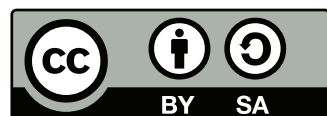
URN: urn:nbn:de:tuda-tuprints-54002

URL: <http://tuprints.ulb.tu-darmstadt.de/id/eprint/5400>

This document is provided by tuprints,
E-Publishing-Service of Technische Universität Darmstadt.

<http://tuprints.ulb.tu-darmstadt.de>

tuprints@ulb.tu-darmstadt.de



This work is published under the terms of the Creative Commons *Namensnennung – Weitergabe unter gleichen Bedingungen 3.0 Deutschland* (CC BY-SA) licence:

<https://creativecommons.org/licenses/by-sa/3.0/de/>

ABSTRACT

Word sense disambiguation (wSD)—the task of determining which meaning a word carries in a particular context—is a core research problem in computational linguistics. Though it has long been recognized that supervised (machine learning-based) approaches to wSD can yield impressive results, they require an amount of manually annotated training data that is often too expensive or impractical to obtain. This is a particular problem for under-resourced languages and domains, and is also a hurdle in well-resourced languages when processing the sort of lexical-semantic anomalies employed for deliberate effect in humour and wordplay. In contrast to supervised systems are knowledge-based techniques, which rely only on pre-existing lexical-semantic resources (LSRs). These techniques are of more general applicability but tend to suffer from lower performance due to the informational gap between the target word’s context and the sense descriptions provided by the LSR.

This dissertation is concerned with extending the efficacy and applicability of knowledge-based word sense disambiguation. First, we investigate two approaches for bridging the information gap and thereby improving the performance of knowledge-based wSD. In the first approach we supplement the word’s context and the LSR’s sense descriptions with entries from a distributional thesaurus. The second approach enriches an LSR’s sense information by aligning it to other, complementary LSRs.

Our next main contribution is to adapt techniques from word sense disambiguation to a novel task: the interpretation of puns. Traditional NLP applications, including wSD, usually treat the source text as carrying a single meaning, and therefore cannot cope with the intentionally ambiguous constructions found in humour and wordplay. We describe how algorithms and evaluation methodologies from traditional word sense disambiguation can be adapted for the “disambiguation” of puns, or rather for the identification of their double meanings.

Finally, we cover the design and construction of technological and linguistic resources aimed at supporting the research and application of word sense disambiguation. Development and comparison of wSD systems has long been hampered by a lack of standardized data formats, language resources, software components, and workflows. To address this issue, we designed and implemented a modular, extensible framework for wSD. It implements, encapsulates, and aggregates reusable, interoperable components using UIMA, an industry-standard information processing architecture. We have also produced two large sense-annotated data sets for under-resourced languages or

domains: one of these targets German-language text, and the other English-language puns.

ZUSAMMENFASSUNG

Lesartendisambiguierung (engl. *word sense disambiguation*, oder *wsd*) ist ein Kernforschungsproblem der Computerlinguistik und beschreibt die Aufgabe, festzustellen, welche Bedeutung ein bestimmtes Wort in einem bestimmten Kontext hat. Schon seit langem hat man erkannt, dass überwachte (d. h. auf maschinellem Lernen basierende) Ansätze für *wsd* zu beeindruckenden Ergebnissen führen können, jedoch benötigen diese eine große Menge an manuell annotierten Trainingsdaten, deren Herstellung oder Beschaffung oft zu aufwändig oder unpraktisch ist. Dies ist insbesondere ein Problem bei Sprachen und Domänen, für die wenige Ressourcen zur Verfügung stehen, und wenn es um die Verarbeitung der lexikalisch-semanticen Anomalien geht, die typischerweise für Humor und Wortspiele eingesetzt werden. Im Gegensatz zu überwachten Systemen verlassen sich wissensbasierte Verfahren nur auf bereits bestehende lexikalisch-semantiche Ressourcen (*LSRs*). Obwohl diese Verfahren breiter anwendbar sind, kommt es häufig zu Qualitätseinbußen aufgrund der Informationslücke zwischen dem Kontext des Zielworts und den Bedeutungsbeschreibungen, die die *LSR* zur Verfügung stellt.

Diese Dissertation beschäftigt sich mit der Verbesserung der Wirksamkeit und Anwendbarkeit wissensbasierter Lesartendisambiguierung. Ihre Hauptbeiträge sind die drei folgenden: Zunächst untersuchen wir zwei Ansätze zur Überbrückung der Informationslücke und damit zur Verbesserung der Leistung von wissensbasiertem *wsd*. Im ersten Ansatz erweitern wir den Kontext des Wortes und die Bedeutungsbeschreibungen der *LSR* mit Einträgen aus einem distributionellen Thesaurus, der zweite Ansatz ergänzt die Bedeutungsinformationen einer *LSR* durch die Verknüpfung mit anderen, komplementären *LSRs*.

Unser nächster Hauptbeitrag ist die Anpassung von *wsd*-Techniken an eine neue Aufgabe: die Interpretation von Wortspielen. Traditionelle linguistische Datenverarbeitung, einschließlich *wsd*, behandelt den Quelltext normalerweise so, als ob er nur eine einzige Bedeutung trägt und kann deshalb nicht mit absichtlich mehrdeutigen Konstruktionen von Humor und Wortspielen umgehen. Wir beschreiben, wie man Algorithmen und Evaluierungsmethoden der traditionellen Lesartendisambiguierung anpassen kann, um Wortspiele zu „disambiguieren“, oder besser gesagt, um ihre doppelte Bedeutung zu erkennen.

Schließlich beschreiben wir die Konzeption und Konstruktion technischer und linguistischer Ressourcen, die die Forschung und Anwendung wissensbasierter Lesartendisambiguierung unterstützen. Die Entwicklung und der Vergleich von wsd-Systemen wurden schon seit langem durch einen Mangel an standardisierten Datenformaten, Sprachressourcen, Softwarekomponenten und Arbeitsabläufen behindert. Um dieses Problem anzugehen, haben wir ein modulares, erweiterbares Framework für wsd konzipiert und umgesetzt. Es implementiert, kapselt und aggregiert wiederverwendbare, kompatible Komponenten mit UIMA, einer Informationsverarbeitungsarchitektur nach Industriestandard. Darüber hinaus haben wir zwei große Korpora erstellt, in denen Wörter mit den entsprechenden Wortbedeutungen annotiert wurden: eines für deutschsprachigen Text, und eines für englischsprachige Wortspiele.

ACKNOWLEDGMENTS

Though it is my name that appears on the cover of this thesis, a great many others contributed in one way or another to its realization.

Chronologically, the first people to whom I am indebted are Willard R. Espy, for introducing me to the whimsical world of wordplay; Dmitri A. Borgmann, for showing me that it was a subject worthy of serious, even scholarly, study; and Jim Butterfield, for instilling in me a lifelong love of writing and sharing software. May the world never forget the contributions of these late, great writers and popularizers.

It is my great fortune to have conducted my research at the Ubiquitous Knowledge Processing Lab under the aegis of Prof. Iryna Gurevych. I am honoured to have been accepted into the vibrant and successful academic community she created, and grateful to have been able to contribute to and draw upon its diverse talents and resources. I am particularly grateful to Prof. Gurevych for having afforded me the opportunity to carry out important basic research, as well as the freedom to pursue some of its unusual but no less important applications.

Thanks are also due to my current and former colleagues at UKP who made these last few years such a rewarding experience. I am particularly indebted to the ten other co-authors of the papers upon which much of this dissertation is based and with whom I have had many fruitful discussions; to Dr. Michael Matuschek, Lisa Beinborn, and Susanne Neumann for helping to improve my German and for providing many welcome distractions from my work; and to all those who contributed to the invaluable DKPro family of tools and resources.

Among colleagues outside UKP, I would like to give special acknowledgment to Dr. Verena Henrich for sharing her data and expertise, to Profs. Rada Mihalcea and Wolf-Tilo Balke for serving as reviewers for this thesis, and to Prof. Christian “Kiki” Hempelmann and others in the ISHS who so warmly welcomed me into the humour studies community.

I am very grateful to the authors, compilers, and annotators of the texts for my data sets: thank you to Daniel Austin, John Black, Dr. Matthew Collins, Emily Jamison, Constanze Hahn, Prof. Christian Hempelmann, Silke Kalmer, Stan Kegel, Annabel Köppel, Don Hauptman, Andrew Lamont, Mladen Turković, Dr. Beatrice Santorini, and Andreas Zimpfer.

Моей дорогой Наде и нашим трём «казикам», Торе, Сёвре, и Пафнутию: спасибо за вашу любовь и поддержку!

Finally, grateful acknowledgment must be made to the various funding agencies that sponsored my work. These are the Volkswagen Foundation as part of the Lichtenberg Professorship Program under

grant № 1/82806, the Federal Ministry of Education and Research under grant № 01IS10054G, and the German Research Foundation as part of the project “Integrating Collaborative and Linguistic Resources for Word Sense Disambiguation and Semantic Role Labeling” (InCoRe, GU 798/9-1) and the research training group “Adaptive Preparation of Information from Heterogeneous Sources” (AIPHES, GRK 1994/1).

CONTENTS

1	INTRODUCTION	1
1.1	Motivation	1
1.2	Contributions	4
1.3	Chapter outline	7
1.4	Publication record	8
1.5	Notational conventions	11
2	WORD SENSE DISAMBIGUATION	13
2.1	Task description	13
2.2	Knowledge sources	14
2.2.1	External knowledge sources	14
2.2.2	Contextual knowledge	16
2.3	Approaches to disambiguation	18
2.3.1	Knowledge-based	18
2.3.2	Supervised	21
2.3.3	Unsupervised	22
2.4	Evaluation methodologies	22
2.4.1	Data sets	24
2.4.2	Evaluation metrics	28
2.4.3	Lower and upper bounds	29
2.4.4	Shared tasks	30
2.5	Chapter summary	32
3	LEXICAL EXPANSION FOR WSD	35
3.1	Motivation	35
3.2	Background and related work	36
3.2.1	The Lesk algorithm	36
3.2.2	Distributional similarity	39
3.3	Approach	41
3.4	Experiments	43
3.4.1	Use of distributional information	44
3.4.2	Data sets	46
3.4.3	Baselines and measures	47
3.4.4	Results	48
3.5	Analysis	51
3.5.1	Error analysis	52
3.6	Conclusion	54
4	WORD SENSE ALIGNMENT AND CLUSTERING	57
4.1	Motivation	57
4.2	Background and related work	58
4.2.1	Lexical-semantic resources	58
4.2.2	Pairwise alignments	59
4.2.3	Word sense clustering	61
4.3	Approach	63
4.4	Results	64

4.5	Evaluation	67
4.5.1	Clustering of wsd results	68
4.5.2	Enriched sense inventory for knowledge-based wsd	71
4.6	Conclusion	73
5	PUN INTERPRETATION	75
5.1	Motivation	75
5.2	Background and related work	77
5.2.1	Puns	77
5.2.2	Word sense disambiguation	78
5.2.3	Computational humour	80
5.2.4	Corpora	81
5.3	Data set	81
5.3.1	Raw data	82
5.3.2	Sense annotation	84
5.3.3	Analysis	85
5.4	Pun identification	88
5.4.1	Task description	88
5.4.2	Approach	88
5.4.3	Evaluation methodology	90
5.4.4	Results	92
5.5	Pun detection	94
5.5.1	Evaluation methodology	95
5.5.2	Baselines and results	96
5.6	Conclusion	97
6	DKPRO WSD	99
6.1	Motivation	99
6.2	Background and related work	100
6.2.1	Resources for wsd	100
6.2.2	Monolithic wsd systems	101
6.2.3	Processing frameworks	101
6.3	System description	103
6.4	Conclusion	109
7	GLASS	111
7.1	Motivation	111
7.2	Background and related work	112
7.2.1	Sense-annotated data for German	112
7.2.2	Lexical substitution	114
7.3	Data set construction	118
7.3.1	Resource and data selection	118
7.3.2	Annotation process	119
7.4	Analysis	120
7.4.1	Interannotator agreement	120
7.4.2	Characterizing lexical substitutions	121
7.5	Conclusion	126
8	CONCLUSION	129

8.1	Summary	129
8.2	Outlook	131
A	PUN ANNOTATION GUIDELINES	135
B	PUN ANNOTATION ADJUDICATION GUIDELINES	143
C	SENSE LINKING GUIDELINES	147
D	SENSE LINKING ADJUDICATION GUIDELINES	153
E	GLASS INTERANNOTATOR AGREEMENT	157
BIBLIOGRAPHY		163
CURRICULUM VITÆ		191
INDEX		195

LIST OF FIGURES

Figure 1	Outline of the thesis	9
Figure 2	Sample wsd text processing pipeline	17
Figure 3	Intuition behind lexical expansion	42
Figure 4	Results on SemEval-2007 for simplified and simplified extended Lesk, by number of lexical expansions	48
Figure 5	Flowchart for the connected components algorithm	65
Figure 6	Selecting pun words in Punnotator	84
Figure 7	Selecting definitions in Punnotator	84
Figure 8	Punnotator XML output (excerpt)	85
Figure 9	A UIMA aggregate analysis engine	103
Figure 10	A sample DKPro WSD pipeline	104
Figure 11	DKPro WSD's interactive visualization of a graph connectivity WSD algorithm	107
Figure 12	An HTML report produced by DKPro WSD	108
Figure 13	Format of the Cholakov <i>et al.</i> (2014) data set's XML files	118
Figure 14	Sample line from the Cholakov <i>et al.</i> (2014) data set's stand-off annotation files	118
Figure 15	Punnotator pun selection page	137
Figure 16	Punnotator pun annotation page	139
Figure 17	Pun annotation adjudication page	144
Figure 18	Ubyline login page	147
Figure 19	Ubyline home page	148
Figure 20	Ubyline sense linking overview page	148
Figure 21	Ubyline sense example linking page	149

LIST OF TABLES

Table 1	Data sets for monolingual English SENSEVAL and SemEval WSD tasks	33
Table 2	A DT entry with features, showing terms similar to the noun <i>paper</i>	45
Table 3	Results on SemEval-2007 by part of speech	49
Table 4	Results on SENSEVAL-2 and -3	51
Table 5	Confusion matrix for SL+0 and SL+100	52

Table 6	Confusion matrix for SEL+o and SEL+100	53
Table 7	Distribution of synonym sets by cardinality in the two- and three-way conjoint alignments	61
Table 8	Word sense distribution in WordNet, Wiktionary, and the three-way conjoint alignment	66
Table 9	SENSEVAL-3 WSD accuracy using our MFF-purified clusters and random clustering of equivalent granularity	71
Table 10	SENSEVAL-3 WSD accuracy using our LFF-purified clusters and random clustering of equivalent granularity	71
Table 11	SENSEVAL-3 WSD accuracy using simplified Lesk, with and without alignment-enriched sense glosses	72
Table 12	SENSEVAL-3 WSD accuracy using simplified extended Lesk with 30 lexical expansions, with and without alignment-enriched sense glosses	72
Table 13	Coverage, precision, recall, and F ₁ for various pun disambiguation algorithms	92
Table 14	Coverage, precision, and recall for SEL+cluster ₂ , and random baseline recall, according to part of speech	94
Table 15	Classification contingency matrix	95
Table 16	Baseline results for the pun detection and location tasks	97
Table 17	Comparison of sense-tagged corpora for German	113
Table 18	Lexical substitution data sets from evaluation campaigns	115
Table 19	Number of lemmas in GLASS by part of speech and polysemy in GermaNet	122
Table 20	Percentage of substitutes in successfully sense-annotated items in GLASS by their connection to the sense(s) through various semantic relations in GermaNet	123
Table 21	Paraset purity and common core statistics for GLASS, by part of speech	125

LIST OF FORMULAS

1.1	Iverson bracket	11
2.1	Word sense disambiguation by semantic similarity	20
2.2	Minimum percentage agreement	25
2.3	Maximum percentage agreement	25
2.4	Mean Dice agreement	25
2.5	Cohen's κ	26
2.6	Expected agreement for Cohen's κ	26
2.7	Krippendorff's α	27
2.8	Set of all applied annotations	27
2.9	Number of annotators who applied a given annotation to a given item	27
2.10	Number of items with a given annotation	27
2.11	Distance function for singleton sense annotations	27
2.12	MASI distance metric	28
2.13	Jaccard index	28
2.14	MASI monotonicity factor	28
2.15	Disambiguation score	28
2.16	Disambiguation coverage	28
2.17	Disambiguation precision	29
2.18	Disambiguation recall	29
2.19	F-score	29
2.20	Precision and recall for random sense baseline	29
3.1	Original Lesk algorithm (pairwise)	36
3.2	Original Lesk algorithm (generalized)	36
3.3	Simplified Lesk algorithm	38
3.4	Extended Lesk algorithm (pairwise)	38
3.5	Simplified extended Lesk algorithm	38
4.1	Equivalence class of s under \sim	63
4.2	Quotient set of S by \sim	63
4.3	Full alignment of two sources	63
4.4	Full alignment of a set of sources	63
4.5	Conjoint alignment of a set of sources	63
4.6	Alignment cardinality	64
4.7	Snow's random clustering score	69
4.8	Number of ways of clustering N_i senses, G_i of which are correct and one of which is incorrectly chosen	69

4.9	Number of ways of clustering N_i senses, where an incorrectly chosen sense is clustered with none of the G_i correct senses	69
4.10	Probability of an incorrectly chosen sense being clustered together with at least one correct sense	70
4.11	Probability of an incorrectly chosen sense being clustered together with at least one correct sense (recast for ease of programmatic computation)	70
4.12	Probability of an incorrectly chosen sense being clustered together with the only correct sense	70
5.1	MASI distance function adapted to pun annotation	86
5.2	Pun disambiguation score	91
5.3	Random sense baseline score for pun disambiguation	91
5.4	Precision (as in information retrieval)	96
5.5	Recall (as in information retrieval)	96
5.6	Accuracy (as in information retrieval)	96
5.7	Random baseline precision for pun detection	96
5.8	Random baseline precision for pun location	96
7.1	Cluster purity	125

INTRODUCTION

1.1 MOTIVATION

Polysemy is a fundamental characteristic of all natural languages. Writers, philosophers, linguists, and lexicographers have long recognized that words have multiple meanings, and moreover that more frequently used words have disproportionately more senses than less frequent ones (Zipf, 1949). For instance, take the following sentences:

- (1) The river *runs* through the forest.
- (2) He *runs* his own business.
- (3) My new notebook *runs* GNU/Linux.

The word *runs* has distinct meanings in each sentence: in the first case, it means to flow as liquid from one point to another; in the second, to manage or be in charge of something; and in the third, to execute computer software. It would be easy to construct further examples; according to the former chief editor of the *Oxford English Dictionary*, the verb form alone of the word *run* has no fewer than 645 distinct senses (Winchester, 2011).

Despite this plurality of meanings, humans do not normally perceive any lexical ambiguity in processing written or spoken language; each polysemous word is unconsciously and automatically understood to mean exactly what the writer or speaker intended (Hirst, 1987). Computers, however, have no inherent ability to process natural language, and as early as the 1940s and 1950s, computing pioneers had recognized polysemy as a major challenge to machine translation (Weaver, 1955; Bar-Hillel, 1960). To see why this is the case, recall that bilingual dictionaries used by humans are not simply one-to-one mappings between the words of the two languages. Rather, for each word in the source language, its meanings are enumerated and then separately mapped to (possibly distinct) words in the target language. For instance, the three occurrences of *runs* in Examples 1 to 3 above would be translated with three different words in German:

- (4) Der Fluß *fließt* durch den Wald.
- (5) Er *betreibt* eine eigene Firma.
- (6) Mein neues Notebook *läuft* unter GNU/Linux.

Choosing the correct translation is therefore predicated on the ability to identify the source word's meaning in context. To a computer, however—even one with a machine-readable bilingual dictionary—there is no obvious programmatic way to relate the words in the



source text to their dictionary meanings, and therefore no way of choosing among the translation candidates.

Subsequent researchers have noted the obstacle lexical polysemy poses to other tasks in computational linguistics. In information retrieval, for example, a computer is expected to search through a collection of documents and return those most relevant to a user's query. Regardless whether that query is given as keywords or in more natural phrasing, lexical ambiguity in the query and collection can result in inaccurate results. (No one who searches the Web for information on the disease AIDS wants to wade through pages of results all about hearing aids.) (Krovetz, 1997) The importance of resolving lexical ambiguity has also been recognized for automated spelling correction (Yarowsky, 1994) and in information extraction (*e.g.*, Markert and Nissim, 2007).

It should come as no surprise, then, that automatically identifying and discriminating between word senses—*i.e.*, *word sense disambiguation*, or WSD—has been the subject of extensive and continuous study in computational linguistics since the very nascence of the field. The *ACL Anthology*¹—an online archive containing some 50 years' worth of research papers in computational linguistics—currently has over 400 papers with “word sense disambiguation” or “WSD” in the title; the two terms appear in the main text of thousands more. Word sense disambiguation has been the subject of several scholarly survey papers (Ide and Véronis, 1998; Navigli, 2009), special issues of journals (Hirschberg, 1998; Kilgarriff and Palmer, 2000; Edmonds and Kilgarriff, 2002b; Preiss and Stevenson, 2004), encyclopedia entries (Edmonds, 2005; Mihalcea, 2011), and entire books or chapters thereof (Manning and Schütze, 1999; Agirre and Edmonds, 2007; Yarowsky, 2010; Kwong, 2013). Its popularity has led to a long-running series of workshops and evaluation competitions (see §2.4.4) and has given rise to closely related tasks such as named entity linking, word sense induction, and lexical substitution.

Despite the considerable research attention WSD has received, producing disambiguation systems which are both practical and highly accurate has remained an elusive open problem. The task has even been described as “AI-complete”, by analogy to NP-completeness in complexity theory (Mallery, 1988). Much of the difficulty is due to the myriad ways in which the lexical disambiguation task can be formalized and parameterized—in some scenarios it is every word in a text which must be disambiguated, whereas in others it is only all occurrences of a single word; some scenarios may prescribe a particular inventory of senses with which words must be annotated, where others may expect systems to induce their own; some may be concerned with a particular text domain, whereas others may not. Even

¹ <http://aclanthology.info/>

subtle variations in the task setup may necessitate drastic changes in the approach and resources used for disambiguation.

A second source of difficulty has to do with the nature of word senses themselves. While the discreteness of word senses is an implicit assumption of WSD (Kwong, 2013), in reality the boundaries are notoriously hard to distinguish. Consider the following sentences:

- (7) He made a *box* for the cookies.
- (8) He ate a *box* of cookies.
- (9) He bought a *box* of cookies.

In the first of these examples, the word *box* clearly refers to a type of container, and in the second sentence, the same term refers to the *contents* of such a container. The third sentence presents us with a quandary—is its *box* used in the sense of the first sentence, or of the second sentence, or of both? Or is it perhaps better to say that it is used in a third distinct sense of “the container and its contents”?

In fact, there is no decisive way of delimiting the boundaries between word meanings. The divisions made by lexicographers are, to a large degree, subjective editorial decisions made for the convenience of human readers and to constrain a particular dictionary’s intended scope and length. *Merriam-Webster’s Collegiate Dictionary* (2004), for example, distinguishes between all three of the above senses of *box*, whereas *The Oxford Dictionary of Current English* (Thompson, 1993) lists only the first two, and *Harrap’s Easy English Dictionary* (Collin, 1980) lists only the first sense, presumably considering the other two meanings to be trivially subsumed, or perhaps as metonymic uses rather than distinct word senses in their own right.² In any case, traditional lexicographic distinctions are not always useful for computational understanding of text. In fact, there is much evidence to suggest that the sense distinctions of lexicographic resources are far subtler than what is needed for real-world NLP applications, and sometimes even too subtle for humans to reliably recognize (Jorgensen, 1990; Palmer, 2000; Ide and Wilks, 2007). This makes improving upon experimental results difficult, and lessens the benefit of WSD in downstream applications.

A third main source of difficulty is the *knowledge acquisition bottleneck*. All WSD systems employ knowledge sources of some sort to associate words with their contextually appropriate meanings. Depending on the method used, these knowledge sources can include machine-readable versions of traditional dictionaries and thesauri, semantic

² There are some who have even questioned the very existence of word senses, at least as traditionally understood (e.g., Kilgarriff, 1997; Hanks, 2000). An alternative view is that each word is a single lexical entry whose specific meaning is *underspecified* until it is activated by the context (Ludlow, 1996). In the case of *systematically polysemous* terms (i.e., words which have several related senses shared in a systematic way by a group of similar words), it may not be necessary to disambiguate them at all in order to interpret the communication (Buitelaar, 2000).

networks, and raw or manually annotated text corpora. However, the manual construction of such knowledge sources is known to be a particularly arduous task, and it may need to be repeated each time the configuration of the task changes, such as switching to a new domain or sense representation.

Approaches to wsd are commonly categorized according to the types of knowledge sources they employ. *Supervised* techniques use machine learning classifiers which are trained on text to which sense annotations have been manually applied. Though they can yield impressive results, their applicability is restricted to those languages and domains for which there exists a sufficient amount of sense-annotated training data. To date, most research in supervised wsd has been conducted on English text; for most other languages, including German, sense-annotated training data is scarce or nonexistent.

In contrast to supervised systems are *knowledge-based* techniques that rely only on untagged knowledge sources such as dictionaries and raw corpora. While such techniques are of more general applicability, they tend to suffer from lower overall performance. This is commonly attributable to the informational gap between the target word's context and the sense descriptions against which it must be matched. That is, the word meanings encoded in dictionaries and other lexical-semantic resources (LSRs) are so brief or succinct that they bear little obvious relation to the words found in the immediate context of the target.

Traditional approaches to word sense disambiguation, be they supervised or knowledge-based, rest on the assumption that there exists a single unambiguous communicative intention underlying every word in the document or speech act under consideration. Under this assumption, lexical ambiguity arises due to there being a plurality of words with the same surface form but different meanings, and the task of the interpreter is to select correctly among them. However, there exists a class of language constructs known as *paronomasia*, or *puns*, in which homonymic (*i.e.*, coarse-grained) lexical-semantic ambiguity is a *deliberate* effect of the communication act. That is, the writer intends for a certain word or other lexical item to be interpreted as simultaneously carrying two or more separate meanings. Though puns are a recurrent and expected feature in many discourse types, current word sense disambiguation systems, and by extension the higher-level natural language applications making use of them, are completely unable to deal with them.

1.2 CONTRIBUTIONS

In the previous section, we discussed how identification of the contextually appropriate meanings of words is a prerequisite for many

natural language processing tasks. Furthermore, we identified the following barriers to the construction of accurate wsd systems, and to their implementation and evaluation:

- (i) The accuracy of knowledge-based approaches to wsd is hampered by the sparsity of information in the knowledge sources they rely on.
- (ii) The word sense distinctions made by individual LSRs are often too subtle for accurate wsd.
- (iii) Traditional approaches to wsd are incapable of processing the sort of intentional lexical ambiguity found in puns.
- (iv) There are multiple ways in which the task of wsd can be framed, but even small variations in the task configuration may require drastic changes to the approach and resources used to implement and evaluate the systems.
- (v) Most research on wsd to date has been conducted in English; data sets for German are comparatively rare.

The present dissertation is concerned with improving the efficacy and applicability of word sense disambiguation, and in particular knowledge-based wsd. Our five main contributions aim to address the five problems listed above:

1. In order to address Problem (i), we investigate two approaches for bridging the informational gap in knowledge-based wsd. Both approaches involve enriching the information available to wsd systems by aligning or merging it with additional lexical-semantic resources.

Our first approach for bridging the informational gap explores the contribution of distributional semantics. We use a *distributional thesaurus*—a concordance produced from a large automatically parsed corpus—for lexical expansion of the context and sense definitions. We apply this mechanism to traditional knowledge-based wsd algorithms and show that the distributional information significantly improves disambiguation results across several standard data sets. In fact, this improvement exceeds the state of the art for disambiguation without sense frequency information—a situation which is commonly encountered with new domains or with languages for which no sense-annotated corpus is available—and does not appear to have been bested by later methods based on neural networks and word embeddings.

The second approach to bridging the information gap involves enriching an LSR's sense information by automatically aligning it to other, complementary LSRs. Such alignments have been

used in the past to increase the coverage and quality of LSRs, but these attempts have always been between *pairs* of specific resources. To maximize the amount of sense information available to our algorithms, we take the novel step of merging multiple pairwise alignments into a single omnibus alignment. We therefore describe a method for automatically constructing n-way alignments from arbitrary pairwise alignments and apply it to two pre-existing state-of-the-art alignments. We find the resulting three-way alignment to have greater coverage, an enriched sense representation, and coarser sense granularity than both the original resources and their pairwise alignments, though this came at the cost of accuracy. As predicted, we find use of the alignments to enrich a sense inventory with additional sense glosses to significantly improve wsd performance.

2. A commonly proposed solution to the sense granularity problem (Problem (ii)) is to simply switch to a different LSR. However, this option is rarely practical: some applications and most evaluation frameworks are tied to a particular sense inventory, and alternative coarse-grained LSRs may come with drawbacks, such as a lack of coverage. Our own solution is to coarsen an existing LSR by collapsing its senses with the aid of the aforementioned two- and three-way alignments. Specifically, we induce a clustering of senses from one resource by first mapping them to those in the other resources, and then grouping the source senses which map to the same target sense.

We find our coarsened sense inventories to yield significantly higher chance-corrected accuracy in various wsd tasks. In constructing our experiments, we also discovered a significant flaw in a popular cluster evaluation metric; part of our contribution here is therefore a corrected version of this metric.

3. We apply our new word sense disambiguation algorithms and word sense clusters to Problem (iii)—the novel task of computational interpretation of puns. Traditional approaches to wsd, as with NLP generally, usually treat the source text as unambiguous in intention. However, writers sometimes intend for a word to be interpreted as simultaneously carrying multiple distinct meanings. This deliberate use of lexical ambiguity—*i. e.*, punning—is particularly common in advertising and in humour. We describe how traditional, language-agnostic wsd approaches can be adapted to “disambiguate” puns, or rather to identify their double meanings. We document our creation of a large, manually sense-annotated corpus of English puns and use it as a gold standard for evaluating our pun disambiguation algorithms. We observe performance exceeding that of some knowledge-based and supervised baselines.

4. Many of the difficulties of Problem (iv) can be attributed to the historic lack of standardized data formats, resources, components, and workflows for wsd. This problem has long hampered interoperability of wsd software and data, and correspondingly affected the reproducibility of experimental results. To address this issue, and to make possible the rapid development and testing of the tools required for wsd research, we designed and implemented DKPro wsd, a modular, extensible software framework for word sense disambiguation. This framework implements (or encapsulates) and aggregates reusable, interoperable components using UIMA, an industry-standard information processing architecture. This thesis describes in detail the design and operation of this framework, and situates it in the wider context of reusable NLP components developed under the DKPro banner.
5. The dearth of sense-annotated data sets in languages other than English (Problem (v)) has been a particular problem for German. Though some sense-annotated corpora exist, none of them is fully available under a free content licence, and many of them have poor coverage of certain classes of words. Our fifth contribution is the construction and analysis of GLASS, a new German-language sense-annotated data set. The GLASS data set fills a gap in German-language resources by providing a well balanced selection of words with high-quality, manually applied sense annotations. Moreover, because it extends an existing data set of lexical substitutions, it allows for intrinsic and extrinsic evaluations of wsd systems to be carried out on the same data, and is the first resource to permit an empirical study of the relationship between manually annotated word senses and lexical substitutes.

1.3 CHAPTER OUTLINE

The remaining chapters of this thesis are structured as follows:

- Chapter 2 provides an introduction to word sense disambiguation as a field of research. The background it covers is, by and large, necessary to understand and contextualize the research presented in the thesis as a whole. Though subsequent chapters provide background sections of their own, the topics they cover are specific to those chapters.
- In Chapter 3, we detail our approach to using distributional information for knowledge-based wsd.

- Chapter 4 treats the topic of aligning heterogeneous lexical-semantic resources for enhancing WSD and for sense coarsening.
- In Chapter 5, we apply the principles and techniques of the previous two chapters to the novel task of automatic pun interpretation.
- In the next two chapters, we describe the enabling technology and resources we have produced. Chapter 6 describes DKPro WSD, our software framework supporting the rapid development and testing of sense disambiguation systems, and Chapter 7 covers GLASS, our sense-annotated German-language data set for word sense disambiguation and lexical substitution.
- Chapter 8 concludes the thesis by summarizing our contributions and identifying some applications and areas for future development of our ideas.

The diagram in Figure 1 illustrates the research contributions described in the previous section and how they are covered by Chapters 3 through 7.

1.4 PUBLICATION RECORD

Many of the main ideas of this thesis have been previously published, or are pending publication, in peer-reviewed journals and conference proceedings in the fields of computational linguistics and humour studies. The specific publications, and the chapters we have derived from them, are listed below. All the publications have joint authorship, and except as specifically noted below, the material presented in this thesis should be assumed to be primarily the contribution of the present author.

- Chapter 3 is based largely on “Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation” (Miller, Biemann, *et al.*, 2012), a paper presented at the 24th International Conference on Computational Linguistics (COLING 2012) in Mumbai. The ideas for exploring the use of a distributional thesaurus as a source of lexical expansions arose from a series of discussions with Chris Biemann, who also constructed the thesaurus itself. This construction is described in brief at the beginning of §3.4.1; a more detailed treatment of the technique appears in Biemann and Riedl (2013).
- Chapter 4 is based on two papers on word sense alignment. The first of these, “WordNet–Wikipedia–Wiktionary: Construction of a Three-way Alignment” (Miller and Gurevych, 2014), was presented at the 9th International Conference on Language

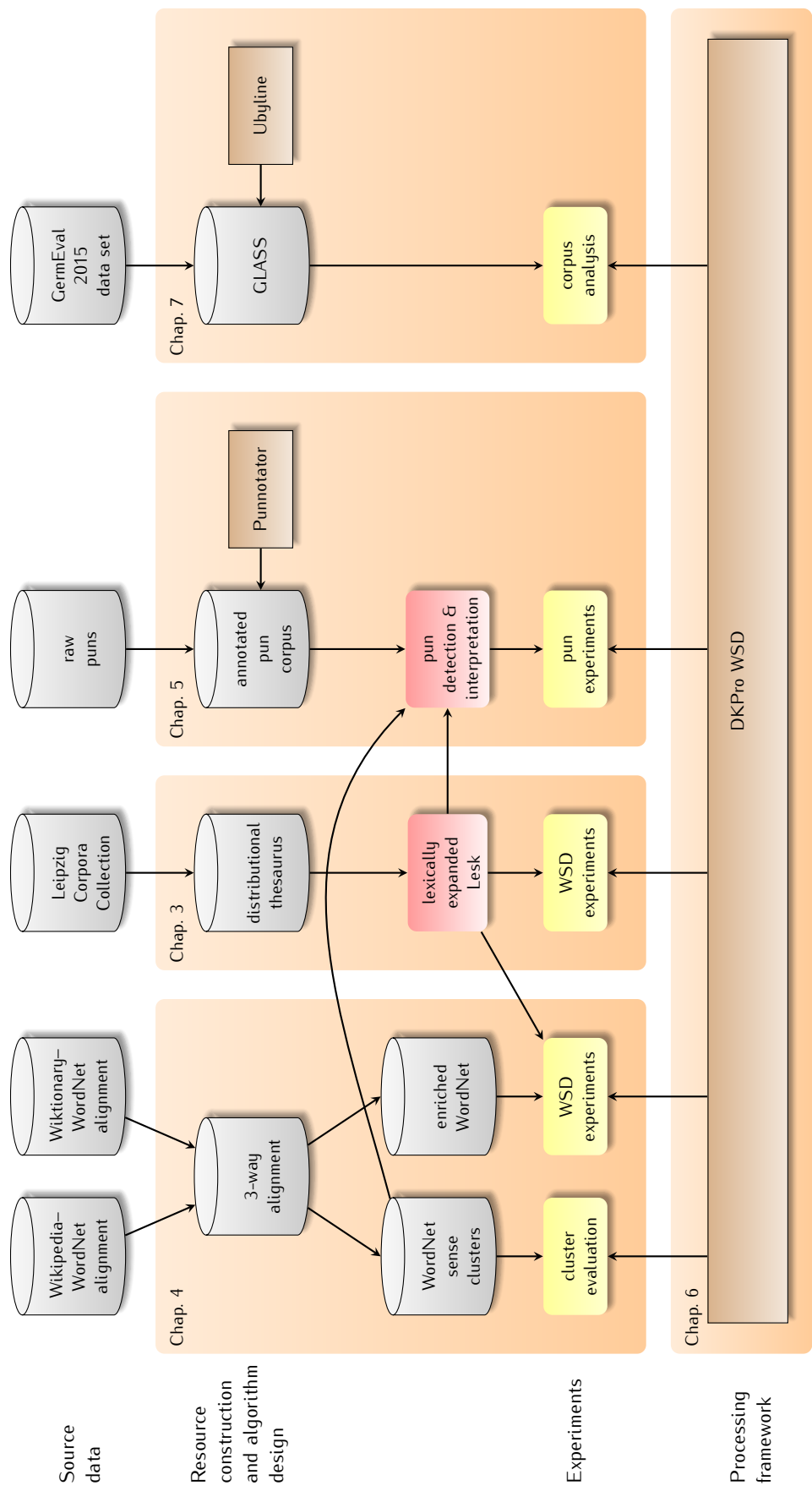


Figure 1. Outline of the thesis

Resources and Evaluation (LREC 2014) in Reykjavík, and the second, “A Language-independent Sense Clustering Approach for Enhanced wsd” (Matuschek, Miller, *et al.*, 2014), was presented at the 12th *Konferenz zur Verarbeitung natürlicher Sprache* (KONVENS 2014) in Hildesheim. The only material from the latter paper presented here is our novel cluster evaluation metric, and some background material on sense clustering in general.

- Chapter 5 is based on a series of talks given at the 26th International Society for Humor Studies Conference (ISHS 2014) in Utrecht, the International Humour Symposium of the 4th Hungarian Interdisciplinary Humour Conference in Komárno in 2014, and the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015) in Beijing. Two of these talks are accompanied by publications: “Automatic Disambiguation of English Puns” (Miller and Turković, 2016) was published in the ISHS-endorsed *European Journal of Humour Research*, and “Automatic Disambiguation of English Puns” (Miller and Gurevych, 2015) appeared in the proceedings of ACL-IJCNLP 2015. Mladen Turković was jointly responsible for the design of the Punnotator annotation system.
- Chapter 6 is based on “DKPro wsd: A Generalized UIMA-based Framework for Word Sense Disambiguation” (Miller, Erbs, *et al.*, 2013), a paper presented at the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013) in Sofia. The early designs of the DKPro wsd framework were joint work with Torsten Zesch, and later contributions supporting named entity linking and word sense induction (which are mentioned but not covered in great detail here) are entirely the work of Nicolai Erbs and Hans-Peter Zorn, respectively.
- Chapter 7 is based primarily on “Sense-annotating a Lexical Substitution Data Set with Ubyline” (Miller, Khemakhem, *et al.*, 2016), a paper to be presented at the 10th International Conference on Language Resources and Evaluation (LREC 2016) in Portorož. The design and implementation of the Ubyline annotation tool, which receives only a cursory treatment in this chapter, is mostly the work of Mohamed Khemakhem, with contributions by Richard Eckart de Castilho and the present author. This chapter also includes background material taken from “GermEval 2015: LexSub – A Shared Task for German-language Lexical Substitution” (Miller, Benikova, *et al.*, 2015), the overview paper of a workshop, co-chaired by the present author, which was held at the 26th International Conference of the German Society for Computational Linguistics and Language Technology (GSCL 2015) in Essen.

1.5 NOTATIONAL CONVENTIONS

As an aid for the reader and to facilitate cross-referencing by the author, all linguistic examples and mathematical formulas in this thesis bear unique identifiers. Linguistic examples are set on their own lines, and labelled on the left with sequential numbers. For example:

(4) Der Fluß *fließt* durch den Wald.

Formulas are numbered sequentially within chapters on the right, as in Formula 1.1 below.

The mathematical expressions in this thesis use standard notation which should be familiar to most readers. A possible exception is our use of the Iverson bracket (Iverson, 1962; Knuth, 1992) for notating conditional expressions:

$$[p] = \begin{cases} 1 & \text{if } p \text{ is true;} \\ 0 & \text{otherwise.} \end{cases} \quad (1.1)$$

2.1 TASK DESCRIPTION

Word sense disambiguation (wSD) is the task of determining the meaning a word carries in a particular context. In computational wSD setups, these meanings are usually references to entries in a predefined *sense inventory*—that is, a dictionary, thesaurus, or other resource that enumerates a set of senses for the words of a particular language. More formally, a sense inventory provides a lexicon of words L , a set of senses S , and a set-valued mapping $I : L \rightarrow \mathcal{P}_{\geq 1}(S)$ that associates every word $w \in L$ with a non-empty set of senses $I(w) \subseteq S$. In word sense disambiguation, we are given some sequence of words $T = (w_1, w_2, \dots, w_n)$ and tasked with finding some set-valued mapping $D : \mathbb{N} \rightarrow \mathcal{P}_{\geq 1}(S)$ such that, for some or all of the n words in T , $D(i) \subseteq I(w_i)$ and $D(i)$ is the subset of senses that best fit the context T .

Though computational wSD systems differ considerably in their approaches, on a high level they all work by extracting linguistic information from the target word's context and comparing it with the linguistic knowledge provided by the sense inventory and possibly other knowledge sources. Contextual clues may be identified by means of standard natural language processing software, including segmenters, tokenizers, stemmers, lemmatizers, part-of-speech taggers, and syntactic parsers; knowledge sources consulted can range from pre-existing dictionaries to language models automatically constructed from raw or manually annotated text corpora.

To illustrate a typical disambiguation problem, consider the following sentence:

(10) A vár már elfoglalta az ellenség.

Here we have chosen a text in Hungarian—a language with which most readers will be unfamiliar—to better demonstrate the challenges faced by machines with no understanding of language. Say the task is to disambiguate the word *vár* with respect to a digitized dictionary (Czuczor and Fogarasi, 1874) which enumerates its three senses as follows:

- s_1 (*verb*) Azon okból, hogy valamely idő lefolyta alatt bizonyos tárgyra vonatkozó kíváncsisága teljesedni, vagyis valami történni fog, kedélyét függőben tartja, illetőleg az esendőség bekövetkezteig bizonyos időt mulaszt, halaszt, elfolytenged.

s_2 (*noun*) Magasabb helyre, nevezetesen hegyoromra rakott, s erődített építmény, vagy több épületből álló védhely, az ellenség megtámadásai ellen.

s_3 (*noun*) Puszta Sopron.

Formally, then, we have $T = (w_1, w_2, w_3, w_4, w_5, w_6) = (A, \text{vár}, \text{már}, \text{elfoglalta}, \text{az}, \text{ellenség})$ and $I(\text{vár}) = \{s_1, s_2, s_3\}$, and the goal is to find the correct sense assignment for w_2 —*i. e.*, $D(2) \subseteq I(\text{vár})$.

Many WSD systems start by running the text through a tokenizer to split it into individual words, and a part-of-speech tagger to identify the word classes. (High-accuracy tokenizers and taggers are widely available for many languages.) The result might be the following:

(11) ($a_{\text{determiner}}$, vár_{noun} , $\text{már}_{\text{adverb}}$, $\text{elfoglalta}_{\text{verb}}$, $\text{az}_{\text{determiner}}$, $\text{ellenség}_{\text{noun}}$, $\cdot_{\text{punctuation}}$)

The output shows that in this context, *vár* is used as a noun. Already this is enough information to discount the candidate sense s_1 , which the dictionary lists as a verb sense.

Further analysis would then be required to determine whether the correct sense assignment is s_2 , s_3 , or perhaps both. A hypothetical disambiguator without access to any further NLP tools or knowledge sources might be programmed to consider the textual similarity between the context and the two definitions. For example, it may observe that, apart from the determiner *az*, the definition for sense s_3 shares no words in common with the context, whereas s_2 shares the content word *ellenség*. The disambiguator might therefore choose s_2 as the only “correct” sense: $D(2) = \{s_2\}$.

2.2 KNOWLEDGE SOURCES

All WSD systems, regardless of the approach they take, make use of knowledge present in the context and in external resources (Agirre and Martinez, 2001; Agirre and Stevenson, 2007). In the following subsections we briefly survey the types of linguistic phenomena useful for resolving ambiguity and the external resources which encode them.

2.2.1 External knowledge sources

External knowledge sources consist of information produced independently of the context and target word to be disambiguated. They may be highly structured, such as databases and ontologies, relatively unstructured, such as prose documents or word lists, or somewhere inbetween. The most frequently encountered external knowledge sources in WSD are as follows:

MACHINE-READABLE DICTIONARIES (MRDs) are electronic versions of traditional paper dictionaries. Most dictionaries provide at minimum the word's part(s) of speech and sense definitions (also called *glosses*). Some will also provide the pronunciation, morphology, etymology, valency, domain, register, derivations, semantically related terms, example sentences, and/or other information for individual words or senses. Among wsd researchers, digitized versions of print dictionaries have largely given way to wordnets, though there has recently been interest in the collaboratively constructed dictionary, Wiktionary (Meyer, 2013).

THESAURI are lexical reference works which group words according to semantic relations—usually synonymy but sometimes also antonymy, and rarely other relations. Thesauri used in early NLP research were rudimentary digitized versions of antiquated editions of *Roget's* (1852); nowadays more modern offerings are available (Jarmasz, 2003; Naber, 2005).

WORDNETS are networks that relate concepts and their lexicalizations via a taxonomy of lexical and semantic relations. Of these, the best known is the Princeton WordNet (Fellbaum, 1998), which is described in more detail below. Wordnets can provide any or all of the same information as dictionaries and thesauri; what sets them apart is their well-defined structure as directed or undirected graphs.

ENCYCLOPEDIAS are similar in some respects to dictionaries, providing lengthy prose descriptions but comparatively little other linguistic information. Coverage tends to focus on nouns and named entities; entries for other parts of speech are rare or absent. Collaboratively constructed encyclopedias such as Wikipedia may also have some characteristics of semantic networks (Mihalcea, 2006; Zesch, Gurevych, *et al.*, 2007).

SENSE-TAGGED CORPORA are texts in which some or all of the words have been manually annotated with sense labels from a particular inventory. These include SemCor (Miller, Leacock, *et al.*, 1993), Open Mind Word Expert (Mihalcea and Chklovski, 2003), and the various SENSEVAL/SemEval data sets described in §2.4.4. The principal use of sense-tagged texts is as training data for supervised wsd systems.

RAW CORPORA are large collections of documents that lack manual sense annotations, though some contain manually or automatically applied annotations of other types. Some raw corpora are expert-curated, multi-million-word general collections such as the British (BNC) and American National Corpus (Ide and Suderman, 2004; Burnard, 2007); others, such as wacky (Baroni *et al.*, 2009), are automatically harvested from the World Wide

Web, Usenet, or other Internet sources. Raw and automatically tagged text can be used to produce custom collocation resources (see below) or as data for a bootstrapping or word sense induction system (see §§2.3.2 and 2.3.3).

COLLOCATION RESOURCES provide ready-made statistics on the frequency and co-occurrences of words, and sometimes also on their types and grammatical relations. Examples of popular pre-built collocation resources include the Web 1T corpus (Brants and Franz, 2006) and the concordances distributed with the aforementioned BNC and Wacky corpora. *Word embeddings* are a family of collocation resource types in which word frequencies are mapped to vectors of real numbers in a low-dimensional space.

2.2.1.1 WordNet

WordNet (Fellbaum, 1998) is a lexical-semantic network relating English words and concepts, developed and maintained at Princeton University. Its fundamental advantage over traditional dictionaries and thesauri is its basic unit of organization, the *synset*, a set of word forms expressing the same meaning. Words with more than one distinct meaning are represented in as many distinct synsets. Each synset is linked to other synsets by means of semantic relations such as hypernymy/hyponymy (kind-of or “is-a” relations), holonymy/meronymy (part-whole relations). Some of these relations form taxonomies terminating at a root node.

Besides the word forms (*synonyms*) and semantic relations, WordNet’s synsets are characterized by a part of speech, a unique numeric identifier within that part of speech, a gloss, and usually one or more example sentences. Each pairing of a word form and synset is unique, and is identified by means of a *sense key*.

Owing largely to its wide coverage and liberal licensing terms, WordNet is the *de facto* standard sense inventory for use with English WSD. The current version, 3.1, contains 117 791 synsets and 207 235 senses. It has inspired similar projects in other languages, the most notable for the present thesis being the German-language GermaNet (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010).

2.2.2 Contextual knowledge

The *context* of a target word consists of the surrounding text, possibly up to and including the entire document. Relevant linguistic phenomena contained in the context are commonly identified by preprocessing it with automated NLP tools. A typical preprocessing pipeline will contain at least some of the following steps:

TOKENIZATION splits the text into a sequence of *tokens*—usually the individual lexical items to be disambiguated. Depending on the

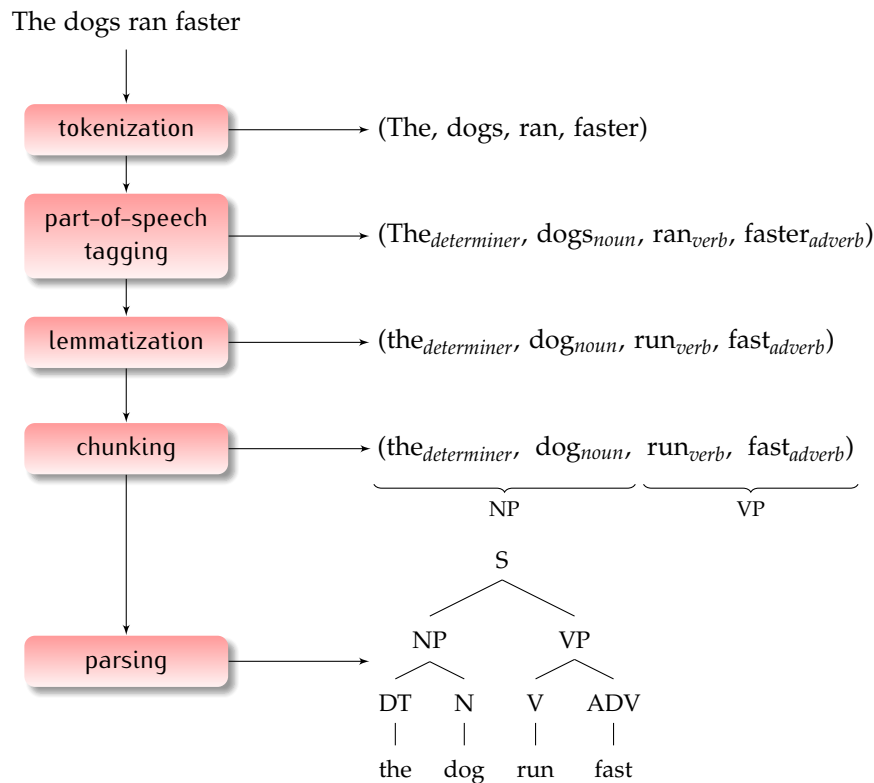


Figure 2. Sample text preprocessing pipeline for wsd

language and the framing of the wsd task, tokenization can be as simple as partitioning the input on the basis of white space and punctuation; more sophisticated approaches may involve splitting compound words or grouping multiword lexemes into single lexical units.

SEGMENTATION identifies discourse-level units within the text, such as sentences and paragraphs. Segmentation can be useful for limiting the size of the context according to well-defined semantic units.

PART-OF-SPEECH TAGGING assigns a grammatical category such as “noun” or “verb” to each word. Most wsd systems use part-of-speech information to narrow down the list of candidate senses; pos tags may also be a prerequisite for downstream preprocessing steps, or as features for machine learning algorithms.

STEMMING OR LEMMATIZATION involve normalizing the text’s word forms. In lemmatization, words are reduced to their uninflected base forms. For example, in English, adjectives in the comparative and superlative forms would be normalized to the positive (*bigger* and *biggest* to *big*), verb participles to the bare infinitive (*eating* and *eaten* to *eat*), and so on. Stemming is a cruder approach that simply removes inflectional affixes; it does not han-

dle irregular forms, and the “base” forms it produces may not be morphologically correct.

CHUNKING breaks up the text’s sentences into coarse-grained, syntactically correlated parts, such as “noun phrase” or “verb phrase”.

PARSING more precisely annotates the syntactic relations within sentences, usually by mapping them to tree structures.

A sample pipeline with some of these steps is illustrated in Figure 2.

2.3 APPROACHES TO DISAMBIGUATION

Word sense disambiguation techniques are commonly categorized according to whether they use machine learning approaches and manually sense-annotated data. The following subsections introduce these categories, so as to better situate our own work in the wider context of WSD.

2.3.1 Knowledge-based

Knowledge-based approaches to word sense disambiguation (Mihalcea, 2007) employ neither machine learning nor data which has been manually sense-annotated.¹ Instead, they rely exclusively on linguistic information gleaned from the target word’s context, from raw corpora, and/or from pre-built lexical-semantic resources. Such knowledge sources are relatively common, at least in comparison to sense-tagged data; these days even many poorly resourced languages have at least a machine-readable dictionary. Though this makes knowledge-based techniques more widely applicable, they do have the drawback of being less accurate than their supervised brethren. As we relate in §2.4.4, accuracy scores in organized evaluation campaigns have typically been 11 to 68% higher for supervised systems.

There is considerable variation in the techniques that knowledge-based disambiguation systems employ. Among the most popular are the following:

SELECTIONAL PREFERENCES. Some of the earliest WSD systems relied on selectional preferences and restrictions—that is, semantic constraints on word combinations. For example, one verb sense of the word *play* expects its subject to be an animate entity and its object to be a musical instrument; if neither of these are present in the context of a given occurrence of *play*, then that

¹ In some (usually older) literature, the term “unsupervised” is used for these techniques. Nowadays that term usually refers to the sort of word sense induction systems described in §2.3.3.

sense can be discounted. These preferences, encapsulating commonsense knowledge about classes of concepts, are provided by some sense inventories, but they can also be learned automatically from raw corpora (Agirre and Martínez, 2001). Though using selectional preferences presents an intuitive approach to WSD, it is not one that has historically achieved high accuracy.

GLOSS OVERLAPS. Gloss overlap techniques operate on the assumption that terms appearing in a target's correct definition are also likely to occur in its context or in the appropriate definitions of nearby words, whereas terms in the other sense definitions of the target are not likely to co-occur. This idea was first advanced by Lesk (1986), for whom the family of algorithms implementing it has come to be named.

In general, Lesk-like algorithms require nothing more than the tokenized context and sense definitions provided by a dictionary or other LSR, though they can benefit from additional preprocessing such as POS tagging and lemmatization. They work by looking up each sense of the target word in the LSR and comparing its definition with the context, or with the definitions of the other words in the context. The target word sense whose definition has the most words in common is selected as the correct sense.

Despite its simplicity, the original Lesk algorithm performs surprisingly well, and for this reason has spawned a number of variants and extensions, including one of our own. These are all described in further detail in Chapter 3.

SEMANTIC RELATEDNESS. Counting the lexical overlaps of glosses is one way of measuring the relatedness of two senses. The use of a wordnet as the sense inventory permits the use of a number of further, graph- or tree-based similarity measures. For example, Leacock and Chodorow (1998) define a measure based on the length of the shortest path between the two senses, normalized to the depth of the taxonomy. Resnik (1995), on the other hand, measures similarity by calculating the information content of the two senses' lowest superordinate in the taxonomy. Other notable similarity measures that have been used for WSD include those by Wu and Palmer (1994), Agirre and Rigau (1996), Jiang and Conrath (1997), Lin (1998a), Hirst and St-Onge (1998), and Mihalcea and Moldovan (1997).

The above measures all define some function $\text{sim}(s_1, s_2)$ returning a value between 0 and 1, representing total dissimilarity or similarity, respectively, of the senses s_1 and s_2 . The measures can be applied to the task of WSD as follows: Take the first candidate sense of the target and compute its similarity with each candidate sense of each word in the surrounding context. Sum

the maximum similarity value obtained for each context word. Repeat this process for the remaining candidate senses of the target. Select whichever candidate sense has the highest similarity sum over the context words. More formally,

$$D(i) = \arg \max_{s \in I(w_i)} \sum_{w_j \in T: w_j \neq w_i} \max_{s' \in I(w_j)} \text{sim}(s, s'). \quad (2.1)$$

The above-cited semantic similarity measures have been tested in various comparative evaluations (Pedersen *et al.*, 2005; Budanitsky and Hirst, 2006; Henrich, 2015); which measure performs best varies according to the particular task and evaluation metrics.

GRAPH CONNECTIVITY. There exist a number of approaches to disambiguation which, while graph-based, do not operate by directly measuring the similarity of sense pairs. Rather, they build a graph representing the context and use it to disambiguate all its words simultaneously. Many of these approaches are inspired by *lexical chains* (Morris and Hirst, 1991)—that is, sequences of semantically related words which appear in a text, independently of its grammatical structure.

An early algorithm based on graph connectivity is that of Mihalcea (2005), which builds a graph with vertices representing all possible senses of words in a context, and edges representing the semantic relations between them. A ranking algorithm such as PageRank (Brin and Page, 1998) is then applied to automatically select the most likely sense of each word. More recently, Navigli and Lapata (2010) describe a family of algorithms which, like Mihalcea’s, begin with a graph representing the sense candidates of the context. They then iteratively expand the graph by copying semantically related senses from WordNet, stopping only when they have reached all vertices reachable from the original source vertices. Each word in the context is then disambiguated according to which of its sense vertices best match some graph-based criteria, such as degree or eigenvector centrality. In some configurations, the performance of these algorithms can exceed even that of supervised systems (Ponzetto and Navigli, 2010).

In addition to the above general techniques, knowledge-based wsd systems employ various heuristics to resolve otherwise undecidable cases. One such heuristic, *one sense per discourse* (Gale *et al.*, 1992), is based on the observation that a word tends to be used in the same sense throughout a given discourse. So if, among those occurrences of a given word in a document that can be automatically disambiguated with high confidence, a plurality are tagged with a certain sense, then

it is highly likely that those occurrences which could not be tagged with high confidence are used in the same sense. A related heuristic, *one sense per collocation* (Yarowsky, 1993), holds that a word tends to preserve its meaning in similar contexts. For example, if a wsd system is reasonably certain to have correctly tagged a word in a particular sentence, then any subsequent occurrences in similar sentences (say, sharing some of the same content words) are probably assignable to the same sense.

Sense frequency is sometimes also used as a heuristic, though as it presupposes the existence of a hand-crafted semantic concordance (see §2.4.3), any system relying on it is more properly categorized as supervised.

2.3.2 Supervised

Supervised approaches to word sense disambiguation (Màrquez *et al.*, 2007) are those which make use of manually sense-annotated data, typically in the context of a machine learning setup. Here wsd is framed as a *classification problem*, where a *class* (word sense) must be predicted for each new *observation* (target word). The prediction is carried out by an algorithm known as a *classifier*, which has been previously *trained* on class assignments known to be correct. Because the set of possible classes varies with the lemma, wsd classifiers are usually trained and applied separately for each lemma.

Which linguistic features are used to train and apply the classifier varies from implementation to implementation. All of the contextual phenomena described in §2.2.2 have been used at one point or another, and several studies have investigated their respective impacts in certain scenarios (*e.g.*, Hoste *et al.*, 2002; Lee and Ng, 2002; Yarowsky and Florian, 2002). Likewise, there are a variety of classifier types which have been used in wsd; these include decision lists, tables, and trees; Bayesian and neural networks; k-nearest neighbour (knn) classifiers; maximum entropy classifiers; and support vector machines (svms). Each type has its own advantages and disadvantages, though recent studies point to svms and instance-based classifiers such as knn as the best algorithms accuracy-wise.

Supervised wsd systems are prized for their high precision, though this comes at the cost of recall: classifiers generally cannot disambiguate words not present in the training data, nor can they tag known words with previously unseen senses. Furthermore, the construction of manually tagged training examples is a phenomenally expensive process. Ng (1997) suggests that 500 manually tagged examples per word are required for high-precision wsd; based on typical annotation rates, Mihalcea and Chklovski (2003) estimate that it would take upwards of 80 person-years to produce enough training data for all the ambiguous words of the English language alone.

There are various ways in which researchers have tried to mitigate the knowledge acquisition bottleneck faced by supervised systems. In *semi-supervised* and *minimally supervised* wsd, the amount of training data is increased by automatically acquiring new annotations from untagged corpora. In *bootstrapping*, for example, a classifier is trained on a small number of examples and then applied to disambiguate further instances from an unlabelled corpus. Any examples disambiguated with high confidence are added to the set of training data. The process is then repeated, either using the same classifier or a new one, for a certain number of iterations or until some other conditions are met. Other approaches to minimally supervised wsd exploit *monosemous relatives*—words which are synonymous with a given polysemous target word but which have only one sense. A large untagged corpus is queried for contexts containing the monosemous relative; these contexts are then counted as training data for the original polysemous term (Gonzalo and Verdejo, 2007).

2.3.3 Unsupervised

Unsupervised approaches to disambiguation (Pedersen, 2007) apply machine learning but do not make use of manually sense-annotated data. Rather than framing wsd as a classification task, unsupervised systems learn to *cluster* word occurrences according to their meanings. Unlike supervised and knowledge-based techniques, they do not apply sense labels from a predefined sense inventory, but rather induce sense distinctions of their own. For this reason, they are perhaps better referred to as *word sense induction* (wsi) or *word sense discrimination* rather than word sense disambiguation systems.

The main benefit of unsupervised approaches is their ability to make wide-coverage, high-accuracy sense distinctions without the need for expensive manual training data. Whether their inability to apply labels from an existing LSR is an advantage or a disadvantage depends on whether the intended downstream application prescribes a particular sense inventory. In any case, the lack of a reference inventory makes wsi systems difficult to evaluate intrinsically; no two word sense induction systems are likely to induce the same clustering, though each clustering may be equally valid in consideration of its granularity and other factors.

2.4 EVALUATION METHODOLOGIES

Evaluation methodologies for wsd systems fall into two categories. The first of these, known variously as *extrinsic*, *in vivo*, or *end-to-end* evaluations, measure the system's contribution to the overall performance of some wider application, such as machine translation. The

second category is *intrinsic* or *in vitro* evaluations, where systems are tested as standalone applications using specially constructed benchmarks.

Because they are performed in the context of a particular real-world application, extrinsic evaluation methods are viewed as a more appropriate assessment of a system's ultimate utility (Edmonds and Kilgariff, 2002a). While isolated studies have demonstrated the usefulness of wsd in machine translation (*e.g.*, Vickrey *et al.*, 2005; Carpuat and Wu, 2007; Chan *et al.*, 2007) and information retrieval (Schütze and Pedersen, 1995; Stokoe *et al.*, 2003; Zhong and Ng, 2012), standardized extrinsic evaluations of wsd systems have been rare. Some progress towards such standardization can be seen in the SENSEVAL and SemEval shared tasks for translation (Kurohashi, 2001; Chklovski, Mihalcea, *et al.*, 2004), lexical substitution (McCarthy and Navigli, 2009), and search result clustering (Navigli and Vannella, 2013). However, only the last of these tasks simulated a fully end-to-end NLP application.

In intrinsic evaluation, with which the remainder of this section is concerned, the sense annotations applied by automated systems are directly compared with annotations applied by humans on the same data. Intrinsic evaluations are popular because the frameworks are easy to define and implement (Palmer, Ng, *et al.*, 2007). However, they do have a number of drawbacks. For one, they are tied to a particular sense inventory; a set of manually applied sense labels from one inventory cannot be used to evaluate wsd systems that use a different inventory. Intrinsic evaluations have also been criticized as being artificial in that their findings may not correlate with system performance in real-world tasks (Ide and Véronis, 1998).

Intrinsic evaluation can be used for two variants of the disambiguation task. In *all-words* disambiguation, systems are expected to provide a sense annotation for every word in a given text, or at least for every content word. In the other variant, *lexical sample*, systems are given a fixed set of lemmas and tasked with disambiguating all their occurrences in a document or collection of short texts.

All-words disambiguation is the more demanding task to evaluate, as it requires a broad-coverage sense inventory and considerable effort to produce the manually annotated data set. It is also harder to apply supervised wsd methods to all-words scenarios, which require sufficient numbers of manually annotated examples for each lemma-sense pairing. However, all-words is a more natural task, as the distributions of target words and senses resemble those found in real-world texts.

The lexical sample task, by contrast, allows for greater flexibility in the choice of sense inventory, which need cover only those few target lemmas appearing in the data set. It is also potentially easier to produce test data for, since all instances of a given lemma can be

tagged at once (a process known as *targetted tagging*), rather than having annotators move sequentially from one word to the next. Because lexical sample data sets usually contain a greater minimum number of instances per lemma, they are particularly suitable for use with supervised disambiguation systems.

2.4.1 Data sets

A prerequisite for intrinsic evaluation is a *data set*—i. e., a body of natural text in which human annotators have applied sense *labels* (also *tags* or *annotations*) to the words. This text may be a single, long document, a large collection of isolated sentences, or anything inbetween; the point is to have a sufficient number of tagged word *occurrences* (also known as *items*, *instances*, *targets*, or *examples*) to permit a statistically meaningful evaluation. The selection of texts and lemmas to annotate is determined largely by the use-case for the systems under evaluation. For the lexical sample task, it is common to select lemmas in such a way as to ensure a particular distribution across part of speech, word frequency, polysemy, domain, or other characteristics of interest. A uniform distribution can facilitate post-hoc analysis of system performance with respect to the different characteristics, whereas following the distributions of natural text could substitute for an all-words data set which would be otherwise too expensive to produce.

The sense labels that human annotators apply to the items of the data set are cross-references to the meanings listed in a particular sense inventory. Ideally, only one sense label is applied to each item, though many data sets permit annotators to apply multiple sense labels for cases where the distinction in meaning is unclear or unimportant. Some data sets also provide special sense labels to mark instances whose meaning is not provided by the sense inventory. Evaluation frameworks vary on how to deal with these “unassignable” labels. Some treat them as ordinary sense labels which disambiguation systems are free to apply themselves; others simply exclude unassignable instances from the scoring process.

2.4.1.1 Interannotator agreement

Quantitative analysis is easier when dealing with categories that are firmly delineated. However, word sense distinctions are not always clear, and human annotators can disagree on the appropriate sense assignment for a given instance. For this reason, it is usual to have each instance in a data set independently tagged by at least two annotators. The degree to which two or more humans agree on their annotations is known as *intertagger* or *interannotator agreement* (ITA or IAA, respectively).

Measuring IAA is important in the early stages of data set construction in order to identify and correct problems in the annotation guidelines. It can also be used to identify unreliable annotators, a problem which is particularly common when crowdsourcing annotations. Post-construction, IAA gives an idea of the overall trustworthiness of the data set. Just as importantly, though, it serves as a notional upper bound on system performance, since the consistency of automated systems cannot be expected to exceed that of humans (Palmer, Ng, *et al.*, 2007; Navigli, 2009).

There are a number of different ways of calculating IAA between two annotators (Artstein and Poesio, 2008). The most basic is known as *raw*, *observed*, or *percentage agreement*; it is the proportion of instances for which both annotators agreed. The calculation is clear enough for data sets where there is only one sense tag per item, but for those which permit instances to carry multiple sense labels, it is necessary to adopt a strategy which can deal with the case of partial matches. Three such strategies have seen common use (Véronis, 1998):

MINIMUM AGREEMENT counts an item as agreed when the annotators assign exactly the same senses. That is, for two annotators X and Y ,

$$A_{\min} = \frac{1}{n} \sum_{i=1}^n [D_X(i) = D_Y(i)]. \quad (2.2)$$

MAXIMUM AGREEMENT counts an item as agreed when the annotators assign at least one of the same senses:

$$A_{\max} = \frac{1}{n} \sum_{i=1}^n [D_X(i) \cap D_Y(i) \neq \emptyset]. \quad (2.3)$$

MEAN DICE AGREEMENT aims to account for partial agreement using the Dice (1945) coefficient:

$$A_{\text{Dice}} = \frac{2}{n} \sum_{i=1}^n \frac{|D_X(i) \cap D_Y(i)|}{|D_X(i)| + |D_Y(i)|}. \quad (2.4)$$

Regardless of the strategy used, percentage agreement is commonly denoted A_o , and ranges from 0 (no agreement, or systematic disagreement) to 1 (complete agreement). For the case where there are more than two annotators, one can calculate *average pairwise percentage agreement*—that is, the mean agreement among all possible pairs of annotators for every occurrence—though this can obscure important patterns of disagreement (Krippendorff, 2004).

Though percentage agreement is reported in many papers, it is not a measure which is comparable across studies as it does not account for agreement due to chance (Artstein and Poesio, 2008). For example, consider two studies which annotate the same occurrences of the same lemma, but using different dictionaries. One dictionary is coarse-grained and lists only two senses for the lemma; the other is more comprehensive, listing five senses. In this scenario, the study using the coarse-grained dictionary will have a higher percentage agreement due to chance: even if the annotators in both studies selected senses completely randomly, those using the concise dictionary would have an expected agreement of $1/2$ versus the other group's $1/5$.

Chance agreement is a factor even in standalone studies, because the sense distribution for any given lemma is rarely uniform. To reuse part of the previous example, consider an annotation study using the coarse-grained dictionary with only two senses for the lemma. If it is known that 97% of the occurrences of the lemma use the first sense, and only 3% use the second, then we would expect two annotators to agree on $(97\% \times 97\% + 3\% \times 3\%) = 94.18\%$ of the occurrences. A seemingly high percentage agreement of 90%, then, is actually much lower than that expected by chance.

The problems associated with percentage agreement have led to the development of IAA measures which attempt to correct for chance. One of the best known and most widely used in computational linguistics is 's (1960) κ :

$$\kappa = \frac{A_o - A_e^\kappa}{1 - A_e^\kappa}, \quad (2.5)$$

where A_e^κ is the probability of the two annotators agreeing on any sense label. The probability of an annotator applying a given sense label is estimated by using the proportion of items they actually tagged with this label. Thus,

$$\begin{aligned} A_e^\kappa &= \sum_{s \in S} \Pr(s|X) \cdot \Pr(s|Y) \\ &\approx \sum_{s \in S} \frac{\sum_{i=1}^n [s \in D_X(i)]}{n} \frac{\sum_{i=1}^n [s \in D_Y(i)]}{n} \\ &\approx \frac{1}{n^2} \sum_{s \in S} \sum_{i=1}^n \sum_{j=1}^n [s \in D_X(i)][s \in D_Y(j)]. \end{aligned} \quad (2.6)$$

Cohen's κ provides a score in the intervals between 0 and ± 1 , where -1 signifies systematic disagreement, 1 signifies perfect agreement, and 0 is chance agreement.

Despite its popularity in WSD, κ has a number of serious drawbacks. For one, it is not applicable to data sets that permit multiple sense annotations per target. More seriously, the statistic has been sharply criticized for its tendency to treat annotator bias as agreement; Krippendorff (2004) goes so far as to call κ "worthless" as an indicator of reliability.

Researchers are increasingly recognizing 's (1980) α as a more reliable agreement statistic. Though conceptually and computationally difficult, it is versatile in its handling of multiple annotators, multiple sense labels per target, missing data, and small sample sizes. The statistic is computed as

$$\alpha = 1 - \frac{nm - 1}{m - 1} \cdot \frac{\sum_{i=1}^n \sum_{j \in S'} \sum_{k \in S'} c(i, j) \cdot c(i, k) \cdot \delta(j, k)}{\sum_{j \in S'} \sum_{k \in S'} e(j) \cdot e(k) \cdot \delta(j, k)} \quad (2.7)$$

where

- there are m annotators X_1, X_2, \dots, X_m ;
- S' is the set of all annotations (sense tags, or sets thereof if multiple tags per item are allowed) applied by the annotators:

$$S' = \bigcup_{i=1}^n \bigcup_{j=1}^m D_{X_j}(i); \quad (2.8)$$

- $c(i, s)$ is the number of annotators who applied annotation s to item i :

$$c(i, s) = \sum_{j=1}^m [D_{X_j}(i) = s]; \quad (2.9)$$

- $e(s)$ is the total number of items with annotation s :

$$e(s) = \sum_{i=1}^n \left[s \in \bigcup_{j=1}^m D_{X_j}(i) \right]; \quad (2.10)$$

and

- $\delta(s_i, s_j)$ is some distance function comparing two annotations s_i and s_j and returning a value between 0 (if the two annotations are equal) and 1 (if they are maximally different).

The α metric ranges in $(-1, 1]$, where 1 indicates perfect agreement, -1 perfect disagreement, and 0 is the expected score for random labelling. Krippendorff (1980, p. 147) recommends requiring $\alpha \geq 0.800$ as an acceptable level of agreement, or where tentative conclusions are still permissible, $\alpha \geq 0.667$.

For data sets where annotations consist of singleton sense labels, α 's distance function can be a simple

$$\delta(s_i, s_j) = [s_i \neq s_j]. \quad (2.11)$$

For scenarios which permit multiple sense tags per item, the MASI ("Measuring Agreement on Set-valued Items") set comparison metric can be used (Passonneau, 2006). This is defined as

$$\delta_{\text{MASI}}(s_i, s_j) = 1 - J(s_i, s_j) \cdot M(s_i, s_j) \quad (2.12)$$

where $J(s_i, s_j)$ is the Jaccard (1901) index:

$$J(s_i, s_j) = \frac{|s_i \cap s_j|}{|s_i \cup s_j|} \quad (2.13)$$

and $M(s_i, s_j)$ is a “monotonicity” factor defined as follows:

$$M(s_i, s_j) = \begin{cases} 1 & \text{if } s_i = s_j; \\ 2/3 & \text{if } s_i \subset s_j \text{ or } s_j \subset s_i; \\ 1/3 & \text{if } s_i \cap s_j \neq \emptyset; \\ 0 & \text{otherwise.} \end{cases} \quad (2.14)$$

2.4.2 Evaluation metrics

Once a data set has been produced, the manually applied “gold standard” sense annotations are compared, one at a time, with the corresponding system annotations. The aggregate comparisons are then expressed in terms of evaluation metrics adapted from the field of information retrieval. The metrics described below are described by, *inter alia*, Navigli (2009) and Palmer, Ng, *et al.* (2007).

The simplest method of comparing individual items is the *exact match* criterion which, as with the minimum agreement strategy for IAA (see Formula 2.2), awards a score of 1 when the two annotations are exactly equal and 0 when they are not. In many task setups, however, systems are permitted to guess multiple annotations for a given instance, assigning a *confidence* or *probability* score to each guess, with the sum of probabilities not exceeding 1. In this case, the system score is the sum of probabilities it assigned to the correct sense tags. So for the gold standard G and the system Y , the score for item i is

$$\text{score}(i) = \sum_{s \in D_Y(i) \cap D_G(i)} \Pr_Y(s|i). \quad (2.15)$$

That is, the multiple tags are scored disjunctively.

A disambiguation system may choose to tag all the instances in a data set, or only a subset. Its *coverage* is the proportion of instances for which it produced an annotation:

$$C = \frac{1}{n} \sum_{i=1}^n [D(i) \neq \emptyset]. \quad (2.16)$$

The *precision* of a system is the proportion of its sense assignments which are correct:

$$P = \frac{\sum_{i=1}^n \text{score}(i)}{\sum_{i=1}^n [D(i) \neq \emptyset]}. \quad (2.17)$$

Recall—also sometimes called *accuracy*—is the fraction of items in the entire data set correctly disambiguated by the system:

$$R = \frac{1}{n} \sum_{i=1}^n \text{score}(i). \quad (2.18)$$

Precision measures the quality of a system's sense disambiguations, whereas recall measures its robustness in correctly handling the broadest possible range of input. Whether one or the other metric is to be favoured depends on the application. For the general case, it is common to combine the two metrics as follows:

$$F_1 = \frac{2PR}{P + R}. \quad (2.19)$$

This weighted harmonic mean is known as the F_1 -measure or *F-score*. Note that, according to the above definitions, it is always the case that $P \geq R$, and $F_1 = P = R$ whenever $P = R$.

2.4.3 Lower and upper bounds

System performance for any natural language processing task is normally interpreted with reference to one or more baselines. In word sense disambiguation, systems are typically compared with two naïve disambiguators, the *random sense* baseline and the *most frequent sense* (MFS) baseline (Gale, 1992; Miller, Chodorow, *et al.*, 1994). These baselines serve as *lower bounds*, meaning that any reasonably sophisticated system ought to be able to exceed their performance.

The random sense baseline assumes that, for each item, the disambiguator randomly selects a single sense from the candidates. Rather than implementing such a disambiguator stochastically, it is more convenient and reproducible to calculate its accuracy computationally:

$$P_{\text{rand}} = R_{\text{rand}} = \frac{1}{n} \sum_{i=1}^n \frac{|D_G(w_i)|}{|I(w_i)|}. \quad (2.20)$$

The MFS baseline presupposes the existence of a large corpus in which all occurrences of words in the data set have been manually sense-tagged, and an index providing the frequency, or at least the ranking, of each sense. Many traditional dictionaries order their entries' word senses in descending order of frequency (Kipfer, 1984), and raw frequency counts are provided by some lexical-semantic resources, including WordNet. As its name suggests, MFS involves always selecting from the candidates that sense which has the highest frequency. The MFS baseline is notoriously difficult to beat, even for supervised disambiguation systems (Preiss, Dehdari, *et al.*, n.d.), and

since it relies on expensive sense-tagged data, it is not considered a benchmark for the performance of knowledge-based systems.

An *upper bound* specifies the best performance a system can be expected to attain. This is usually taken to be the data set's raw inter-annotator agreement (see §2.4.1.1). The value of this statistic varies from data set to data set; depending on the task and sense inventory used, it has been variously reckoned between 67 and 94% (see Table 1 in the following section). It has been argued that the artificiality of the sense annotation task, and possibly also the use of adjudicators to resolve annotator disagreements, renders IAA a misleading upper bound for WSD (Murray and Green, 2004; Edmonds, 2005). A more rigorous upper bound is *replicability*—the level of agreement between two replications of the same manual annotation task, including adjudication (Kilgarriff and Rosenzweig, 2000). However, this approach is seldom used as it greatly increases the already high cost of creating the data set.

2.4.4 Shared tasks

As discussed in the previous chapter, word sense disambiguation systems have proved difficult to compare owing to the myriad ways in which the disambiguation task has been formulated. In hopes of providing some common testbeds on which systems can be compared, WSD researchers have begun organizing evaluation campaigns with standardized task definitions, data sets, and sense inventories. The evaluation frameworks remain popular for assessing novel WSD systems. In this section, we briefly survey the past tasks and data sets relevant to our work. Though much of this work involves techniques that are language-independent, for the sake of convenience we have implemented and tested most of them using English-language resources; the focus below is therefore on monolingual English tasks. (A summary of the respective data sets can be found in Table 1.)

The first organized evaluation competition, SENSEVAL-1 (Kilgarriff and Rosenzweig, 2000), took place in 1998 and featured lexical sample tasks for English, French, and Italian. The data set for the English task consists of 21 578 instances of 35 lemmas with a training–test split of approximately 3:2. The instances were manually annotated with senses from Hector (Atkins, 1992), a custom dictionary produced by lexicographers at Oxford University Press. A total of 18 systems participated in the campaign; the best-performing achieved precision and recall of 77.1%.

SENSEVAL-2 (Edmonds and Kilgarriff, 2002a), which was held in 2001, hosted all-words and lexical sample disambiguation tasks in twelve different languages, plus an English–Japanese translation task. The English tasks used a pre-release version of WordNet 1.7, unfortu-

nately now lost.² The lexical sample task featured 21 systems competing on a data set of 12 939 instances of 73 lemmas, with a training–test split of about 2:1. The best-performing knowledge-based and supervised systems achieved accuracies of 40 and 64%, respectively. The all-words task had no training data; the test set consisted of 2 473 instances of over 1 082 unique lemmas. The best-performing of 27 participating system achieved recall of 67%.

The SENSEVAL-3 workshop (Mihalcea and Edmonds, 2004), which took place in 2004, broadened its scope beyond pure WSD to include similar tasks such as semantic role labelling. As at SENSEVAL-2, there were both all-words and lexical sample tasks for English; this time both used the official release of WordNet 1.7.1 (Mihalcea, Chklovski, and Kilgarriff, 2004; Snyder and Palmer, 2004). The all-words task had a test set of 2041 instances of over 960 lemmas, with the best-performing of 26 knowledge-based and supervised systems achieving accuracies of 58 and 65%, respectively. The lexical sample task used a data set of 11 804 instances of 57 lemmas with a 2:1 training–test split. There were 27 participating systems, the best of which achieved accuracies of 66% (knowledge-based) and 73% (supervised).

By 2007, the name of the campaign had changed to *SemEval* in order to better reflect the growing inclusion of tasks not directly related to word sense labelling. The first workshop under the new moniker, SemEval-2007 (Agirre, Màrquez, *et al.*, 2007), marked the first appearance of lexical substitution and word sense induction tasks (Agirre and Soroa, 2007; McCarthy and Navigli, 2007). Also held were coarse-grained variants of the usual all-words and lexical sample tasks for English, as well as a traditional fine-grained all-words task.

The fine-grained all-words task (Pradhan *et al.*, 2007)—somewhat misleadingly named, as not all content words were annotated—used a test set of 466 tagged instances of over 327 lemmas with senses from WordNet 2.1. Of the 14 participants, the best supervised system achieved an F-score of 59%, and the best knowledge-based system an F-score of 53%. For the coarse-grained variant (Navigli, Litkowski, *et al.*, 2007), annotators applied WordNet 2.1 annotations to 2269 instances of 1183 word types. All sense keys in WordNet were then semi-automatically clustered to form coarse-grained senses. Participating systems received credit for disambiguating a sense if they chose the annotator’s sense key or any other sense key in the same cluster. Of the fourteen participating systems, the top-performing knowledge-based and supervised ones achieved F-scores of 70% and 83%, respectively.

The coarse-grained lexical sample task at SemEval-2007 (Pradhan *et al.*, 2007) took a different approach than the all-words task. Rather than artificially coarsening WordNet senses, the organizers used the

² Personal communication with Martha Palmer, Christiane Fellbaum, *et al.*, 19 December 2011 through 10 January 2012.

inherently coarse-grained OntoNotes (Hovy *et al.*, 2006) as the sense inventory. The task’s data set was comprised of 11 280 instances of 65 verbs, plus 15 852 instances of 35 nouns, in a roughly 9:2 training–test split. Thirteen systems participated in the task, achieving maximum F-scores of 89% (supervised) and 54% (knowledge-based).

The SemEval-2010 workshop (Erk and Strapparava, 2010) saw the beginning of a marked shift away from traditional wsd tasks. The domain-specific all-words task (Agirre, López de Lacalle, *et al.*, 2010) was the event’s only wsd task that included an English data set. This consists of 8157 occurrences of 997 nouns and 635 verbs annotated with synset identifiers from WordNet 3.0. Among the 29 participating systems, the peak F-scores were 56% (supervised) and 50% (knowledge-based).

Since 2012, SemEval has become an annual evaluation campaign, though so far only the 2013 and 2015 events (Manandhar and Yuret, 2013; Nakov *et al.*, 2015) have featured tasks closely related to wsd. These include three cross- and multilingual wsd tasks, a monolingual Chinese task, a word sense induction task, and a new *in vivo* evaluation involving clustering of information retrieval results.

2.5 CHAPTER SUMMARY

In this chapter, we formally defined the task of word sense disambiguation. We surveyed the types of linguistic knowledge and resources systems rely on to solve it, with a particular focus on WordNet, a popular English-language semantic network. We gave high-level overview of approaches to wsd, including supervised methods, which require a collection of hand-annotated training examples, and knowledge-based techniques, which do not. Finally, we described how wsd systems are evaluated, from the construction and quality assessment of gold-standard data to the performance metrics and baselines on which wsd systems are measured.

TASK	INVENTORY	TRAINING		TEST		
		TYPES	TOKENS	TYPES	TOKENS	INTERANNOTATOR AGREEMENT
SENSEVAL-1 lexical sample	Hector	30	13 127	35	8 451	$A_o = 0.95^*$
SENSEVAL-2 all-words	WordNet 1.7-pre	—	—	> 1 082	2 473	$A_o = 0.75$
SENSEVAL-2 lexical sample	WordNet 1.7-pre	73	8 611	73	4 328	$A_o = 0.86$
SENSEVAL-3 all-words	WordNet 1.7.1	—	—	> 960	2 041	$A_o = 0.725$
SENSEVAL-3 lexical sample	WordNet 1.7.1	57	7 860	57	3 944	$A_o = 0.673$; $\kappa = 0.58$
SemEval-2007 fine-grained all-words	WordNet 2.1	—	—	> 327	466	$A_o = 0.86$ (nouns), 0.72 (verbs)
SemEval-2007 coarse-grained lexical sample	OntoNotes	100	22 281	100	4 851	$A_o > 0.90$
SemEval-2007 coarse-grained all-words	WordNet 2.1	—	—	1 183	2 269	$A_o \approx 0.938$
SemEval-2010 domain-specific all-words	WordNet 3.0	—	—	8 157	1 632	N/A (only one annotator)

* Based on replicability

Table 1. Data sets for monolingual English SENSEVAL and SemEval wsd tasks

3.1 MOTIVATION

Though supervised approaches to word sense disambiguation usually achieve impressive results, their accuracy comes at a high cost. For each sense of each lemma to be disambiguated, it is necessary to manually annotate a large number of examples to use as training data. Producing this training data is a particularly expensive process; it has been estimated that it would take decades to annotate enough examples to train a classifier to disambiguate all the polysemous words in English (Mihalcea and Chklovski, 2003). A number of approaches have been advanced for mitigating this knowledge acquisition bottleneck, including streamlining the annotation process via planning, modelling, and adoption of best practices (Fort *et al.*, 2012); the use of labour-saving annotation tools (Miller, Khemakhem, *et al.*, 2016); and the use of semi-supervised methods for automatically acquiring new training examples (see §2.3.2). However, none of these approaches fully solve the knowledge acquisition bottleneck; they can only reduce the amount of required training data or the time required to collect it.

In contrast to supervised methods are knowledge-based systems, which rely only on pre-existing knowledge sources such as machine-readable lexicons, semantic networks, and raw corpora. Knowledge-based approaches are therefore much cheaper to implement, and many of them can be readily adapted to disambiguate texts in different domains and languages. However, they have the drawback of being considerably less accurate than their supervised counterparts. This is commonly due to some variant of the *information gap* problem. Generally speaking, this means that the linguistic data provided by the approach's knowledge sources often has little or nothing in common with the linguistic data gleaned from context of the disambiguation target. Without such an overlap, systems may be forced to guess between minimally differentiated senses, or to simply avoid attempting disambiguation at all.

The information gap problem was first noticed with gloss overlap algorithms (see §2.3.1), which look for words shared between the target's definitions and contextual knowledge. For these algorithms, the problem is more specifically referred to as the *lexical gap*. Various solutions have been proposed for it, with varying degrees of success, though none have approached the accuracy of even the most basic supervised approaches.

In this chapter, we propose a new method for bridging the lexical gap which is based on statistics collected from a large, unannotated background corpus. Specifically, we enrich the textual information from the context and the sense inventory with lexical expansions produced by a distributional thesaurus. We examine the contribution of these expansions to two popular knowledge-based algorithms, including one which already tries to address the lexical gap through other means. We show that, especially in situations for which no sense frequency information is available, improvements from adding more knowledge and from adding lexical expansions add up, allowing us to improve over the state of the art for knowledge-based all-words disambiguation.

3.2 BACKGROUND AND RELATED WORK

3.2.1 The Lesk algorithm

The very earliest approaches to word sense disambiguation tended to use small, hand-crafted sense inventories geared towards the lexical sample task. The advent of full-scale machine-readable dictionaries in the 1980s led to the first systems capable of disambiguating all words in unrestricted text. Probably the first such attempt was that of Lesk (1986), whose system requires nothing more than the target words in context and a machine-readable dictionary.

The basic idea behind the original Lesk algorithm is that two words in the same context can be simultaneously disambiguated by looking up their respective definitions in a dictionary and finding the maximum overlap between each combination of their senses. More formally, say we have some context $T = (w_1, w_2, \dots, w_n)$ containing a list of words, and a dictionary $I : L \rightarrow \mathcal{P}_{\geq 1}(S)$ that associates each word $w \in L$ with a set of candidate senses $I(w) \subseteq S$. Furthermore, each sense $s \in S$ has an associated gloss $G(s) = (g_1, g_2, \dots, g_m)$, which like T is also a list of words. To disambiguate any pair of words w_i and w_j , it is sufficient to find

$$\text{lesk}(w_i, w_j) = \arg \max_{s_i \in I(w_i), s_j \in I(w_j)} |G(s_i) \cap G(s_j)|. \quad (3.1)$$

The method can be trivially adapted to simultaneously disambiguate any set of two or more context words $T' = \{t_1, t_2, \dots, t_k\} \subseteq T$:

$$\text{lesk}(T') = \arg \max_{s_{t_1} \in I(t_1), \dots, s_{t_k} \in I(t_k)} \sum_{i=1}^k \sum_{j=i+1}^k |G(s_{t_i}) \cap G(s_{t_j})|. \quad (3.2)$$

To get some idea of how the original Lesk algorithm operates, consider disambiguating the word *bat* in the following example:

(12) He hit the ball with the *bat*.

Say that the dictionary consulted by the algorithm contains entries for *hit*, *ball*, and *bat*, with the following definitions:

HIT

1. Cause movement by striking.
2. Affect or afflict suddenly, usually adversely.

BALL

1. Round object that is hit or thrown in sports.
2. Lavish dance requiring formal attire.

BAT

1. Small, nocturnal flying mammal.
2. Wooden club used to hit a ball in various sports.

Since Lesk needs at least two target words as input, let's select *ball* as the second target. The algorithm then compares the definitions for every possible combination of senses of the two words: ($ball_1$, bat_1), ($ball_1$, bat_2), ($ball_2$, bat_1), and ($ball_2$, bat_2). Of these, only $ball_1$ and bat_2 have any words in common (*hit*, *in*, and *sports*), so the algorithm selects these two senses. In this case, the selections happen to be correct.

Though conceptually simple, the Lesk algorithm performs surprisingly well. In some informal small-scale experiments, Lesk (1986) reported accuracy of 50 to 70%. The algorithm and its later refinements and extensions have therefore seen widespread adoption, either as baselines against which more sophisticated systems are compared, or as disambiguation systems in their own right. Their precise efficacy is hard to judge across implementations, however, owing to the vagueness of the original specification. The description given in Lesk (1986) leaves unspecified such important details as how multiple occurrences of the same word are counted, whether to lemmatize the glosses, whether to apply a stop list to remove non-content words, whether or how to normalize the overlap count with respect to the gloss lengths, *etc.* It is probably safe to say that no two implementations of the algorithm work in precisely the same way.

What all implementations of the original Lesk algorithm have in common, however, is their computational complexity. That is, when disambiguating running text, there is a combinatorial explosion in the number of sense glosses that need to be compared. The algorithm quickly becomes intractable as the size of the context increases. A popular Lesk variant that solves the original's intractability is known as *simplified Lesk* (Kilgarriff and Rosenzweig, 2000). This version disambiguates one word at a time by comparing each of its definitions to the context in which the word is found:

$$\text{simplified lesk}(w_i) = \arg \max_{s_i \in I(w_i)} |G(s_i) \cap T|. \quad (3.3)$$

Recall the example above, where we wish to disambiguate the word *bat* with the following context and definitions:

(12) He hit the ball with the *bat*.

BAT

1. Small, nocturnal flying mammal.
2. Wooden club used to hit a ball in various sports.

The simplified Lesk algorithm would look for words shared by the context and each of the two sense definitions. In this case, bat_1 has an overlap of 0, and bat_2 an overlap of 2 (*hit* and *ball*), so the latter would be correctly chosen.

Both the original and simplified Lesk algorithms are susceptible to the aforementioned lexical gap problem. We would have encountered this problem in the original Lesk example on the previous page if we had happened to choose *hit* instead of *ball* as the second disambiguation target. In this case, the algorithm would have found no overlapping words at all between the two sets of sense definitions, and would therefore have been unable to disambiguate either word. To avoid this problem, Lesk himself proposed increasing the size of the context window and disambiguating several words simultaneously, as in Formula 3.2. However, this triggers the intractability problem previously mentioned. Moreover, Vasilescu *et al.* (2004) later found that the Lesk algorithm's accuracy was generally better for smaller contexts.

3.2.1.1 Extending the Lesk algorithm

The dictionary Lesk used in his original study was the *Oxford Advanced Learner's Dictionary* which, in addition to glosses, provides example sentences for many senses. A further solution Lesk proposed for bridging the lexical gap was to include these example sentences along with the definitions for the purpose of counting lexical overlaps. That is, for a sense example sentence $E(s) = (e_1, e_2, \dots, e_l)$, the original and simplified Lesk algorithms would be modified as follows:

$$\text{ext. lesk}(w_i, w_j) = \arg \max_{s_i \in I(w_i), s_j \in I(w_j)} |(G(s_i) \cup E(s_i)) \cap (G(s_j) \cup E(s_j))|. \quad (3.4)$$

$$\text{simp. ext. lesk}(w_i) = \arg \max_{s_i \in I(w_i)} |(G(s_i) \cup E(s_i)) \cap T| \quad (3.5)$$

Kilgarriff and Rosenzweig (2000) found that including the example sentences (for simplified Lesk) led to significantly better performance than using the definitions alone.

A further refinement to the extended Lesk algorithms was proposed by Banerjee and Pedersen (2002). They observed that, where

there exists a lexical resource like WordNet which also provides semantic relations between senses, these can be used to augment definitions with those from related senses (such as hypernyms and hyponyms). This variant of the extended Lesk algorithm was found to be a great improvement over the original. Subsequent researchers (e.g., Ponzetto and Navigli, 2010) have combined the “simplified” and “extended” approaches into a “simplified extended” algorithm, in which augmented definitions are compared not with each other, but with the target word context.

To give an example, consider the problem of disambiguating the word *interest* in the following sentence:

(13) The loan *interest* is paid monthly.

Assume the disambiguation algorithm employs a semantic network which defines two senses of *interest* as follows:

INTEREST

1. Fixed charge for borrowing money.
2. Sense of concern with something.

The simplified Lesk algorithm cannot attempt a disambiguation in this case, since neither of the definitions has any words in common with the context. However, say the semantic network links each sense to its hyponyms:

INTEREST

1. Fixed charge for borrowing money.
 - ↳ **SIMPLE INTEREST** Interest paid on the principal alone.
 - ↳ **COMPOUND INTEREST** Interest paid on both the principal and accrued interest.
2. Sense of concern with something.
 - ↳ **ENTHUSIASM** Excited or intense positive attention.

Here the extended simplified Lesk algorithm succeeds, since there is now some lexical overlap (*paid*) between the surrounding context and the “extended” definition of *interest*₁.

3.2.2 Distributional similarity

The basic notion of distributional similarity, initially popularized by Firth (1957), is that words which tend to appear in similar contexts tend to have similar meanings. This principle has long been applied in word sense disambiguation, initially in the form of ‘s (1993) “one sense per collocation” heuristic. That is, once the sense of a polysemous word is known in a given context, subsequently observed occurrences of that word in a similar context (say, as the object of the

same verb) can be assumed to bear the same meaning. This heuristic has limited utility in knowledge-based wsd, however, because it requires a set of known collocations as seed data. The contextual cues used to constrain the word meanings must either be deduced from the definitions and examples provided by the sense inventory, or else remembered from previous high-confidence disambiguations on the input text. An early attempt at harnessing distributional information in a supervised setting is that of Tugwell and Kilgarrieff (2001). In their technique, “word sketches” (Kilgarrieff and Tugwell, 2001) consisting of common patterns of usage of a word were extracted from a large pos-tagged corpus and presented to a human operator for manual sense annotation. The pattern–sense associations were then used as input to a bootstrapping wsd algorithm employing Yarowsky’s technique.

Many more recent approaches to automatic wsd rely on distributional information to model the “topicality” of the context, sense definition, or vocabulary words. These include using latent semantic analysis (LSA), latent Dirichlet allocation (LDA), and other word embedding techniques (Gliozzo *et al.*, 2005; Cai *et al.*, 2007; Li *et al.*, 2010; Chen, Liu, *et al.*, 2014; Rothe and Schütze, 2015); collecting additional text material per sense as in topic signatures (Martínez *et al.*, 2008); and clustering for word sense induction as features (Agirre, Martínez, *et al.*, 2006; Biemann, 2013). The importance of bridging the lexical gap is reflected in all these recent advances, be it in knowledge-based or supervised wsd scenarios.

Distributional similarity has also found a number of applications outside of word sense disambiguation, including anaphora resolution, information retrieval, text simplification, *etc.* (Weeds, 2003). Of greatest relevance to our own work is its use in automatic generation of thesauri (Hindle, 1990; Grefenstette, 1994; Lin, 1998b; Curran and Moens, 2002). Recall from §2.2.1 that a thesaurus is a lexical reference work which groups words according to semantic relations, such as synonymy and antonymy. Traditional thesauri are hand-crafted by expert lexicographers, but the distributional hypothesis suggests that the process can be automated by looking for patterns of lexical collocations in a large corpus. That is, distributional similarity is used as a predictor of semantic similarity, or at least semantic relatedness. The distributional thesauri (DTs) produced by these automated techniques have been criticized for their inability to distinguish between synonyms, antonyms, hypernyms, *etc.*, as traditional thesauri do (Lin *et al.*, 2003; Weeds, 2003). As we report below, however, this does not appear to be a serious impediment to their use in wsd.

3.3 APPROACH

While distributional similarity in some form or another has been widely used in wsd, our own approach is the first to use distributional thesauri in particular. In our method, we use a dt to expand the lexical representations of the context and sense definition with additional, semantically related terms. On this expanded representation, we are able to apply the well-known overlap-based methods without any modification. Lexical expansion has already proven useful in semantic text similarity evaluations (Bär, Biemann, *et al.*, 2012), which is a task related to matching sense definitions to contexts.

The intuition behind our approach is depicted in Figure 3, which reproduces the *interest* sentence from Example 13. The correct sense definition from our sense inventory (“fixed charge for borrowing money”) has no words in common with the context, and thus would not be selected by a rudimentary overlap-based wsd algorithm such as simplified Lesk. But with the addition of ten lexical expansions per content word (shown in smaller text), we increase the number of overlapping word pairs (shown in boldface) to seven.

Observe also that this expansion of linear text sequences into a two-dimensional representation makes conceptual associations (*cf.* the associative relations of Saussure (1916)) explicit, allowing for purely symbolic matching rather than using a vector-space representation such as LSA. The main differences to vector-space approaches are the following: On the one hand, vector-space approaches usually use dimensionality reduction in order to handle sparsity, which results in a fixed number of topics/dimensions. While very salient collection-specific topics are handled well by this approach, rare topics are either conflated into a single rump topic, or distributed amongst the salient topics. Our dt-based expansion technique has no notion of dimensions since it works on the word level. Thus it does not suffer from this kind of sampling error that is inevitable when representing a large vocabulary with a small fixed number of dimensions or topics. On the other hand, while vector-space models do a good job at ranking candidates according to their similarity,¹ they fail to efficiently generate a top-ranked list of possible expansions: due to its size, it is infeasible to rank the full vocabulary every time. Lexical expansion methods based on distributional similarity, however, generate a short list of highly similar candidates.

The lexical expansions shown in Figure 3 were generated by the same dt used in our experiments. However, for the general case, we make no assumptions about the method that generates the lexical expansions, which could just as easily come from, say, translations via bridge languages, paraphrasing systems, or lexical substitution systems.

¹ See Rapp (2004) for an early success of vector-space models on a semantic task.

The	loan	<u>interest</u>	is	paid	monthly.
	mortgage			paying	annual
	loans			pay	weekly
	debt			pays	yearly
	financing			owed	quarterly
	mortgages			generated	hefty
	credit			invested	daily
	lease			spent	regular
	bond			collected	additional
	grant			raised	substantial
	funding			reimbursed	recent

INTEREST ₁	fixed	charge	for	borrowing	money
	solved	charges		spending	dollars
	hefty	counts		borrow	cash
	resolved	charging		lending	funds
	monthly	cost		borrowed	billions
	additional	conviction		debt	monies
	existing	allegation		investment	millions
	reduced	pay		raising	trillions
	done	suspicion		inflows	funding
	current	count		investing	resources
	substantial	part		borrowings	donations

Figure 3. Example showing the intuition behind lexical expansion for matching a context (top) to a sense definition (bottom). The term to be disambiguated is underlined and the matching terms are in boldface.

3.4 EXPERIMENTS

Our experiments measure the contribution of various lexical expansion schemes to the simplified and simplified extended variants of the Lesk algorithm. We chose these algorithms because of their simplicity and transparency, making it easy for us to trace through their operation and see exactly how and where the lexical expansions help or hinder disambiguation. Furthermore, Lesk variants perform remarkably well despite their simplicity, making them popular choices as baselines and as starting points for developing more sophisticated wsd algorithms.

Our experiments with the simplified Lesk algorithm use only the definitions provided by WordNet; they are intended to model the case where we have a generic MRD which provides sense definitions, but no additional lexical-semantic information such as example sentences or semantic relations. Such scenarios are typical of many languages and domains, where there is no WordNet-like resource and no manually sense-annotated corpus which could be used for supervised wsd or for a backoff to the Indexmost frequent sense baseline. Our hope is that by providing an accurate wsd systems that relies on the existence of an MRD only, we might pave the way to wider application of lexical disambiguation in NLP applications, particularly for less-resourced languages.

By contrast, the experiments with the simplified extended Lesk algorithm assume the existence of a WordNet-like resource with a taxonomic structure; the definition text for a sense is therefore constructed from the gloss, synonyms, and example sentences provided by WordNet, plus the same information for all senses in a direct semantic relation. This setup specifically targets situations where such a resource serves as the sense inventory but no large sense-annotated corpus is available for supervised wsd (thus precluding use of the most frequent sense backoff). This is the case for many languages, where wordnets are available but manually tagged corpora are not, and also for domain-specific wsd using the English WordNet. Whereas other approaches in this setting (Ponzetto and Navigli, 2010; Henrich, Hinrichs, and Vodolazova, 2012; and our own methods to be presented in Chapter 4) aim at improving wsd accuracy through the combination of several lexical resources, here we restrict ourselves to WordNet and bridge the lexical gap with non-supervised, data-driven methods.

How one computes the overlap between two strings was left unspecified by Lesk; we therefore adopt the simple approach of removing occurrences of the target word, treating both strings as bags of case-insensitive word tokens, and taking the cardinality of their intersection. We do not preprocess the texts by stemming, lemmatization, or stop word filtering, since the terms in the distributional thesaurus are likewise unprocessed (as in Figure 3), and because preliminary

experiments showed that such preprocessing brought no benefit. We do, however, use POS tagging to discount senses not matching the target's word class and to constrain the entries returned by the DT. We use the sentence containing the target word as the context. The sense with the highest overlap with the context is assigned a probability of 1; when $k \geq 2$ senses are tied for the highest overlap count, these senses are assigned a probability of $1/k$. All other senses are assigned a probability of 0. The probabilities are then used for scoring during evaluation, as in Formula 2.15 on page 28.

3.4.1 Use of distributional information

In this section we describe the creation and the use of our distributional thesaurus.² In the fashion of Lin (1998b), we parsed a ten million-sentence English news corpus from the Leipzig Corpora Collection (Biemann, Heyer, *et al.*, 2007) with the Stanford parser (de Marneffe *et al.*, 2006) and used collapsed dependencies to extract features for words: each dependency triple (w_1, r, w_2) denoting a directed dependency of type r between words w_1 and w_2 results in a feature (r, w_2) characterizing w_1 , and a feature (w_1, r) characterizing w_2 . Words are thereby represented by the concatenation of the surface form and the POS as assigned by the parser. After counting the frequency of each feature for each word, we apply a significance measure (the log-likelihood test (Dunning, 1993)), rank features per word according to their significance, and prune the data, keeping only the 300 most salient features per word. The similarity of two words is given by the number of their common features (which we will shortly illustrate with an example). The pruning operation greatly reduces run time at thesaurus construction, rendering memory reduction techniques like that of Goyal *et al.* (2012) unnecessary. Despite its simplicity and the basic count of feature overlap, we found this setting to be equal to or better than more complex weighting schemes in word similarity evaluations. Across all parts of speech, the DT contains five or more similar terms for a vocabulary of over 150 000 words.

To illustrate the DT, Table 2 shows the top three most similar words to the noun *paper*, together with the features which determine the similarities. Amongst their 300 most salient features as determined by the significance measure, *newspaper* and *paper* share 45, *book* and *paper* share 33, and *article* and *paper* share 28; these numbers constitute the terms' respective similarity scores.

The DT is used to expand the context and the sense definitions in the following way: For each content word (that is, adjectives, nouns, adverbs, and verbs) we retrieve the n most similar terms from the DT and add them to the textual representation. Since our overlap-

² A more detailed treatment of the technique can be found in Biemann and Riedl (2013).

TERM	SCORE	SHARED FEATURES
newspaper/NN	45/300	told/VBD/-dobj column/NN/-prep/in local/JJ/amod editor/NN/-poss edition/NN/-prep/of editor/NN/-prep/of hometown/NN/nn industry/NN/-nn clips/NNS/-nn shredded/JJ/amod pick/VB/-dobj news/NNP/appos daily/JJ/amod writes/VBZ/-nsubj write/VB/-prep/for wrote/VBD/-prep/for wrote/VBD/-prep/in wrapped/VBN/-prep/in reading/VBG/-prep/in reading/VBG/-dobj read/VBD/-prep/in read/VBD/-dobj read/VBP/-prep/in read/VB/-dobj read/VB/-prep/in record/NN/prep/of article/NN/-prep/in reports/VBZ/-nsubj reported/VBD/-nsubj printed/VBN/amod printed/VBD/-nsubj printed/VBN/-prep/in published/VBN/-prep/in published/VBN/partmod published/VBD/-nsubj sunday/NNP/nn section/NN/-prep/of school/NN/nn saw/VBD/-prep/in ad/NN/-prep/in copy/NN/-prep/of page/NN/-prep/of pages/NNS/-prep/of morning/NN/nn story/NN/-prep/in
book/NN	33/300	recent/JJ/amod read/VB/-dobj read/VBD/-dobj reading/VBG/-dobj edition/NN/-prep/of printed/VBN/amod industry/NN/-nn described/VBN/-prep/in writing/VBG/-dobj wrote/VBD/-prep/in wrote/VBD/rcmod write/VB/-dobj written/VBN/rcmod written/VBN/-dobj wrote/VBD/-dobj pick/VB/-dobj photo/NN/nn co-author/NN/-prep/of co-authored/VBN/-dobj section/NN/-prep/of published/VBN/-dobj published/VBN/-nsubjpass published/VBD/-dobj published/VBN/partmod copy/NN/-prep/of buying/VBG/-dobj buy/VB/-dobj author/NN/-prep/of bag/NN/-nn bags/NNS/-nn page/NN/-prep/of pages/NNS/-prep/of titled/VBN/partmod
article/NN	28/300	authors/NNS/-prep/of original/JJ/amod notes/VBZ/-nsubj published/VBN/-dobj published/VBD/-dobj published/VBN/-nsubjpass published/VBN/partmod write/VB/-dobj wrote/VBD/rcmod wrote/VBD/-prep/in written/VBN/rcmod wrote/VBD/-dobj written/VBN/-dobj writing/VBG/-dobj reported/VBD/-nsubj describing/VBG/partmod described/VBN/-prep/in copy/NN/-prep/of said/VBD/-prep/in recent/JJ/amod read/VB/-dobj read/VB/-prep/in read/VBD/-dobj read/VBD/-prep/in reading/VBG/-dobj author/NN/-prep/of titled/VBN/partmod lancet/NNP/nn

Table 2. A DT entry with features, showing terms similar to the noun *paper*

based approaches treat contexts and sense definitions as unordered bags of words, we do not need to take precautions with respect to the positions of words and expansions within the texts. The bags of words are filtered by removing occurrences of the disambiguation target. Then, we count the overlaps as usual between the expanded context and sense definitions. In our experiments we test $n = 10, 20, \dots, 100$.

We had the intuition that the optimal number of expansions may depend on the part of speech of the word to be disambiguated, and perhaps also on the parts of speech of the words being expanded. Therefore, we parameterized our expansion procedure such that the part of speech of the target word determined the number of expansions, and also whether all words were expanded or only those of a certain part of speech.

3.4.2 Data sets

As discussed in §2.4, data sets for WSD can generally be classified as *fine-grained* or *coarse-grained* according to the granularity of the sense inventory used for the annotations. Another common distinction is between the *all-words* task, in which the aim is to provide an annotation for every content word in long running texts, and the *lexical sample* task, where several instances from the same small set of target words are annotated in (usually very short) contexts. We tested our systems on several coarse- and fine-grained data sets, and in both the all-words and lexical sample settings. However, most of our analysis will focus on the coarse-grained all-words scenario, as all-words provides a wider and more natural distribution of target words and senses, and because the fine sense distinctions of WordNet are considered a major obstacle to accurate WSD. Additionally, as we discuss below, the fine-grained data sets available to us have various issues which render them unsuitable for comparisons with the state of the art.

Our coarse-grained data set is from the SemEval-2007 English all-words disambiguation task (Navigli, Litkowski, *et al.*, 2007). It consists of five non-fiction documents from various sources, where each of the 2269 content words (362 adjectives, 1108 nouns, 208 adverbs, and 591 verbs) has been annotated with clusters of senses from WordNet 2.1. For this data set only, we make a slight modification to our algorithm to account for this clustering: instead of choosing the WordNet sense with the highest overlap, we add up the overlap counts of each cluster's constituent senses, and then select the best cluster.

For our fine-grained experiments, we used the all-words and lexical sample tasks from SENSEVAL-2 (Kilgarriff, 2001; Palmer, Fellbaum, *et al.*, 2001) and SENSEVAL-3 (Snyder and Palmer, 2004). With these data sets, however, several factors hinder direct comparison to previously

published results. There are a number of errors in the gold standard annotations, and the methodology of the original task is different from what has subsequently become common. Specifically, not all of the target words have a corresponding entry in the sense inventory, and systems were originally expected to mark these “unassignable” senses as such. In the case of SENSEVAL-2, the gold standard annotations were made using an unpublished (and now lost) version of WordNet. Subsequent researchers have adopted a variety of mutually incompatible methods for dealing with these issues. For our runs, we use Rada Mihalcea’s WordNet 3.0 conversions of the corpora³ and remove from consideration all “unassignable” target word instances. We do not fix the erroneous annotations, which means that even our baselines cannot achieve 100% coverage.

3.4.3 Baselines and measures

We use the evaluation metrics standard in word sense disambiguation research (see §2.4.2). Each disambiguation target receives a *score* equal to the probability the system assigned to the correct sense.⁴ *Coverage* is the proportion of target word instances for which the system attempted a sense assignment, *precision* (*P*) is the sum of scores for the correct sense assignments divided by the number of target word instances for which the system made an attempt, and *recall* (*R*, also known as *accuracy*) is the sum of scores for the correct sense assignments divided by the number of target word instances. The *F₁-measure* is the harmonic mean of precision and recall: $F_1 = 2PR \div (P + R)$. Note that according to these definitions, $P \leq R$, and when coverage is 100%, $P = R = F_1$. In this chapter we express all these measures as a percentage (*i. e.*, in the range [0, 100]).

Our systems were compared against the computed random baseline described in §2.4.3. We also report accuracy of the most frequent sense (MFS) baseline, which always chooses the sense which occurs most frequently in SemCor (Miller, Leacock, *et al.*, 1993), a very large manually annotated corpus. Note that unlike our knowledge-based systems, MFS is a supervised baseline, and cannot actually be applied to the use cases for which our non-supervised systems are intended. Nonetheless, it is included here as it gives some idea of what accuracy could be achieved, at minimum, were one to go to the considerable expense of creating a manually tagged training corpus. Note that MFS is a notoriously difficult baseline to beat even for supervised systems.

³ <https://web.eecs.umich.edu/~mihalcea/downloads.html#sensevalsemcor>

⁴ Where the probability is less than 1, this is mathematically equivalent to the average score which would have been obtained, over repeated runs, of choosing a sense at random to break any ties. It is effectively a backoff to a random sense baseline, ensuring 100% coverage even when there is no overlap.

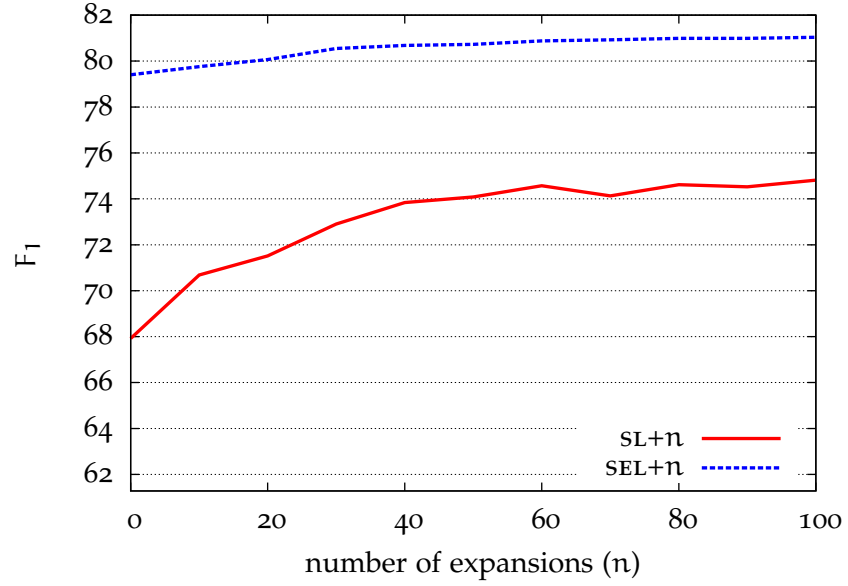


Figure 4. Results (F_1) on the SemEval-2007 corpus for simplified and simplified extended Lesk, by number of lexical expansions

3.4.4 Results

On the SemEval-2007 data set, the basic configuration of simplified Lesk (SL+0)—*i. e.*, without any lexical expansions—achieves an overall F_1 of 67.92, which is already much better than the random baseline ($F_1 = 61.28$). When we tried adding a fixed number of lexical expansions to all content words, we observed that accuracy generally increased sublinearly with the number of expansions. The highest accuracy was obtained by using 100 expansions (the maximum number we tried); we denote this configuration SL+100. SL+100’s F_1 -measure of 74.81 represents a relative increase of more than 10% over SL+0. The simplified extended Lesk configuration also benefitted from lexical expansions, though the effect was less pronounced: the basic version without expansions (SEL+0) achieves $F_1 = 79.40$, and adding 100 lexical expansions (SEL+100) yields a relative performance increase of just over 2%, to $F_1 = 81.03$. As with simplified Lesk, accuracy increased sublinearly with the number of expansions. This effect is visualized in Figure 4, which plots the F_1 -measure for the two algorithms according to the number of lexical expansions used. (The baseline in this plot is the accuracy of the random baseline.)

Table 3 shows the F_1 -measure of our two baselines (top), our algorithms (middle), and some state-of-the-art knowledge-based systems (bottom), broken down by the target words’ parts of speech. In each column the best result, excluding the supervised MFS baseline, is shown in boldface. The third-party systems are as follows:

SYSTEM	PART OF SPEECH				
	ADJ.	NOUN	ADV.	VERB	ALL
MFS baseline	84.25	77.44	87.50	75.30	78.89
random baseline	68.54	61.96	69.15	52.81	61.28
SL+O	75.32	69.71	69.75	59.46	67.92
SL+100	82.18	76.31	78.85	66.07	74.81
SEL+O	87.19	81.52	74.87	72.26	79.40
SEL+100	88.40	83.45	80.29	72.25	81.03
TKB-UO	78.73	70.76	74.04	62.61	70.21
MII+ref	82.04	80.05	82.21	70.73	78.14
Chen _{s2c}	—	81.6	—	—	75.8
WN++-DC	—	79.4	—	—	—

Table 3. Results (F_1) on the SemEval-2007 corpus by part of speech

TKB-UO (Anaya-Sánchez *et al.*, 2007) was the top knowledge-based system at the SemEval-2007 competition. It is a clustering-based system which uses WordNet 2.1 (but not its sense frequency information) as its sole source of knowledge.

MII+REF (Li *et al.*, 2010), a topic model approach. Among third-party systems, it achieves the highest overall result we are aware of. The system maps sense descriptions and target word contexts to a topic distribution vector as sampled by LDA (Blei *et al.*, 2003).

CHEN_{s2c} (Chen, Liu, *et al.*, 2014) uses neural networks to learn word and sense embeddings using only the glosses from WordNet and a large, unannotated background corpus. Words are then disambiguated by comparing their associated sense vectors to a word vector constructed from the context. The results reported here are for the “unsupervised” variant of their algorithm that doesn’t use the MFS backoff. Among third-party systems, it has the highest accuracy on nouns.

WN++-DC (Ponzetto and Navigli, 2010) disambiguates nouns in a sentence by building a graph of candidate senses linked by semantic relations, and then for each target word selecting the sense with the highest vertex degree. When using semantic relations from WordNet alone the method achieves $F_1 = 74.5$, but when WordNet is enriched with additional semantic relations from Wikipedia, an online encyclopedia, performance increases to $F_1 = 79.4$. Note that, uniquely among the results in the table, WN++-DC does not achieve full coverage ($P = 87.3$, $R = 72.7$).

POS-OPTIMIZED RESULTS. We also tried using different expansion strategies for target words of different parts of speech: for each target word *pos*, we tried expanding only adjectives, only nouns, *etc.*, and tried each of these scenarios for the same eleven values of *n* as previously. Because this procedure involved tuning on the test data, we do not include the results for comparison in Table 3. However, they are interesting as they give an upper bound on performance for the case where the expansions-per-pos parameters are optimized on a set of manually annotated training examples—that is, a mildly supervised variant of our otherwise knowledge-based algorithms.

For simplified Lesk, we found that accuracy for nouns, adverbs, and verbs remained highest when all content words were given 100 expansions, but adjectives fared better when all content words were given only 60 expansions. With this configuration we achieve an overall F_1 of 74.94. The best-performing simplified extended Lesk configuration achieves $F_1 = 81.27$ when for adjectives we apply 20 expansions to all content words; for nouns, 60 expansions to all content words; for adverbs, 80 expansions to all content words; and for verbs, 30 expansions to adverbs only. That verbs benefit from adverb expansions is not surprising, given that the latter often serve to modify the former. Why the optimal *number* of expansions should vary with the target word part of speech is not as clear. In any case, the extra performance gains from pos expansion optimization were quite small, not exceeding a quarter of a percentage point over the non-optimized versions.

FINE-GRAINED RESULTS. As with the coarse-grained task, we discovered that using lexical expansions resulted in an improvement in accuracy in the fine-grained tasks. However, in this setting we did not observe the same continuously improving accuracy from using more and more expansions; in all but one case, adding expansions helped to a point, after which accuracy started to decrease. This effect was particularly noticeable with simplified extended Lesk, where peak accuracy was achieved with around 30 expansions. For simplified Lesk, the optimum was less stable across the corpora, ranging from 60 to 100 expansions. We believe that this is because the expanded terms provided by the *DT* reflect broad conceptual relations which, taken in aggregate, do not precisely map to the narrow sense distinctions of the sense inventory. This is not a problem when we stick to the first highly salient expansions provided by the *DT*, but beyond this the conceptual relations become too tenuous and fuzzy to facilitate disambiguation.

Table 4 shows the results of our systems and baselines on the *SENSEVAL-2* lexical sample and all-words tasks and the *SENSEVAL-3* all-words task. For simplified extended Lesk we show the results of using 30 expansions (*SEL+30*); as simplified Lesk had no consistent peak ac-

SYSTEM	SENSEVAL-2	SENSEVAL-2	SENSEVAL-3
	LEXICAL SAMPLE	ALL-WORDS	ALL-WORDS
MFS baseline	41.56	65.36	65.63
random baseline	15.46	39.54	32.89
SL+0	17.10	39.02	35.41
SL+100	20.92	45.69	37.17
SEL+0	28.60	54.22	48.76
SEL+30	32.72	57.77	53.09

Table 4. Results (F_1) on the SENSEVAL-2 and -3 corpora

curacy we stick with 100 expansions (SL+100). The results, while quite expectedly lower than the coarse-grained scores in absolute terms, nonetheless validate the utility of our approach in fine-grained tasks. Not only does the use of expansions significantly increase the accuracy, but in the case of the SENSEVAL-2 corpora, the relative increase is much higher than that of the coarse-grained tasks. For SL+100, the relative improvements over the unexpanded algorithms for the lexical sample and all-words data sets are 22.3% and 17.1%, respectively, and for SEL+100 they are 14.4% and 6.5%, respectively.

3.5 ANALYSIS

In this section, we discuss our results and put them in the perspective of applicability of WSD systems. Our lexical expansion mechanism leads to a relative improvement of up to 22% in the fine-grained evaluation and 10% in the coarse-grained evaluation for the “simple” setup. This is achieved by merely adding lexical items to the representation of the sense description and context, and without changing the algorithm. Especially in situations where there exists a reasonably coarse-grained MRD for the language or domain, this is a major improvement over previous approaches on applications where one is not in the comfortable situation of having sense frequency information. In our opinion, this scenario has been neglected in the past, despite occurring in practice much more often than the case where one has access to a rich LSR, let alone sufficient training data for supervised disambiguation.

The expansions from distributional similarity are complementary to those coming from richer knowledge resources, as our results for fitting simplified extended Lesk with DT expansions show: even in the situation where a richer lexical resource allows for bridging the lexical gap via descriptions from related senses, we still see an additional relative improvement of 2% to 14% when comparing the F_1 -measure

		SL+100			
		UNASSIGNED	INCORRECT	CORRECT	TOTAL
SL+0 {	UNASSIGNED	0.2	8.1	9.7	18.0
	INCORRECT	0.1	14.1	6.2	20.4
	CORRECT	0.0	2.8	58.8	61.6
	TOTAL	0.4	25.0	74.7	100.0

Table 5. Confusion matrix for SL+0 and SL+100

of the SEL+n system against the SEL+0 baseline. Not only does this system outperform all previous approaches to coarse-grained WSD without MFS backoff, it is also able to outperform the MFS baseline itself, both generally and for certain parts of speech.

We emphasize that while the DT uses additional text data for computing the similarity scores used in the lexical expansion step, the overall system is purely knowledge-based because it is not trained on sense-labelled examples; the DT similarities are computed on the basis of an automatically parsed but otherwise unannotated corpus. This marks an important difference from the system described in Navigli and Velardi (2005) which, although it also uses collocations extracted from large corpora, avails itself of manual sense annotations wherever possible.

While the comparison of results to other methods on the same coarse-grained data sets suggests that lexical expansion using a distributional thesaurus leads to more precise disambiguation systems than word or topic vectors, our point is somewhat different: Realizing lexical expansions and thus explicitly generating associated terms to a textual representation opens up a new way of thinking about bridging lexical gaps and semantic matching of similar meaning. In light of the fact that distributional similarity and overlap-based approaches to WSD have existed for a long time now, it is somewhat surprising that this avenue had not been explored earlier.

3.5.1 Error analysis

In order to better understand where and how our system is succeeding and failing, we now present an error analysis of the results, both in aggregate and for some individual cases. To begin, we computed a confusion matrix showing the percentage of the 2269 SemEval-2007 target word instances for which the SL+0 and SL+100 algorithms made a correct disambiguation, made an incorrect disambiguation, or failed to make an assignment at all without resorting to the random choice backoff (see Table 5). Table 6 shows the same confusion matrix for the

		SEL+100			
		UNASSIGNED	INCORRECT	CORRECT	TOTAL
SEL+0 {	UNASSIGNED	0.1	3.5	4.0	7.6
	INCORRECT	0.0	14.9	2.8	17.7
	CORRECT	0.0	1.9	72.9	74.7
	TOTAL	0.1	20.3	79.6	100.0

Table 6. Confusion matrix for SEL+0 and SEL+100

SEL+0 and SEL+100 algorithms.⁵ As can be seen, the pattern of contingencies is similar. Because of the sheer size of the expanded sense descriptions and contexts with this task, however, in the following analysis we stick to the simplified Lesk scenario.

As we hypothesized, using lexical expansions successfully bridges the lexical gap: whereas the basic simplified Lesk was able to make a sense assignment (be it correct or incorrect) in only 82.0% of cases, SL+100 could do so 99.6% of the time. SL+100 was able to correctly disambiguate over half of all the target words for which SL+0 failed to make any sense assignment. This contingency—some 9.7% of all instances—accounts for the majority of SL+100’s improvement over SL+0. However, in 6.2% of cases SL+100’s improvement resulted from successfully revising an incorrect answer of SL+0. We randomly selected ten of these cases and found that in all of them, all the overlaps for SL+0 were from a small number of non-content words (*the, of, in, etc.*), with the chosen sense achieving only one or two more overlaps than the runners-up; thus, the lexical gap is still at fault here. By contrast, the expanded sense definitions and contexts used by SL+100 for these cases always contained dozens of overlapping content words, and the overlap count for the chosen sense was markedly higher than for the runners-up.

What is also interesting to consider is the 0.2% of cases where both algorithms neglected to make a sense assignment, apparently signifying SL+100’s failure to bridge the lexical gap. We manually examined all of these instances and found that for all but one, the systems failed to disambiguate the target words because the sentences containing them were extremely short, usually with no other content words apart from the target word. It is unlikely that any knowledge-based algorithm restricting itself to sentential context could succeed in such cases, and no reasonable number of lexical expansions is likely to help. Our choice to use sentential context was motivated by simplicity and expediency; a more refined WSD algorithm could, of course, use a sliding or dynamically sized context window and thereby avoid this problem. The remaining case was a sentence of normal length where

⁵ Totals in both tables may not add up exactly due to rounding.

SL+0 found no overlapping content words between the definition and the context, but SL+100 produced a two-way tie between two of the clusters, one of which was the correct one.

It is also of interest to know why SL+0 was able to correctly disambiguate some words which SL+100 could not; these represent 2.8% of the instances. Again, we drew a random sample of these instances, and observed that in all of them, the only overlaps found by SL+0 were for non-content words; the fact that it happened to choose the correct sense cluster can therefore be chalked up to chance.

Though it has been relatively easy to identify the reasons behind SL+100's correct assignments, and behind its failures to make any assignment at all, it is not so easy to deduce the causes of its incorrect assignments. We observe that the system had disproportionate difficulties with verbs, which constitute 35% of the incorrect disambiguations but only 26% of all target words in the corpus. Particularly troublesome were verbs such as *be*, *go*, *have*, and *do*, which are often used as auxiliaries. On their own they contribute little or no semantic information to the sentence, and their dictionary definitions tend to explain their grammatical function, so there is little opportunity for meaningful lexical or conceptual overlaps. A related problem was observed for adverbs and adjectives: the problematic cases here were often generic terms of restriction, intensification, or contrast (*e.g.*, *different*, *just*, *only*, *so*) which are used in a wide variety of semantic contexts and whose dictionary definitions focus on usage, or else constitute concise rephrasings using equally generic terms. Purely definition-based disambiguation approaches are unlikely to help in any of these cases; an accurate knowledge-based approach would probably need to be aware and make use of information beyond the lexical-semantic level, such as verb frames and semantic roles, or incorporate the grammatical structure around the target word for matching.

3.6 CONCLUSION

We have proposed a novel method for word sense disambiguation based on word overlap between sense descriptions and the target word context. Our method uses lexical expansions from a distributional thesaurus computed over dependency-context similarities over a large background corpus. We found that applying our conceptually simple extension to two traditional knowledge-based methods successfully bridged the lexical gap, resulting in performance gains exceeding that of state-of-the-art knowledge-based systems, and approaching or even exceeding the MFS baseline. The concept of lexical expansion is a promising avenue to enrich classic, word-based NLP algorithms with additional lexical material. The intuitions of overlap-based approaches are thereby complemented by a method that makes

associations explicit and bridges the lexical gaps for semantically similar contexts that are expressed in a different wording.

In the time since our method was originally published (Miller, Bie-mann, *et al.*, 2012) it has become a benchmark, model, or component for other knowledge-based ambiguity resolution systems. Schwab *et al.* (2013), for example, present a study of nineteen state-of-the-art wsd algorithms, including seven of their own knowledge-based systems based on ant colony optimization, simulated annealing, and genetic algorithms. They find that lexically expanded Lesk remains the top-performing knowledge-based system. Basile *et al.* (2014) describe an overlap-based disambiguator which strongly resembles ours. The biggest difference is that, rather than performing purely symbolic matching of the expanded texts, they represent the gloss and context as vectors in a geometric space generated by a distributional semantic model. Their method was found to outperform all systems from the SemEval-2013 wsd task (Navigli, Jurgens, *et al.*, 2013). Finally, our method has shown promise for use with non-traditional sense inventories. At the SemEval-2013 task for end-to-end evaluation of wsi and wsd (Navigli and Vannella, 2013), Zorn and Gurevych (2013) use our disambiguation method with an automatically induced sense inventory. The system was then applied in a search result clustering task, where it was the best performing system according to one of the evaluation metrics.

In recent years, there has been a great resurgence of interest in the use of neural networks, and particularly deep learning, for natural language processing. In supervised wsd, these techniques have led to state-of-the-art performance (Rothe and Schütze, 2015), though they suffer from the same knowledge acquisition bottleneck that plagues all supervised systems. In this chapter we compared the results of our system against those of Chen, Liu, *et al.* (2014), a notable attempt at leveraging the benefits of both neural networks and distributional information in a purely knowledge-based setting. The latter system proves to be considerably less accurate than ours when disambiguating nouns, and across all parts of speech does not even approach the accuracy of an ordinary and well-known Lesk variant. This suggesting that neural network-based techniques are not (yet) serious contenders for knowledge-based wsd.

Despite our own method's strengths, we can point to a number of ways in which it could be improved. First of all, since a DT is static and thus not dependent on the context, it generates spurious expansions, such as the similar terms for *charge* in Figure 3, which is obviously dominantly used in its "criminal indictment" sense in the background corpus. At best, these expansions, which implicitly capture the sense distribution in the background corpus, result in less overlap with the correct sense description—but they might well result in assigning incorrect senses. A straightforward improvement

would be to alter the lexical expansion mechanism so as to be sensitive to the context—something that is captured, for example, by LDA sampling (Blei *et al.*, 2003). A further extension would be to have the number of lexical expansions depend on the DT similarity score (be it static or contextualized) instead of the fixed number we used here.

In the future, we would like to examine the interplay of lexical expansion methods in disambiguation systems with richer knowledge resources (*e.g.*, Navigli and Ponzetto, 2010; Gurevych, Eckle-Kohler, *et al.*, 2012) and apply our approach to other languages with fewer lexical resources. We would also like to more directly establish the language independence of our approach by running some further experiments on non-English data sets. (At the time of writing, experiments on our own German-language GLASS data set and some of the other sense-annotated corpora we cover in Chapter 7 are already underway.) Also, it seems promising to apply lexical expansion techniques to text similarity, text segmentation, machine translation, and semantic indexing.

4.1 MOTIVATION

Lexical-semantic resources (LSRs) are used in a wide variety of natural language processing tasks, including machine translation, question answering, automatic summarization, and word sense disambiguation. Their coverage of concepts and lexical items, and the quality of the information they provide, are crucial for the success of these tasks, which has motivated the manual construction of full-fledged electronic LSRs. However, the effort required to produce and maintain such expert-built resources is phenomenal (Briscoe, 1991). Early attempts at resolving this knowledge acquisition bottleneck focused on methods for automatically acquiring structured knowledge from unstructured knowledge sources (Hearst, 1998). More recent contributions treat the question of automatically connecting or merging existing LSRs which encode heterogeneous information for the same lexical and semantic entities, or which encode the same sort of information but for different sets of lexical and semantic entities. These approaches have until now focused on pairwise linking of resources, and in most cases are applicable only to the particular resources they align.

In this chapter we address the novel task of automatically aligning arbitrary numbers and types of LSRs through the combination of existing pairwise alignments, which in theory reduces the number of specialized algorithms required to find concept pairs in any n resources from as many as $\binom{n}{2} = n! \div 2(n-2)!$ to as little as $n-1$. In particular, we produce an aligned LSR using existing pairwise alignments of WordNet, Wikipedia, and Wiktionary, three LSRs with very different sizes, scopes, and methods of construction.

There are a number of potential benefits and applications of such an aligned resource:

INCREASED COVERAGE. Coverage is crucial for almost every natural language processing task. However, as individual LSRs vary with respect to the senses and lexical items they include, it is possible that no one of them may cover all the senses and lexical items in a given data set. This problem can be mitigated by unifying LSRs with complementary coverage into a single aligned resource.

ENRICHED SENSE REPRESENTATION. Even where a sense or lexical item exists in a given LSR, the quality and quantity of the information it records may not be sufficient for a given NLP appli-

cation. An aligned resource allows applications to avail themselves of the sum of the aligned resources' data for each sense or lexical entry. For a given sense, for example, an application may have access to its hypernyms and hyponyms from one resource, its etymology and pronunciation from another, and its subcategorization frames from yet another.

DECREASED SENSE GRANULARITY. In some cases, an LSR may provide sufficient sense coverage, but the distinctions it makes between senses may be too fine-grained for a given application. For example, poor performance of word sense disambiguation systems has often been blamed on the fact that the distinctions among senses of WordNet, the standard sense inventory used in the research and evaluation competitions, are too subtle to permit human annotators, let alone automated systems, to reliably distinguish them (Navigli, 2006). However, many annotated corpora, along with most wsd systems, rely on WordNet as a sense inventory; choosing a different sense inventory would make it hard to retrain supervised systems and impossible to compare results against the existing body of research.

When two resources of differing granularities are aligned, multiple senses from the finer-grained resource are often mapped to the same sense from the coarse-grained one. This many-to-one mapping can be used to induce a sense clustering on the fine-grained inventory. Treating these clusters as coarse-grained senses allows the fine-grained inventory to continue to be used in scenarios where broader sense distinctions are more useful.

4.2 BACKGROUND AND RELATED WORK

4.2.1 Lexical-semantic resources

The oldest type of lexical-semantic resource is the *dictionary*. In its simplest form, a dictionary is a collection of *lexical items* (words, multiword expressions, *etc.*) for which the various *senses* (or *concepts*) are enumerated and explained through brief prose *definitions*. Many dictionaries provide additional information at the lexical or sense level, such as etymologies, pronunciations, example sentences, and usage notes. A *wordnet*, like a dictionary, enumerates the senses of its lexical items, and may even provide some of the same sense-related information, such as definitions and example sentences. What distinguishes a wordnet, however, is that the senses and lexical items are organized into a network by means of conceptual-semantic and lexical relations. *Encyclopedias* are similar to dictionaries, except that their concept descriptions are much longer and more detailed.

WORDNET. *WordNet* (Fellbaum, 1998) is an expert-built semantic network for English which has seen myriad applications. For each sense (in WordNet parlance, a *synset*) WordNet provides a list of synonymous lexical items, a definition, and zero or more example sentences showing use of the lexical items.¹ Within each version of WordNet, synsets can be uniquely identified with a label, the *synset offset*, which encodes the synset's part of speech and its position within an index file. Synsets and lexical items are connected to each other by various semantic and lexical relations, respectively, in a clear-cut subsumption hierarchy. The version of WordNet we use here, 3.0, contains 117 659 synsets and 206 941 lexical items.

WIKTIONARY. *Wiktionary*² is an online, free content dictionary collaboratively written and edited by volunteers. It includes a wide variety of lexical and semantic information such as definitions, pronunciations, translations, inflected forms, pictures, example sentences, and etymologies, though not all lexical items and senses have all of this information. The online edition does not provide a convenient and consistent means of directly addressing individual lexical items or their associated senses; however, the third-party API JWKTl (Zesch, Müller, *et al.*, 2008) can assign unique identifiers for these in snapshot editions downloaded for offline use. A snapshot of the English edition from 3 April 2010 contains 421 847 senses for 335 748 English lexical items.

WIKIPEDIA. *Wikipedia*³ is an online free content encyclopedia; like Wiktionary, it is produced by a community of volunteers. Wikipedia is organized into millions of uniquely named *articles*, each of which presents detailed, semi-structured knowledge about a specific concept. Among LSRs, encyclopedias do not have the same established history of use in NLP as dictionaries and wordnets, but Wikipedia has a number of features—particularly its network of internal hyperlinks and its comprehensive article categorization scheme—which make it a particularly attractive source of knowledge for NLP tasks (Zesch, Gurevych, *et al.*, 2007; Gurevych and Wolf, 2010).

4.2.2 Pairwise alignments

Each of the aforementioned resources has different coverage (primarily in terms of domain, part of speech, and sense granularity) and encodes different types of lexical and semantic information. There is a considerable body of prior work on connecting or combining

¹ In this chapter we use the term *sense* in a general way to refer to the concepts or meanings described by an LSR. This is in contrast to the WordNet documentation, where it refers to the pairing of a lexical item with a synset.

² <https://www.wiktionary.org/>

³ <https://www.wikipedia.org/>

them at the concept level in order to maximize the coverage and quality of the data; this has ranged from largely manual alignments of selected senses (Meyer and Gurevych, 2010; Dandala *et al.*, 2013) to minimally supervised or even fully automatic alignment of entire resource pairs (Ruiz-Casado *et al.*, 2005; de Melo and Weikum, 2009; Meyer and Gurevych, 2011; Niemann and Gurevych, 2011; Hartmann and Gurevych, 2013; Matuschek and Gurevych, 2013; Navigli and Ponzetto, 2013).⁴ In our work, we use the alignments from Meyer and Gurevych (2011) and Matuschek and Gurevych (2013), which were among the few that were publically available in a transparent, documented format at the time of our study. We briefly describe them below.

THE WORDNET–WIKTIONARY ALIGNMENT. A method for automatically aligning English Wiktionary senses with WordNet 3.0 synsets is described by Meyer and Gurevych (2011). They use the same basic alignment method described above: for a given WordNet synset, the set of target candidates is initially set to all Wiktionary senses with headwords corresponding to the any of the synset’s lexical items. Each pairing of the WordNet synset and the candidate is then scored with two textual similarity measures, the WordNet sense being represented by its definition and lexical items and those of its hypernym, and the Wiktionary sense by its word, definition, usage examples, and synonyms. If the scores exceed certain thresholds, the pair is considered an alignment. The method is thus not restricted to producing 1:1 alignments; any given identifier may be mapped to zero, one, or multiple targets.

The 3 April 2010 snapshot of Wiktionary used by the authors contains 421 847 senses for 335 748 English lexical items. Their published alignment file consists of 56 952 aligned pairs, but as the same Wiktionary sense is sometimes paired with multiple WordNet synsets, the set of aligned pairs can be reduced mathematically (see §4.3) to 50 518 $n:1$ sense mappings, where n ranges from 1 to 7. As with the WordNet–Wikipedia alignment, the vast majority of the mappings (89%) are 1:1; the full distribution of mapping sizes is shown in the second row of Table 7. The authors found that 60 707 (51.60%) of the WordNet synsets and 371 329 (88.02%) of the Wiktionary senses could not be aligned.

THE WORDNET–WIKIPEDIA ALIGNMENT. The Dijkstra-wsa algorithm, a state-of-the-art graph-based approach to sense alignment, is described by Matuschek and Gurevych (2013). The algorithm begins by representing two resources to be aligned as disconnected components

⁴ A different approach with some of the same benefits is to provide a unified interface for accessing multiple LSRs in the same application, as in DKPro LSR (Garoufi *et al.*, 2008) and UBY (Gurevych, Eckle-Kohler, *et al.*, 2012).

	$A_c(\{WN, WP\})$	$A_c(\{WN, WT\})$	$A_c(\{WN, WP, WT\})$
2	23 737	45 111	0
3	4 801	4 601	9 987
4	1 355	656	2 431
5	492	99	1 666
6	234	35	654
7	112	12	441
8	54	4	209
9	28	0	164
≥ 10	44	0	401
total	30 857	50 518	15 953

Table 7. Distribution of synonym sets by cardinality in the two- and three-way conjoint alignments

of a graph, where every sense is represented by a vertex, and the connecting edges represent semantic relations, hyperlinks, or any other relatedness measure provided by the corresponding resource. Next, edges are added to represent “trivial” sense alignments—an alignment is considered trivial when two senses have the same attached lexeme, and the lexeme is monosemous in each resource. Then, for each yet-unaligned sense in one of the components, ‘s (1959) algorithm is run to find the closest sense in the other component with a matching lemma and part of speech. If such a sense exists and its distance is below a certain threshold, an aligning edge between the two senses is added. At the end of this process, a more traditional gloss similarity-based approach is used to attempt to align any remaining unaligned senses.

Matuschek and Gurevych (2013) used Dijkstra-wsa to align WordNet 3.0 with a snapshot of the English edition of Wikipedia containing 3 348 245 articles, resulting in 42 314 aligned pairs. Here, too, the set of aligned pairs can be mathematically reduced to 30 857 $n:1$ mappings, where $1 \leq n \leq 20$. The alignment achieved $F_1 = 0.67$ on the aforementioned well-balanced reference dataset.

4.2.3 Word sense clustering

Clustering fine-grained sense distinctions into coarser units has been a perennial topic in wsd. Past approaches have included using text- and metadata-based heuristics (definition text, domain tags, *etc.*) to derive similarity scores for sense pairs in machine-readable dictionaries (Dolan, 1994; Chen and Chang, 1998), exploiting semantic hierarchies to group senses by proximity or ancestry (Peters *et al.*, 1998; Buitelaar, 2000; Mihalcea and Moldovan, 2001; Tomuro, 2001; Ide,

2006), grouping senses which lexicalize identically when manually translated (Resnik and Yarowsky, 1999), using distributional similarity of senses (Agirre and Lopez de Lacalle, 2003; McCarthy, 2006), exploiting disagreements between human annotators of sense-tagged data (Chklovski and Mihalcea, 2003), heuristically mapping senses to learned semantic classes (Kohomban and Lee, 2005), and deep analysis of syntactic patterns and predicate–argument structures (Palmer, Babko-Malaya, *et al.*, 2004; Palmer, Dang, *et al.*, 2007).

Comparison of these approaches is hampered by the fact that evaluations often are not provided in the papers, are applicable only for the particular LSR used in the experiment, do not provide a random baseline for reference, and/or provide only intrinsic measures such as “reduction in average polysemy” which do not directly speak to the clusterings’ correctness or utility for a particular task. Though many of the above authors cite improved WSD as a motivation for the work, most of them do not actually investigate how their clusterings impact state-of-the-art disambiguation systems. The only exception is Palmer, Dang, *et al.* (2007), who compare results of a state-of-the-art WSD system, as well as human interannotator agreement, on both fine-grained and clustered senses. To ensure that the measured improvement was not due solely to the reduced number of sense choices for each word, they also evaluate a random clustering of the same granularity.

Apart from the above-noted approaches, there has also been interest recently in techniques which reduce WordNet’s sense granularity by aligning it to another, more coarse-grained resource at the level of word senses. Navigli (2006) induces a sense mapping between WordNet and the *Oxford Dictionary of English* (Soanes and Stevenson, 2003) on the basis of lexical overlaps and semantic relationships between pairs of sense glosses. WordNet senses which align to the same *Oxford* sense are clustered together. The evaluation is similar to that later used by Palmer, Dang, *et al.* (2007), except that rather than actually running a WSD algorithm, Navigli expediently takes the raw results of a SENSEVAL WSD competition (Snyder and Palmer, 2004) and does a coarse-grained rescoring of them. The improvement in accuracy is reported relative to that of a random clustering, though unlike in Palmer, Dang, *et al.* (2007) there is no indication that the granularity of the random clusters was controlled. It is therefore hard to say whether the clustering really had any benefit.

Snow *et al.* (2007) and Bhagwani *et al.* (2013) extend Navigli’s approach by training machine learning classifiers to decide whether two senses should be merged. They make use of a variety of features derived from WordNet as well as external sources, such as the aforementioned *Oxford*–WordNet mapping. They also improve upon Navigli’s evaluation technique in two important ways: first, they ensure their baseline random clustering has the same granularity as their in-

duced clustering, and second, the random clustering performance is computed precisely rather than estimated stochastically. While their methods result in an improvement over their baseline, they do require a fair amount of annotated training data, and their features are largely tailored towards WordNet-specific information types. This makes the methods' transferability to resources lacking this information rather difficult.

4.3 APPROACH

Since synonymy is reflexive, symmetric, and transitive (Edmundson, 1967), we can define an equivalence relation \sim on a set of arbitrary sense identifiers $S = \{s_1, s_2, \dots\}$ such that $s_i \sim s_j$ if s_i and s_j are synonyms (*i.e.*, if the senses they refer to are equivalent in meaning). The *synonym set* of an identifier $s \in S$, denoted $[s]_S$, is the equivalence class of s under \sim :

$$[s]_S = \{\sigma \in S \mid \sigma \sim s\}. \quad (4.1)$$

The set of all such equivalence classes is the quotient set of S by \sim :

$$S / \sim = \{[s]_S \mid s \in S\}. \quad (4.2)$$

For any pair of disjoint sets U and V such that $S = U \cup V$ and there exist some $u \in U$ and some $v \in V$ for which $u \sim v$, we say that u and v are an *aligned pair* and that

$$A_f(\{U, V\}) = S / \sim \quad (4.3)$$

is a *full alignment* of the *sources* $\{U, V\}$. More generally, for any set of disjoint sets $W = \{W_1, W_2, \dots\}$ where $S = \bigcup W$ and there exist distinct $W_i, W_j \in W : \exists u \in W_i, v \in W_j : u \sim v$, we say that

$$A_f(W) = S / \sim \quad (4.4)$$

is a full alignment of W .

Full alignments may include synonym sets which do not contain at least one identifier from each of their sources. The *conjoint alignment* which excludes these synonym sets is defined as

$$A_c(W) = \{[s]_S \mid s \in S, \forall W_i \in W : \exists u \in W_i \cap [s]_S\}. \quad (4.5)$$

The cardinality of a full or conjoint alignment is a count of its synonym sets. The number of individual identifiers referenced in an alignment $A(W)$ can also be computed:

$$|A(W)| = \left| \bigcup A(W) \right|. \quad (4.6)$$

If $|A(W)| = |S|$ then $A(W)$ must be a full alignment.

Given a set of identifiers and a set of aligned pairs, we construct a disconnected graph where each aligned pair is represented as an edge connecting two vertices. Finding all the synonym sets is then equivalent to computing the connected components in the graph. For this we use the algorithm described by Hopcroft and Tarjan (1973), which requires time and space proportional to the greater of the number of identifiers or the number of aligned pairs. It works by performing a depth-first search from an arbitrary starting point, marking all the vertices it finds as part of the first connected component. The process is then repeated, starting from an unmarked vertex, to find subsequent connected components, until there are no further unmarked vertices. The algorithm is illustrated as a flowchart in Figure 5.

As discussed in §4.2.1, there exists a unique identifier for each sense or concept described by our three resources: for WordNet, it is the synset offset; for Wikipedia, it is the article title; and for Wiktionary, it is the jwktl ID. Let WT , WN , and WP be disjoint sets of unique sense identifiers from Wiktionary, WordNet, and Wikipedia, respectively; the combined set of all their identifiers is $S = WT \cup WN \cup WP$. The Dijkstra-wsa data corresponds to a set of ordered pairs $(n, p) \in WN \times WP$ where $n \sim p$. This data was sufficient for us to employ the connected component algorithm to compute $A_c(\{WN, WP\})$, the conjoint alignment between WordNet and Wikipedia. We reconstructed the full alignment, $A_f(\{WN, WP\})$, by adding the unaligned identifiers from the original Wikipedia and WordNet databases. Similarly, the Meyer and Gurevych (2011) data contains a set of pairs $(n, k) \in WN \times WKT$ such that $n \sim k$, but it also contains a list of unaligned singletons from both WN and WT . We therefore directly computed both $A_f(\{WN, WT\})$ and $A_c(\{WN, WT\})$ using the connected component algorithm.

4.4 RESULTS

The conjoint three-way alignment of WordNet, Wiktionary, and Wikipedia is a set of 15 953 synonym sets relating 63 771 distinct sense identifiers (27 324 WordNet synsets, 19 916 Wiktionary senses, and 16 531 Wikipedia articles). Of the synonym sets, 9987 (63%) contain exactly one identifier from each source; Table 7 gives further details on synonym set sizes. Since our WordNet–Wikipedia alignment is for nouns only, the synonym sets in the conjoint three-way alignment consist entirely of nouns. The full three-way alignment groups all 3 887 751 identifiers from the original sources into 3 789 065 synonym sets: 69 259 of these are described by adjectives, 3 613 514 by nouns, 12 415 by adverbs, 76 992 by verbs, and 16 885 by other parts of speech.

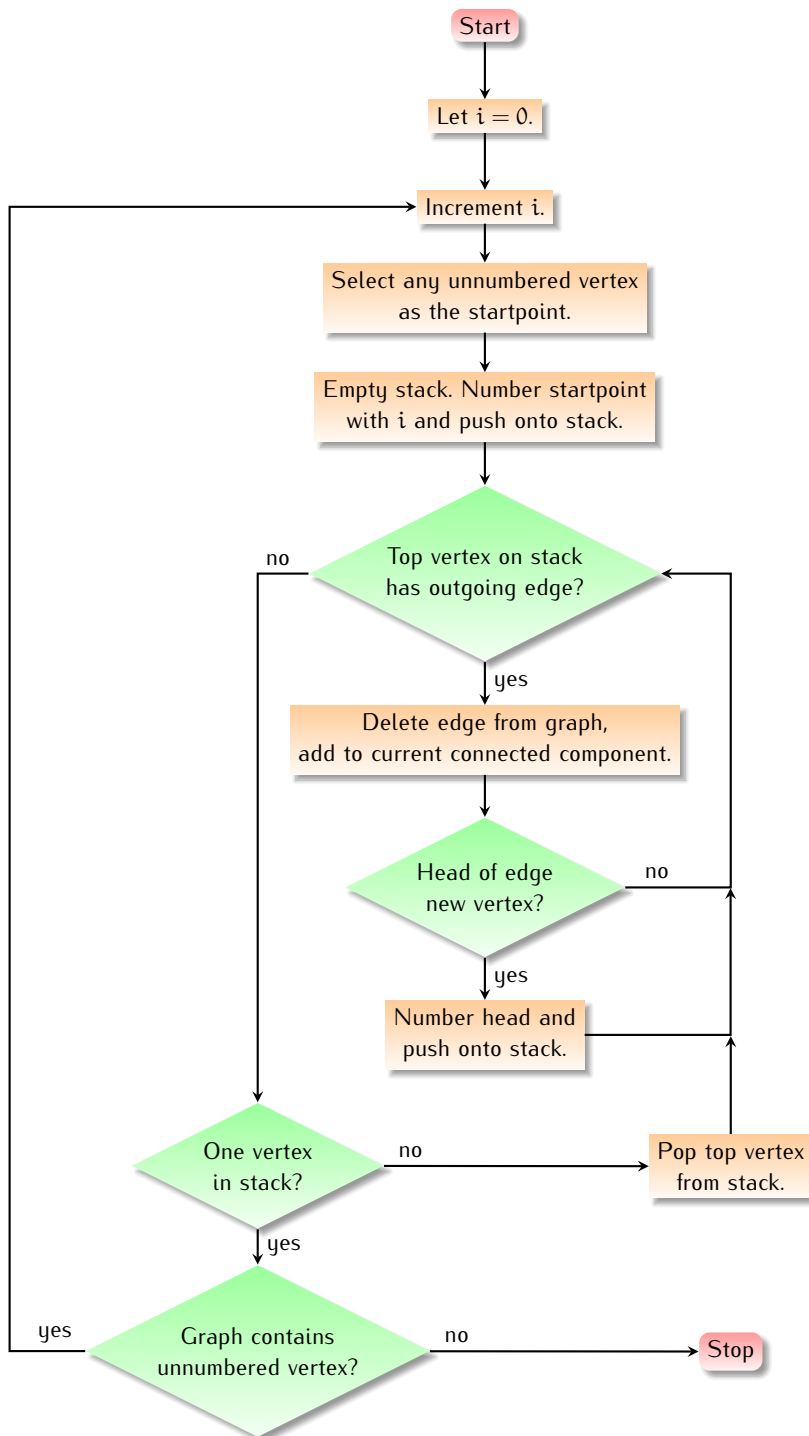


Figure 5. Flowchart for the connected components algorithm

RESOURCE	SENSE DISTRIBUTION						$\bar{\omega}$	$\hat{\omega}$
	1	2	3	4	≥ 5			
WN	83.4%	10.4%	3.1%	1.3%	1.8%	1.32	59	
WT	85.2%	9.4%	2.8%	1.1%	1.3%	1.26	58	
$A_c(\{WN, WP, WT\})$	91.0%	6.4%	1.6%	0.6%	0.5%	1.14	16	

Table 8. Word sense distribution in WordNet, Wiktionary, and the three-way conjoint alignment

Coverage of lexical items is not as straightforward to analyze owing to how Wikipedia treats them. Concepts in Wikipedia are canonically identified by an article title, which is typically a lexical item optionally followed by a parenthetical description which serves to disambiguate the concept from others which would otherwise share the same title. Lexical synonyms for the concept, however, are not explicitly and consistently encoded as they are in WordNet synsets. These synonyms are sometimes given in the unstructured text of the article, though identifying these requires sophisticated natural language processing. Many redirect page titles⁵ and incoming hyperlink texts—which are much easier to compile—are also synonyms, but others are anaphora or circumlocutions, and Wikipedia does not distinguish between them.

If we make no attempt to identify lexical synonyms from Wikipedia other than the article title, we find that the three-way conjoint alignment covers at least 44 803 unique lexical items, 42 165 of which are found in WordNet, 17 939 in Wiktionary, and 16 365 in Wikipedia. Moreover 20 609 of these lexical items are unique to WordNet and 2638 to Wikipedia. (There are no lexical items unique to Wiktionary.) We can also calculate the *sense distribution* of the conjoint alignment—that is, the percentage of lexical items which have a given number of senses k . Table 8 shows this distribution for WordNet, Wiktionary, and the conjoint three-way alignment; and also the average ($\bar{\omega}$) and maximum ($\hat{\omega}$) number of senses per lexical item.

We observe that while the distributions for the unaligned resources are similar, the conjoint alignment demonstrates a marked shift towards monosemy. Though Zipf’s law of meaning (Zipf, 1949) suggests that this might be the result of poor coverage of very high frequency lexical items, we found that the conjoint alignment actually covers 97 of the 100 most common (and 934 of the 1000 most common) nouns occurring in the British National Corpus (Clear, 1993).

Informal spot checks of synonym sets show them to be generally plausible, which is to be expected given the accuracy of the source

⁵ In Wikipedia parlance, a *redirect page* is an empty pseudo-article which simply refers the visitor to a different article. They are analogous to “see” cross-references in indices (Booth, 2001, p. 118–122).

alignments. However, the incidence of questionable or obviously incorrect mappings seems disproportionately higher in larger synonym sets. For example, one synonym set of cardinality 21 quite reasonably groups together various equivalent or closely related senses of the noun *hand*, but also includes senses for “Palm os” and “left-wing politics”, since in the two-way alignments they had been mistakenly aligned with the anatomical senses for *palm* and *left hand*, respectively. It appears that such errors are not only propagated but actually exaggerated by our algorithm, resulting in noisy data.

4.5 EVALUATION

There are several different ways in which sense alignments can be formally evaluated. The conceptually simplest is comparison with human judgments as Meyer and Gurevych (2011) and Matuschek and Gurevych (2013) did with their pairwise alignments. However, there are many reasons why this sort of evaluation is not appropriate for an alignment of more than two resources: First, it disregards the transitive nature of synonymy. That is, if the two-way alignment contains the pairs (n_1, k) and (n_2, k) , then those two pairs are considered in the evaluation, but not the implied pair (n_1, n_2) . This was perhaps more acceptable for the two-way alignments where only a small minority of the mappings are not 1:1, but our three-way alignments rely more heavily on the transitive property; indeed, in the conjoint alignment 100% of the synonym sets were produced by exploiting it. Second, even if we modify the evaluation setup such that the implied pairs are also considered, since the number of identifiers per synonym set is much higher in the three-way alignment, there is a combinatorial explosion in the number of candidate pairs for the judges to consider. Finally, considering sense pairs in isolation may not be the most appropriate way of evaluating what are essentially *clusters* of ostensibly synonymous sense descriptions.

We could therefore reduce the problem to one of evaluating clusters of senses from a single resource—that is, for every synonym set in the full alignment, we remove sense identifiers from all but one resource, and treat the remainder as a coarse-grained clustering of senses. Established intrinsic or extrinsic sense cluster evaluation techniques can then be applied. An example of the former would be computing the entropy and purity of the clusters with respect to a human-produced gold standard (Zhao and Karypis, 2003). However, while such gold standards have been constructed for early versions of WordNet (Agirre and Lopez de Lacalle, 2003; Navigli, 2006), they have not, to our knowledge, been produced for the more recent version used in our alignment. A possible extrinsic cluster evaluation would be to take the sense assignments of a state-of-the-art word

sense disambiguation (wSD) system and rescore them on clustered versions of the gold standard (Navigli, 2006; Snow *et al.*, 2007). That is, the system is considered to disambiguate a term correctly not only if it chooses the gold-standard sense, but also if it chooses any other sense in that sense's cluster. The improvement for using a given sense clustering is measured relative to a computed random clustering of equivalent granularity.

Cluster evaluations are appropriate if constructing the alignment is simply a means of decreasing the granularity of a single sense inventory. However, they do not measure the utility of the alignment as an LSR in its own right, which calls for extrinsic evaluations in scenarios where unaligned LSRs are normally used. One previous study (Ponzetto and Navigli, 2010) demonstrated marked improvements in accuracy of two different knowledge-based wSD algorithms when they had access to additional definition texts or semantic relations from a WordNet–Wikipedia alignment. Conventional wisdom in wSD is that for knowledge-based approaches, more data is always better, so a three-way alignment which provides information from Wiktionary as well could boost performance even further. A complication with this approach is that our alignment method produces coarse-grained synonym sets containing multiple senses from the same resource, and so without additional processing a wSD algorithm would not distinguish between them. For use with existing fine-grained data sets, such synonym sets could either be removed from the alignment, or else the wSD algorithm would need to be written in such a way that if it selects such a synonym set as the correct one, it performs an additional, finer-grained disambiguation within it.

In this study we performed two of the aforementioned types of wSD-based evaluations. The first evaluation is a cluster-based one where we rescore the results of existing wSD systems using the clusters induced by our three-way alignment; we describe the methodology and present the results in §4.5.1. In our second evaluation, we use our three-way alignment to enrich WordNet glosses with those from aligned senses in the other two resources, and then use our enriched sense inventory with a knowledge-based wSD algorithm; this is covered in §4.5.2. For both evaluations we use the freely available DKPro wSD framework (see Chapter 6).

4.5.1 Clustering of wSD results

4.5.1.1 Methodology

A common extrinsic method for evaluating sense clusterings is to take the raw assignments made by existing word sense disambiguation systems on a standard data set and then rescore them according to the clustering. That is, a system is considered to have correctly dis-

ambiguated a term not only if it chose a correct sense specified by the data set's answer key, but also if it chose any other sense in the same cluster as a correct one. Of course, any clustering whatsoever is likely to increase accuracy, simply by virtue of there being fewer answers for the system to choose among. To account for this, accuracy obtained with each clustering must be measured relative to that of a random clustering of equivalent granularity.⁶

The random clustering score for each instance in the data set can be determined mathematically. Snow *et al.* (2007) and Bhagwani *et al.* (2013) use

$$\text{score}_{\text{rand}}(i) = \sum_{\omega \in \Omega_i} \frac{|\omega|(|\omega| - 1)}{N_i(N_i - 1)}, \quad (4.7)$$

where Ω_i is the set of clusters over the $N_i = |I(w_i)|$ senses of a given term w_i , and $|\omega|$ is the number of senses in the cluster ω . However, this formula is accurate only when the gold standard specifies a single correct answer for the instance. In practice, wsd data sets can specify multiple possible correct senses for an instance, and a system is considered to have correctly disambiguated the target if it selected any one of these senses. The SENSEVAL-3 all-words corpus used by Snow *et al.* (2007) and Bhagwani *et al.* (2013) is such a data set (some 3.3% of the instances have two or more “correct” senses) so the scores they report underestimate the accuracy of the random baseline and inflate their clustering methods' reported improvement.

To arrive at a formula which works in the general case, consider that for an instance where the target word has N_i senses, $G_i = |D_G(i)|$ of which are correct in the given context, and one of which is an incorrectly chosen sense, the total number of ways of distributing these senses among the clusters is

$$N_i \binom{N_i - 1}{G_i} = \frac{N_i!}{G_i!(N_i - G_i - 1)!}. \quad (4.8)$$

Of these, the number of distributions which cluster the incorrectly chosen sense together with none of the correct senses is

$$\sum_{\omega \in \Omega_i} |\omega| \binom{N_i - |\omega|}{G_i} = \sum_{\omega \in \Omega_i} \frac{|\omega|(N_i - |\omega|)!}{G_i!(N_i - |\omega| - G_i)!}, \quad (4.9)$$

where the summation includes only those clusters where $N_i - |\omega| \geq G_i$. The probability that the incorrectly chosen sense is clustered together with at least one correct sense is therefore

$$\text{score}_{\text{rand}}(i) = 1 - \sum_{\omega \in \Omega_i} \frac{|\omega|(N_i - |\omega|)!(N_i - G_i - 1)!}{N_i!(N_i - |\omega| - G_i)!} \quad (4.10)$$

⁶ Controlling for granularity is vital, since it is trivial to construct clusterings which effect arbitrarily high wsd accuracy. Consider the extreme case where for each word, *all* the senses are clustered together; this clustering would have 100% wsd accuracy and thus easily beat an uncontrolled random baseline, but not a granularity-controlled one.

or, recast for ease of programmatic computation,

$$\text{score}_{\text{rand}}(i) = 1 - \sum_{\omega \in \Omega_i} \frac{|\omega| \prod_{i=0}^{G_i-1} (N_i - |\omega| - i)}{\prod_{i=0}^{G_i} (N_i - i)}. \quad (4.11)$$

For the case where there really is only one correct gold-standard answer, Formula 4.10 becomes

$$\begin{aligned} \text{score}_{\text{rand}}(i) &= 1 - \sum_{\omega \in \Omega_i} \frac{|\omega| (N_i - |\omega|)! (N_i - 1 - 1)!}{N_i! (N_i - |\omega| - 1)!} \\ &= 1 - \sum_{\omega \in \Omega_i} \frac{|\omega| (N_i - |\omega|)}{N_i (N_i - 1)} \\ &= \sum_{\omega \in \Omega_i} \frac{|\omega|}{N_i} - \sum_{\omega \in \Omega_i} \frac{|\omega| (N_i - |\omega|)}{N_i (N_i - 1)} \\ &= \sum_{\omega \in \Omega_i} \frac{|\omega| (|\omega| - 1)}{N_i (N_i - 1)}, \end{aligned} \quad (4.12)$$

which agrees with Formula 4.7 above.

4.5.1.2 Experiment and results

Like Snow *et al.* (2007), we use the raw sense assignments of the three top-performing systems in the SENSEVAL-3 English all-words disambiguation task (Snyder and Palmer, 2004): GAMBL (Decadt *et al.*, 2004), SenseLearner (Mihalcea and Faruque, 2004), and the Koç University system (Yuret, 2004). Whereas our alignment uses WordNet 3.0, the SENSEVAL-3 data set uses WordNet 1.7.1, so we use the wn-Map mappings (Daudé *et al.*, 2003) to convert the WordNet 1.7.1 synset offsets to WordNet 3.0 synset offsets. Furthermore, because some of the WordNet synset clusters induced by our alignment contain no one common lexical item, we “purify” these clusters by splitting them into smaller ones such that each synset in the cluster shares at least one lexical item with all the others. We tested two cluster purification approaches: in the first, we create a new cluster by taking from the original cluster all synsets containing its most common lexical item, and repeat this until the original cluster is empty. We refer to this technique as *most-frequent first*, or MFF. The second approach (*least-frequent first*, or LFF) works similarly, except that new clusters are constructed according to the *least* common lexical item.

The results of this evaluation using MFF and LFF clusters are shown in Tables 9 and 10, respectively. The table columns show, in order, the systems’ original accuracy scores,⁷ the accuracies rescored according to the WordNet clustering induced by our full three-way alignment, the accuracies rescored according to a random clustering of equivalent granularity, and the improvement of our clustering relative to

⁷ The slight difference in scores with respect to those reported in Snyder and Palmer (2004) is an artifact of the conversion from WordNet 1.7.1 to WordNet 3.0.

SYSTEM	BASE	MFF	RANDOM	Δ
GAMBL	65.21	69.04	68.57	+0.46
SenseLearner	64.72	68.06	68.11	-0.05
Koç	64.23	67.63	67.18	+0.49
average	64.72	68.24	67.95	+0.29

Table 9. SENSEVAL-3 WSD accuracy using our MFF-purified clusters and random clustering of equivalent granularity

SYSTEM	BASE	LFF	RANDOM	Δ
GAMBL	65.21	68.89	68.39	+0.50
SenseLearner	64.72	67.91	67.86	+0.05
Koç	64.23	67.61	67.07	+0.54
average	64.72	68.14	67.77	+0.36

Table 10. SENSEVAL-3 WSD accuracy using our LFF-purified clusters and random clustering of equivalent granularity

the random one. As can be seen, the effect of our clusters on system performance is practically indistinguishable from using the random clusterings. In fact, in no case was the difference significant according to 's (1947) test ($\chi^2 \leq 1.78$, $df = 1$, $\chi^2_{1,0.95} = 3.84$). By comparison, Snow *et al.* (2007) report a modest average improvement of 3.55 percentage points, though they do not report the results of any significance testing, and as mentioned above, their method for measuring the improvement over random clustering is in any case flawed. (We are unable to compare our results with those of Navigli (2006) as his automatically induced clusterings are not published.)

4.5.2 Enriched sense inventory for knowledge-based wsd

In this evaluation we attempted to measure the contribution of additional sense information from aligned senses to knowledge-based word sense disambiguation. First, we enriched the glosses of WordNet senses with those from their aligned Wiktionary and Wikipedia senses. (In the case of Wikipedia, we used the first paragraph of the article.) We then ran a popular knowledge-based wsd baseline, the simplified Lesk algorithm (Kilgarriff and Rosenzweig, 2000), on the aforementioned SENSEVAL-3 data set. This algorithm selects a sense for the target word solely on the basis of how many words the sense gloss and target word context have in common, so additional, accurate gloss information should help close the lexical gap and therefore increase both coverage and accuracy.

GLOSSES	C	P	R	F ₁
standard	26.85	69.23	18.59	29.30
enriched	29.17	67.26	19.62	30.38

Table 11. SENSEVAL-3 WSD accuracy using simplified Lesk, with and without alignment-enriched sense glosses

GLOSSES	C	P	R	F ₁
standard	98.61	53.46	52.71	53.08
enriched	98.76	51.07	50.44	50.75

Table 12. SENSEVAL-3 WSD accuracy using simplified extended Lesk with 30 lexical expansions, with and without alignment-enriched sense glosses

The results of this evaluation are shown in Table 11. As predicted, coverage increased somewhat. The overall increase in recall was modest but statistically significant (corrected McNemar’s $\chi^2 = 6.22$, $df = 1$, $\chi^2_{1,0.95} = 3.84$).

The fact that our enriched sense representations boosted the accuracy of this simple baseline motivated us to repeat the experiment with a state-of-the-art knowledge-based WSD system. For this we used the system described in Miller, Biemann, *et al.* (2012), a variant of the simplified extended Lesk algorithm (Banerjee and Pedersen, 2002) which enriches the context and glosses with lexical items from a distributional thesaurus. However, as can be seen in Table 12, recall decreased by 2.27 percentage points; this difference was also statistically significant (corrected McNemar’s $\chi^2 = 6.51$, $df = 1$, $\chi^2_{1,0.95} = 3.84$). It seems, therefore, that the additional gloss information derived from our alignment is not compatible with the lexical expansion technique.

To gain some insight as to why, or at least when, this is the case, we compared the instances incorrectly disambiguated when using the standard glosses but not when using the enriched glosses against the instances incorrectly disambiguated when using the enriched glosses but not when using the standard glosses. Both sets had about the same POS distribution. However, the words represented in the latter set were much rarer (an average of 178 occurrences in SemCor (Miller, Chodorow, *et al.*, 1994), versus 302 for the former set) and more polysemous (7.8 senses on average versus 6.5). The correct disambiguations in the latter set were also more likely to be the most frequent sense (MFS) for the given word, as tabulated in SemCor (71.6% MFS versus 63.3%). Using the enriched sense glosses seems to be slightly worse for shorter contexts—the corresponding second set of misclassified instances had an average sentence length of 123 tokens com-

pared to the other's 127. (By comparison, the average sentence lengths where both methods correctly or incorrectly disambiguated the target word were 135 and 131, respectively.)

4.6 CONCLUSION

In this chapter we described a straightforward technique for producing an n -way alignment of LSRs from arbitrary pairwise alignments, and applied it to the production of a three-way alignment of WordNet, Wikipedia, and Wiktionary. We examined the characteristics of this alignment and identified various approaches to formally evaluating it, along with their particular suitabilities and drawbacks. In the case of the clustering-based evaluation, we identified and corrected a significant flaw in the technique heretofore employed in the field.

Informal examination of the synonym sets in our conjoint alignment shows them to be generally correct, though in many cases existing errors in the source alignments were magnified. Extrinsic evaluation of our full alignment in wsd settings gave mixed results: whereas using the alignment to enrich sense definitions proved useful for a baseline wsd algorithm, the same enriched definitions confounded a more sophisticated approach and significantly decreased its performance. Similarly, use of the alignment to cluster WordNet senses did not show any measurable improvement over a random baseline. (However, as we will report in Chapter 5, the same clustering *does* prove useful in a pun disambiguation task.)

Given these inconsistent results, future work could be directed to refinement of the alignment technique to reduce the noise in the synonym sets. This could involve, for example, filtering outlier senses using text similarity measures similar to those used in the construction of the WordNet–Wiktionary alignment. Alternatively, we could try applying our original technique to pairwise alignments which are known to be more accurate (*i. e.*, with higher precision), as this would reduce the incidence of error cascades. We might also try other ways of using the alignment for knowledge-based wsd—in this evaluation we made use of the resources' glosses only, though of course each resource provides much richer lexical and semantic information which can be exploited.

Traditional approaches to word sense disambiguation rest on the assumption that there exists a single unambiguous communicative intention underlying every word in the document or speech act under consideration. However, there exists a class of language constructs known as *paronomasia*, or *puns*, in which homonymic (*i.e.*, coarse-grained) lexical-semantic ambiguity is a *deliberate* effect of the communication act. That is, the writer¹ intends for a certain word or other lexical item to be interpreted as simultaneously carrying two or more separate meanings. Though puns are a recurrent and expected feature in many discourse types, current word sense disambiguation systems, and by extension the higher-level natural language applications making use of them, are completely unable to deal with them.

In this chapter, we present our arguments for why computational detection and interpretation of puns are important research questions. We discuss the challenges involved in adapting traditional word sense disambiguation techniques to intentionally ambiguous text and perform an evaluation of these adaptations in a controlled setting. We also describe in detail our creation of a large data set of manually sense-annotated puns, including the specialised tool we have developed to apply the sense annotations.

5.1 MOTIVATION

Puns have been discussed in rhetorical and literary criticism since ancient times, and in recent years have increasingly come to be seen as a respectable research topic in traditional linguistics and the cognitive sciences (Delabastita, 1997a). It is therefore surprising that they have attracted very little attention in the fields of computational linguistics and natural language processing. What little research has been done is confined largely to computational mechanisms for pun generation (in the context of natural language generation for computational humour) and to computational analysis of phonological properties of puns (*e.g.*, Binsted and Ritchie, 1994, 1997; Hempelmann, 2003a,b; Ritchie, 2005; Hong and Ong, 2009; Kawahara, 2010). A fundamental task which has not yet been as widely studied is the automatic detection and identification of intentional lexical ambiguity—that is, given

¹ Puns can and do, of course, occur in spoken communication as well. Though much of what we cover in this chapter is equally applicable to written and spoken language, for the purposes of simplification we refer henceforth only to written texts.

a text, does it contain any lexical items which are used in a deliberately ambiguous manner, and if so, what are the intended meanings?

We consider these to be important research questions with a number of real-world applications. For example:

HUMAN-COMPUTER INTERACTION. It has often been argued that humour can enhance human-computer interaction (HCI) (Hempelmann, 2008), and at least one study (Morkes *et al.*, 1999) has already shown that incorporating canned humour into a user interface can increase user satisfaction without adversely affecting user efficiency. Interestingly, the same study found that some users of the humorous interface told jokes of their own to the computer. We posit that having the computer recognize a user's punning joke and produce a contextually appropriate response (which could be as simple as canned laughter or as complex as generating a similar punning joke in reciprocation) could further enhance the HCI experience.

SENTIMENT ANALYSIS. Sentiment analysis is a form of automated text analysis that seeks to identify subjective information in source materials. It holds particular promise in fields such as market research, where it is useful to track a population's attitude towards a certain person, product, practice, or belief, and to survey how people and organizations try to influence others' attitudes. As it happens, puns are particularly common in advertising, where they are used not only to create humour but also to induce in the audience a valenced attitude toward the target (Monnot, 1982; Valitutti *et al.*, 2008). (This attitude need not be positive—a commercial advertisement could use unflattering puns to ridicule a competitor's product, and a public service announcement could use them to discommend undesirable behaviour.) Recognising instances of such lexical ambiguity and understanding their affective connotations would be of benefit to systems performing sentiment analysis on persuasive texts.

MACHINE-ASSISTED TRANSLATION. Among of the most widely disseminated and translated popular discourses today are television shows and movies, which feature puns and other forms of wordplay as a recurrent and expected feature (Schröter, 2005). Puns pose particular challenges for translators, who need not only to recognize and comprehend each instance of humour-provoking ambiguity, but also to select and implement an appropriate translation strategy.² Future NLP systems could assist translators in flagging intentionally ambiguous words for special attention, and where they are not directly translatable (as is

² The problem is compounded in audio-visual media; often one or both of the pun's meanings appears in the visual channel, and thus cannot be freely substituted.

usually the case), the systems may be able to propose ambiguity-preserving alternatives which best match the original pun's double meaning.

DIGITAL HUMANITIES. Wordplay is a perennial topic of scholarship in literary criticism and analysis. Shakespeare's puns, for example, are one of the most intensively studied aspects of his rhetoric, with countless articles and even entire books (Wurth, 1895; Rubinstein, 1984; Keller, 2009) having been dedicated to their enumeration and analysis. It is not hard to imagine how computer-assisted detection, classification, and analysis of puns could help scholars in the digital humanities in producing similar surveys of other *œuvres*.

It would seem that an understanding of lexical semantics is necessary for any implementation of the above-noted applications. However, the only previous studies on computational detection and comprehension of puns that we are aware of focus on phonological and syntactic features. But for the fact that they are incapable of assigning multiple distinct meanings to the same target, word sense disambiguation algorithms could provide the lexical-semantic understanding necessary to process puns in arbitrary syntactic contexts. We are not, in fact, the first to suggest this—Mihalcea and Strapparava (2006) have also speculated that semantic analysis, such as via word sense disambiguation or domain disambiguation, could aid in the detection of humorous incongruity and opposition.

5.2 BACKGROUND AND RELATED WORK

5.2.1 Puns

A *pun* is a writer's use of a word in a deliberately ambiguous way, often to draw parallels between two concepts so as to make light of them. They are a common source of humour in jokes and other comedic works; there are even specialized types of jokes, such as the *feghoot* (Ritchie, 2004, p. 223) and *Tom Swifty* (Lippman and Dunn, 2000), in which a pun always occurs in a fixed syntactic or stylistic structure. Puns are also a standard rhetorical and poetic device in literature, speeches, slogans, and oral storytelling, where they can also be used non-humorously. Shakespeare, for example, is famous for his use of puns, which occur with high frequency even in his non-comedic works.³

³ Keller (2009) provides frequency lists of rhetorical figures in nine of Shakespeare's plays (four comedies, four tragedies, and one history). Puns, in the sense used in this thesis, were observed at a rate of 17.4 to 84.7 instances per thousand lines, or 35.5 on average.

Both humorous and non-humorous puns have been the subject of extensive study, which has led to insights into the nature of language-based humour and wordplay, including their role in commerce, entertainment, and health care; how they are processed in the brain; and how they vary over time and across cultures (*e.g.*, Monnot, 1982; Culler, 1988; Lagerwerf, 2002; Bekinschtein *et al.*, 2011; Bell *et al.*, 2011). Study of literary puns imparts a greater understanding of the cultural or historical context in which the literature was produced, which is often necessary to properly interpret and translate it (Delabastita, 1997b).

Humanists have grappled with the precise definition and classifications of puns since antiquity. Recent scholarship tends to categorize puns not into a single overarching taxonomy, but rather by using clusters of mutually independent features (Delabastita, 1997a). From the point of view of our particular natural language processing application, the most important features are homography and homophony. A *homographic* pun exploits distinct meanings of the same written word, and a *homophonic* pun exploits distinct meanings of the same spoken word. Puns can be homographic, homophonic, both, or neither, as the following examples illustrate:

- (14) A lumberjack's world revolves on its axes.
- (15) She fell through the window but felt no pane.
- (16) A political prisoner is one who stands behind her convictions.
- (17) The sign at the nudist camp read, "Clothed until April."

In (14), the pun on *axes* is homographic but not homophonic, since the two meanings ("more than one axe" and "more than one axis") share the same spelling but have different pronunciations. In (15), the pun on *pane* ("sheet of glass") is homophonic but not homographic, since the word for the secondary meaning ("feeling of injury") is properly spelled *pain* but pronounced the same. The pun on *convictions* ("strongly held beliefs" and "findings of criminal guilt") in (16) is both homographic and homophonic. Finally, the pun on *clothed* in (17) is neither homographic nor homophonic, since the word for the target meaning, *closed*, differs in both spelling and pronunciation. Such puns are commonly known as *imperfect* puns.

Other characteristics of puns particularly important for our work include whether they involve compounds, multiword expressions, or proper names, and whether the pun's multiple meanings involve multiple parts of speech. We elaborate on the significance of these characteristics in the following section.

5.2.2 Word sense disambiguation

An implicit assumption made by all wsd algorithms heretofore engineered is that the targets are used more or less unambiguously. That

is, while the sense inventory may give a multiplicity of senses for a word, at most one of them (or perhaps a small cluster of closely related senses) is correct when that word is used in a particular context. Where a wsd system does select multiple sense annotations for a given target, this is taken to mean that the target has a single coarse-grained meaning that subsumes those senses, or that the distinction between them is unimportant.

The assumption of unambiguity covers not only semantics but also syntax: it is assumed that each target has a single part of speech and lemma (*i.e.*, canonical form) which are known *a priori* or can be deduced with high accuracy using off-the-shelf natural language processing tools. The pool of candidate senses can therefore be restricted to those whose lexicalizations exactly match the target lemma and part of speech. No such help is available for puns, at least not in the general case. Take the following two examples:

(18) Tom moped.

(19) "I want a scooter," Tom moped.

In the first of these sentences, the word *moped* is unambiguously a verb with the lemma *mope*, and would be correctly recognized as such by any automatic lemmatizer and part-of-speech tagger. The *moped* of the second example is a pun, one of whose meanings is a form of the verb *mope* ("to sulk") and the other of which is the noun *moped* ("motorized scooter"). For such cases an automated pun identifier would therefore need to consider all possible lemmas for all possible parts of speech of the target word. The situation becomes even more onerous for heterographic and imperfect puns, which may require the use of pronunciation dictionaries, and application of phonological theories of punning, in order to recover the lemmas. However, as our research interests are in lexical semantics rather than phonology, we focus on puns which are homographic and monolexemic. This should allow us to investigate the problems of pun detection and identification in as controlled a setting as possible.

Homographic puns may be simpler to disambiguate than heterographic ones, at least insofar as identifying the target sense candidates requires only a morphological processor to obtain the lemma (*i.e.*, base form) of the target word, plus a standard machine-readable dictionary in which to look it up. Unlike heterographic and imperfect puns, there is no need for electronic pronunciation dictionaries, nor for the design or application of any phonological theories to recover the target word candidates. However, identification and disambiguation of any type of pun can be complicated when it involves proper names, compounds, or multiword expressions. These can require specialized linguistic preprocessing tools (compound splitters, parsers, *etc.*) or lexical-semantic resources (electronic encyclopedias, gazetteers, *etc.*). For the present work we therefore focus on homo-

graphic puns, with a preference for monolexemic ones which do not involve proper names.

5.2.3 Computational humour

There is some previous research on computational detection and comprehension of humour, though by and large it is not concerned specifically with puns; those studies which do analyze puns tend to have a phonological or syntactic rather than semantic bent. In this subsection we briefly review some prior work which is relevant to ours.

Yokogawa (2002) describes a system for detecting the presence of puns in Japanese text. However, this work is concerned only with puns which are both imperfect and ungrammatical, relying on syntactic cues rather than the lexical-semantic information we propose to use. Taylor and Mazlack (2004) describe an n -gram-based approach for recognizing when imperfect puns are used for humorous effect in a certain narrow class of English knock-knock jokes. Their focus on imperfect puns and their use of a fixed syntactic context makes their approach largely inapplicable to perfect puns in running text. Mihalcea and Strapparava (2005) treat humour recognition as a classification task, employing various machine learning techniques on humour-specific stylistic features such as alliteration and antonymy. Of particular interest is their follow-up analysis (Mihalcea and Strapparava, 2006), where they specifically point to their system's failure to resolve lexical-semantic ambiguity as a stumbling block to better accuracy, and speculate that deeper semantic analysis of the text, such as via word sense disambiguation or domain disambiguation, could aid in the detection of humorous incongruity and opposition.

The previous work which is perhaps most relevant to ours is that of Mihalcea, Strapparava, and Pulman (2010). They build a data set consisting of 150 joke set-ups, each of which is followed by four possible "punchlines", only one of which is actually humorous (but not necessarily due to a pun). They then compare the set-ups against the punchlines using various models of incongruity detection, including many exploiting knowledge-based semantic relatedness such as Lesk (1986). The Lesk model had an accuracy of 56%, which is lower than that of a naïve polysemy model which simply selects the punchline with the highest mean polysemy (66%) and even of a random-choice baseline (62%). However, it should be stressed here that the Lesk model did not directly account for the possibility that any given word might be ambiguous. Rather, for every word in the setup, the Lesk measure was used to select a word in the punchline such that the lexical overlap between each *one* of their possible definitions was maximized. The overlap scores for all word pairs were then averaged, and the punchline with the lowest average score selected as the most humorous.

5.2.4 Corpora

There are a number of English-language pun corpora which have been used in past work, usually in linguistics or the social sciences. In their work on computer-generated humour, Lessard *et al.* (2002) use a corpus of 374 Tom Swifities taken from the Internet, plus a well-balanced corpus of 50 humorous and non-humorous lexical ambiguities generated programmatically (Venour, 1999). Hong and Ong (2009) also study humour in natural language generation, using a smaller corpus of 27 punning riddles derived from a mix of natural and artificial sources. In their study of wordplay in religious advertising, Bell *et al.* (2011) compile a corpus of 373 puns taken from church marquees and literature, and compare it against a general corpus of 1515 puns drawn from Internet websites and a specialized dictionary (Crosbie, 1977). Zwicky and Zwicky (1986) conduct a phonological analysis on a corpus of “several thousand” puns, some of which they collected themselves from advertisements and catalogues, and the remainder of which were taken from previously published collections (Crosbie, 1977; Monnot, 1981; Sharp, 1984). Two studies on cognitive strategies used by second language learners (Kaplan and Lucas, 2001; Lucas, 2004) used a corpus of 58 jokes compiled from newspaper comics, 32 of which rely on lexical ambiguity. Bucaria (2004) conducts a linguistic analysis of a corpus of 135 humorous newspaper headlines, about half of which exploit lexical ambiguity.

Such corpora—particularly the larger ones—are good evidence that intentionally lexical ambiguous exemplars exist in sufficient numbers to make a rigorous evaluation of our proposed tasks feasible. Unfortunately, none of the above-mentioned corpora have been published in full, none of them are systematically sense-annotated, and many of them contain (sometimes exclusively) the sort of imperfect or otherwise heterographic puns which we mean to exclude from consideration. This has motivated us to produce our own corpus of puns, the construction and analysis of which is described in the following section.

5.3 DATA SET

As in traditional word sense disambiguation, a prerequisite for pun disambiguation is a corpus of positive examples where human annotators have already identified the ambiguous words and marked up their various meanings with reference to a given sense inventory. For pun detection, a corpus of negative examples is also required. In this section we briefly review the data sets which have been used in past work and describe the creation of our own.

5.3.1 Raw data

Our aim was to collect somewhere around 2000 puns in short contexts, as this number of instances is typical of testing data sets used in past WSD competitions such as SENSEVAL and SemEval (see §2.4.4). To keep the complexity of our disambiguation method and of our evaluation metrics manageable in this pilot study, we decided to consider only those examples meeting the following four criteria:

ONE PUN PER INSTANCE Of all the lexical units in the instance, one and only one may be a pun. Adhering to this restriction makes pun detection within contexts a binary classification task, which simplifies evaluation and leaves the door open for use of certain machine learning algorithms.

ONE CONTENT WORD PER PUN The lexical unit that forms the pun must consist of, or contain, only a single content word (*i. e.*, a noun, verb, adjective, or adverb), excepting adverbial particles of phrasal verbs. (For example, a pun on *car* is acceptable because it is a single content word, whereas a pun on *to* is not because it is not a content word. A pun on *ice cream* is unacceptable, because although it is a single lexical unit, it consists of two content words. A pun on the phrasal verb *put up with* meets our criteria: although it has three words, only one of them is a content word.) This criterion is important because, in our observations, it is usually only one word which carries ambiguity in puns on compounds and multiword expressions. Processing these cases would require the annotator (whether human or machine) to laboriously partition the pun into (possibly overlapping) sense-bearing units and to assign sense sets to each of them.

TWO MEANINGS PER PUN The pun must have exactly two distinct meanings. Though sources tend to agree that puns have only two senses (Redfern, 1984; Attardo, 1994), our annotators identified a handful of examples where the pun could plausibly be analyzed as carrying three distinct meanings. An example from the *Pun of the Day* website is as follows:

- (20) At a job interview, I decided to lie and say I had experience as an illusionist and as a window cleaner. They saw right through me.

Here the pun is on *saw* (*through*), which bears the following three meanings as given by WordNet:

SEE perceive by sight or have the power to perceive by sight (“You have to be a good observer to see all the details”; “Can you see the bird in that tree?”; “He is blind—he cannot see”)

SEE THROUGH perceive the true nature of (“We could see through her apparent calm”)

SAW cut with a saw (“saw wood for the fireplace”)

To simplify our manual annotation procedure and our evaluation metrics we excluded these rare outliers from our corpus.

WEAK HOMOGRAPHY While the WSD approaches we plan to evaluate would probably work for both homographic and heterographic puns, admitting the latter would require the use of pronunciation dictionaries and application of phonological theories of punning in order to recover the target lemmas (Hempelmann, 2003a). As our research interests are in lexical semantics rather than phonology, we focus for the time being on puns which are more or less homographic. More precisely, we stipulate that the lexical units corresponding to the two distinct meanings must be spelled exactly the same way, with the exception that particles and inflections may be disregarded. This somewhat softer definition of homography allows us to admit a good many morphologically interesting cases which were nonetheless readily recognized by our human annotators. An example is shown below:

(21) The marquis and the earl duked it out.

Here the pun is on the noun *duke* and an inflected form of the phrasal verb *duke it out*.

We began by pooling together some of the previously mentioned data sets, original pun collections made available to us by professional humorists, and freely available pun collections from the Web. After filtering out duplicates, these amounted to 7750 candidate instances, mostly in the form of short sentences. About half of these come from the *Pun of the Day* website, a quarter from the personal archive of author Stan Kegel, and the remainder from various private and published collections. We then employed human annotators to filter out all instances not meeting the above-noted criteria; this winnowed the collection down to 1652 positive instances. These range in length from 3 to 44 words, with an average length of 11.8.

For our corpus of negative examples, we followed Mihalcea and Strapparava (2005, 2006) and assembled a collection of 1164 proverbs, famous sayings, and witticisms from various Web sources. These are similar in style to our positive examples; most of them contain humour but none of them contain puns. They range in length from 2 to 27 words, with an average of 8.2, and 988 of them (85%) contain a word which was later annotated as the pun in at least one of the 1652 positive examples.

Select all puns:
(Annotating sentence: 0, from file: 1 example.txt)

" Where do river otters keep their money " " At the bank ! "

NO PUN
☐

MULTIPLE PUNS
☐

OTHER
☐

Annotator: Continue

Figure 6. Selecting pun words in Punnotator

A speaker at the firearms convention had to rifle through his notes .

Select the two sense sets of the word **rifle** - rifle (noun) & rifle (verb)

S1	S2	noun definitions
<input type="checkbox"/>	<input type="checkbox"/>	rifle (rifle) > a shoulder firearm with a long barrel and a rifled bore (e.g. he lifted the rifle to his shoulder and fired)
S1	S2	verb definitions
<input type="checkbox"/>	<input type="checkbox"/>	rifle (rifle, ransack, reave, foray, strip, despoil, plunder, pillage, loot) > steal goods, take as spoils (e.g. During the earthquake people looted the stores that were deserted by their owners)
<input type="checkbox"/>	<input type="checkbox"/>	rifle (rifle, go) > go through in search of something, search through someone's belongings in an unauthorized way (e.g. Who rifled through my desk drawers?)
S1	S2	Proper name - Unassigned
<input type="checkbox"/>	<input type="checkbox"/>	Proper Name
<input type="checkbox"/>	<input type="checkbox"/>	Unassigned - Unknown

Submit & NextEditor

Figure 7. Selecting definitions in Punnotator

5.3.2 Sense annotation

Manual linguistic annotation, and sense annotation in particular, is known to be a particularly arduous and expensive task (Mihalcea and Chklovski, 2003). The process can be sped up somewhat through the use of dedicated annotation support software. However, existing sense annotation tools, such as Stamp (Hovy *et al.*, 2006), SATANic (Passonneau, Baker, *et al.*, 2012), and WebAnno (Yimam *et al.*, 2013), and the annotated corpus formats they write, do not support specification of distinct groups of senses per instance. It was therefore necessary for us to develop our own sense annotation tool, along with a custom SENSEVAL-inspired corpus format.

Our annotation tool, Punnotator, runs as a Web application on a PHP-enabled server. It reads in a simple text file containing the corpus of instances to annotate and presents them to the user one at a time through their web browser. For each instance, the user is asked to select the pun's content word, or else to check one of several boxes in the event that the instance has no pun or is otherwise invalid (see Figure 6). Punnotator then determines all possible lemmas of the selected content word, retrieves their definitions from a sense inventory, and presents them to the user in a table (see Figure 7). Unlike with

```

<text id="churchpun.txt" annotator="Klaus">
  <word id="churchpun.txt_0" senses="1">Jesus</word>
  <word id="churchpun.txt_1" senses="1">is</word>
  <word id="churchpun.txt_2" senses="1">an</word>
  <word id="churchpun.txt_3" senses="1">investment</word>
  <word id="churchpun.txt_4" senses="1">that</word>
  <word id="churchpun.txt_5" senses="1">never</word>
  <word id="churchpun.txt_6" senses="1">loses</word>
  <word id="churchpun.txt_7" senses="2" lemma="interest"
    first_sense="interest%1:07:02:":"
    second_sense="interest%1:21:00:":">interest</word>
  <word id="churchpun.txt_8" senses="1">.</word>
</text>

```

Figure 8. Punnotator XML output (excerpt)

traditional sense annotation tools, definitions from all parts of speech are provided, since puns often cross parts of speech.

The definition table includes two columns of checkboxes representing the two distinct meanings for the pun. In each column, the user checks all those definitions which correspond to one of the pun's two meanings. It is possible to select multiple definitions per column, which indicates that the user believes them to be indistinguishable or equally applicable for the intended meaning. The only restriction is that the same definition may not be checked in both columns. Following SENSEVAL practice, if one or both of the meanings of the pun are not represented by any of the listed definitions, the user may check one of two special checkboxes at the bottom of the list to indicate that the meaning is a proper name or otherwise missing from the sense inventory.

We elected to use the latest version of WordNet (3.1) as the sense inventory for our annotations. Though WordNet has often been criticized for the overly fine granularity of its sense distinctions (Ide and Wilks, 2007), it has the advantage of being freely available, of being the *de facto* standard LSR for use in WSD evaluations, and of being accessible through a number of flexible and freely available software libraries.

Punnotator writes its output to an XML-based corpus format loosely based on the SENSEVAL and SemEval formats but with support for multiple groups of senses per target word (see Figure 8).

5.3.3 Analysis

We employed three trained human judges—all of whom were native English-speaking graduate students in computer science or computational linguistics—to produce our manually annotated data set of 1652 positive instances. Two of the judges independently sense-

annotated the instances, while a third adjudicated all those disagreements which could not be resolved automatically. The final versions of the annotation and adjudication guidelines are reproduced in Appendices A and B, respectively.

The two annotators agreed on which word was the pun in 1634 cases, a raw agreement of 98.91%. For these 1634 cases, we measured inter-annotator agreement on the sense assignments using 's (1980) α . Our distance metric for α is a straightforward adaptation of the MASI set comparison metric (Passonneau, 2006). Whereas standard MASI, $d_M(s_i, s_j)$, compares two annotation sets s_i and s_j (see Formula 2.12 on p. 27), our annotations take the form of unordered *pairs* of sets $\{s_i^1, s_i^2\}$ and $\{s_j^1, s_j^2\}$. We therefore find the mapping between elements of the two pairs that gives the lowest total distance, and halve it:

$$d_{M'}(\{s_i^1, s_i^2\}, \{s_j^1, s_j^2\}) = \frac{1}{2} \min(d_M(s_i^1, s_j^1) + d_M(s_i^2, s_j^2), d_M(s_i^1, s_j^2) + d_M(s_i^2, s_j^1)). \quad (5.1)$$

With this method we observe $\alpha = 0.777$; this is only slightly below the 0.8 threshold recommended by Krippendorff, and far higher than what has been reported in other sense annotation studies (Passonneau, Habash, *et al.*, 2006; Jurgens and Klapaftis, 2013).

Where possible, we resolved sense annotation disagreements automatically by taking the intersection of corresponding sense sets. For cases where the annotators' sense sets were disjoint or contradictory (including the cases where the annotators disagreed on the pun word), we had our third human judge independently resolve the disagreement in favour of one annotator or the other. This left us with 1607 instances; at the time of writing we are planning on releasing the annotated data set as part of a shared task on pun disambiguation. Following are our observations on the qualities of the annotated corpus:

SENSE COVERAGE. Of the 1607 instances in the corpus, the annotators were able to successfully annotate both sense sets for 1298 (80.8%). For 303 instances (18.9%), WordNet was found to lack entries for only one of the sense sets, and for the remaining 6 instances (0.4%), WordNet lacked entries for both sense sets. By comparison, in the SENSEVAL and SemEval corpora the proportion of target words with unknown or unassignable senses ranges from 1.7 to 6.8%. This difference can probably be explained by the differences in genre: WordNet was constructed by annotating a subset of the Brown Corpus, a million-word corpus of American texts published in 1961 (Miller, Leacock, *et al.*, 1993). The Brown Corpus samples a range of genres, including journalism and technical writing, but not joke books. The SENSEVAL and SemEval data sets tend to use the same sort of news

and technical articles found in the Brown Corpus, so it is not surprising that a greater proportion of their words' senses can be found in WordNet.

Our 2596 successfully annotated sense sets have anywhere from one to seven senses each, with an average of 1.08. As expected, then, WordNet's sense granularity proved to be somewhat finer than necessary to distinguish between the senses in our data set, though only marginally so.

PART OF SPEECH DISTRIBUTION. Of the 2596 successfully annotated sense sets, 50.2% contain noun senses only, 33.8% verb senses only, 13.1% adjective senses only, and 1.6% adverb senses only. As previously noted, however, the semantics of puns sometimes transcends part of speech: 1.3% of our individual sense sets contain some combination of senses representing two or three different parts of speech, and of the 1298 instances where both meanings were successfully annotated, 297 (22.9%) have sense sets of differing parts of speech (or combinations thereof). This finding confirms the concerns we raised in §5.2.2 that pun disambiguators, unlike traditional wsd systems, cannot rely on the output of a part-of-speech tagger to narrow down the list of sense candidates.

POLYSEMY. Because puns have no fixed part of speech, each target term in the data set can have more than one "correct" lemma. An automatic pun disambiguator must therefore consider all possible senses of all possible lemmas of a given target. The annotated senses for each target in our data set represent anywhere from one to four different lemmas (without distinction of part of speech), with a mean of 1.2. The number of candidate senses associated with these lemma sets ranges from 1 to 79, with a mean of 12.4.

Of course, a real-world pun disambiguator will not know *a priori* which lemmas are the correct ones for a given target in a given context. On our data set such a system must select lemmas and senses from a significantly larger pool of candidates (on average 1.5 lemmas and 14.2 senses per target). Recall that on average, only 1.08 of these senses are annotated as "correct" in any given sense set.

TARGET LOCATION. During the annotation process it became obvious that the vast majority of puns were located towards the end of the context. As this sort of information could prove helpful to a disambiguation system, we calculated the frequency of target words occurring in the first, second, third, and fourth quarters of the contexts. As predicted, we found that the final quarter of the context is the overwhelmingly preferred pun location

(82.8% of instances), followed distantly by the third (9.3%), second (6.7%) and first (1.2%). This observation accords with previous empirical studies of large joke corpora which found that the punchline occurs in a terminal position more than 95% of the time (Attardo, 1994, ch. 2).

5.4 PUN IDENTIFICATION

5.4.1 Task description

Computational processing of puns involves two separate tasks: In *pun detection*, the objective is to determine whether or not a given context contains a pun, or more precisely whether any given word in a context is a pun. In *pun identification* (or *pun disambiguation*), the objective is to identify the two meanings of a term previously detected, or simply known *a priori*, to be a pun.

Recall from Chapter 2 that in traditional word sense disambiguation, we are given a sense inventory $I : L \rightarrow \mathcal{P}_{\geq 1}(S)$ that associates every word w in a lexicon L with a set of senses $I(w) \in S$, and the task is to produce a set-valued mapping $D : \mathbb{N} \rightarrow \mathcal{P}_{\geq 1}(S)$ which takes a context word w_i and finds the subset $D(i) \subseteq I(w_i)$ that best fits the context. For the case that w_i is a pun with n distinct meanings, the mapping must return a disjoint subset of $I(w_i)$ for each meaning—that is, $D : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{P}_{\geq 1}(S)$, where $\bigcap_{j=1}^n D(i, j) = \emptyset$.

5.4.2 Approach

To understand how traditional word sense disambiguation methods can be adapted to pun identification, recall that they work by attempting to assign a single sense to a given target. If they fail to make an assignment, this is generally for one of the following reasons:

1. The target word does not exist in the sense inventory.
2. The knowledge sources available to the algorithm (including the context and information provided by the sense inventory) are insufficient to link any one candidate sense to the target.
3. The sense information provided by the sense inventory is too fine-grained to distinguish between closely related senses.
4. The target word is used in an intentionally ambiguous manner, leading to indecision between coarsely related or unrelated senses.

We hold that for this last scenario, a disambiguator's inability to discriminate senses should not be seen as a failure condition, but rather

as a limitation of the WSD task as traditionally defined. By reframing the task so as to permit the assignment of multiple senses (or groups thereof), we can allow disambiguation systems to sense-annotate intentionally ambiguous constructions such as puns.

Many approaches to WSD involve computing some score for all possible senses of a target word, and then selecting the single highest-scoring one as the “correct” sense. The most straightforward modification of these techniques to pun disambiguation, then, is to have the systems select the *two* top-scoring senses, one for each meaning of the pun. Accordingly we applied this modification to the following knowledge-based WSD algorithms, which we presented in detail in Chapter 3:

SIMPLIFIED LESK (Kilgarriff and Rosenzweig, 2000) disambiguates a target word by examining the definitions⁴ for each of its candidate senses and selecting the single sense—or in our case, the two senses—which have the greatest number of words in common with the context. As we previously demonstrated that puns often transcend part of speech, our set of candidate senses is constructed as follows: we apply a morphological analyzer to recover all possible lemmas of the target word without respect to part of speech, and for each lemma we add all its senses to the pool.

SIMPLIFIED EXTENDED LESK (Ponzetto and Navigli, 2010) is similar to simplified Lesk, except that the definition for each sense is concatenated with those of neighbouring senses in WordNet’s semantic network.

SIMPLIFIED LEXICALLY EXPANDED LESK (see Chapter 3) is also similar to simplified Lesk, with the extension that every word in the context and sense definitions is expanded with up to 100 entries from a large distributional thesaurus.

The above algorithms fail to make a sense assignment when more than two senses are tied for the highest lexical overlap, or when there is a single highest-scoring sense but multiple senses are tied for the second-highest overlap. We therefore devised two pun-specific tie-breaking strategies. The first is motivated by the informal observation that, though the two meanings of a pun may have different parts of speech, at least one of the parts of speech is grammatical in the context of the sentence, and so would probably be the one assigned by a stochastic or rule-based POS tagger. Our “pos” tie-breaker therefore preferentially selects the best sense, or pair of senses, whose part of speech matches the one applied to the target by the Stanford POS tagger (Toutanova *et al.*, 2003).

⁴ In our implementation, the sense definitions are formed by concatenating the synonyms, gloss, and example sentences provided by WordNet.

For our second tie-breaking strategy, we posit that since humour derives from the resolution of semantic incongruity (Raskin, 1985; Attardo, 1994), puns are more likely to exploit coarse-grained sense distinctions than fine-grained systematic polysemy. We therefore induce two clusterings of WordNet senses by aligning the resource to more coarse-grained LSRS. One of these clusterings is produced with the approach described in Chapter 4, where pairwise WordNet–Wiktionary and WordNet–Wikipedia alignments are merged into a three-way alignment. The second clustering is based on a Dijkstra-wsa alignment of WordNet and OmegaWiki (Meijssen, 2009), a pairwise alignment which we have elsewhere shown to significantly boost wsd performance in a clustering evaluation (Matuschek, Miller, *et al.*, 2014). Our two “cluster” fallbacks work the same as the “pos” one, with the addition that any remaining ties among senses with the second-highest overlap are resolved by preferentially selecting those which are not in the same induced cluster as, and which in WordNet’s semantic network are at least three edges distant from, the sense with the highest overlap.

5.4.3 Evaluation methodology

5.4.3.1 Scoring

Recall from §2.4.2 that in traditional word sense disambiguation, *in vitro* evaluations are conducted by comparing the senses assigned by the disambiguation system to the gold-standard senses assigned by the human annotators. For the case that the system and gold-standard assignments consist of a single sense each, the exact match criterion is used: the system receives a score of 1 if it chose the sense specified by the gold standard, and 0 otherwise. Where the system selects a single sense for an instance for which there is more than one correct gold standard sense, the multiple tags are interpreted disjunctively—that is, the system receives a score of 1 if it chose any one of the gold-standard senses, and 0 otherwise. Overall performance is reported in terms of coverage (the number of targets for which a sense assignment was attempted), precision (the sum of scores divided by the number of attempted targets), recall (the sum of scores divided by the total number of targets in the data set), and F_1 (the harmonic mean of precision and recall) (Palmer, Ng, *et al.*, 2007).

The traditional approach to scoring individual targets is not usable as-is for pun disambiguation, because each pun carries two disjoint but equally valid sets of sense annotations. Instead, we count an item as correct (scoring 1) only if each chosen sense set is a subset of one of the gold-standard sense sets, and no two gold-standard sense sets contain the same chosen sense. That is, if the gold standard sense

sets for item i are $D_G(i, 1)$ and $D_G(i, 2)$, and the system's sense assignments are $D_Y(i, 1)$ and $D_Y(i, 2)$, then the system score is

$$\begin{aligned} \text{score}(i) = & [(D_Y(i, 1) \subseteq D_G(i, 1) \wedge D_Y(i, 2) \subseteq D_G(i, 2) \wedge \\ & D_Y(i, 1) \not\subseteq D_G(i, 2) \wedge D_Y(i, 2) \not\subseteq D_G(i, 1)) \vee \\ & (D_Y(i, 1) \subseteq D_G(i, 2) \wedge D_Y(i, 2) \subseteq D_G(i, 1) \wedge \\ & D_Y(i, 1) \not\subseteq D_G(i, 1) \wedge D_Y(i, 2) \not\subseteq D_G(i, 2))]. \end{aligned} \quad (5.2)$$

As with traditional wsd scoring, various approaches could be used to assign credit for partially correct assignments, though we leave exploration of these to future work.

5.4.3.2 Baselines

System performance in wsd is normally interpreted with reference to one or more baselines. To our knowledge, ours is the very first study of automatic pun disambiguation on any scale, so at this point there are no previous systems against which to compare our results. However, traditional wsd systems are often compared with two naïve baselines (Gale, 1992) which can be adapted for our purposes.

The first of these naïve baselines is to randomly select from among the candidate senses. In §2.4.3 we introduced the concept of a random disambiguator which selects a single sense for each target. Its accuracy is the number of gold-standard senses divided by the number of candidate senses, averaged across the entire data set:

$$P_{\text{rand}} = R_{\text{rand}} = \frac{1}{n} \sum_{i=1}^n \frac{|D_G(w_i)|}{|I(w_i)|}. \quad (2.20)$$

In our pun disambiguation task, however, a random disambiguator must select *two* senses—one for each of the sense sets $D_G(i, 1)$ and $D_G(i, 2)$ —and these senses must be distinct. There are $\binom{|I(w_i)|}{2}$ possible ways of selecting two unique senses for a word w_i , so the random baseline accuracy is

$$P_{\text{rand}} = R_{\text{rand}} = \frac{1}{n} \sum_{i=1}^n \frac{|D_G(i, 1)| \cdot |D_G(i, 2)|}{\binom{|I(w_i)|}{2}}. \quad (5.3)$$

The second naïve baseline for wsd, known as *most frequent sense* (MFS), is a supervised baseline, meaning that it depends on a manually sense-annotated background corpus. As its name suggests, it involves always selecting from the candidates that sense which has the highest frequency in the corpus. As with our test algorithms, we adapt this technique to pun disambiguation by having it select the two most frequent senses (according to WordNet's built-in sense frequency counts). In traditional wsd, MFS baselines are notoriously difficult to beat, even for supervised disambiguation systems, and since

SYSTEM	C	P	R	F ₁
SL	35.52	19.74	7.01	10.35
SEL	42.45	19.96	8.47	11.90
SLEL	98.69	13.43	13.25	13.34
SEL+POS	59.94	21.21	12.71	15.90
SEL+cluster ₃	66.33	20.67	13.71	16.49
SEL+cluster ₂	68.10	20.70	14.10	16.77
random	100.00	9.31	9.31	9.31
MFS	100.00	13.25	13.25	13.25

Table 13. Coverage, precision, recall, and F₁ for various pun disambiguation algorithms

they rely on expensive sense-tagged data they are not normally considered a benchmark for the performance of knowledge-based disambiguators.

5.4.4 Results

Using the freely available DKPro WSD framework (see Chapter 6), we implemented our pun disambiguation algorithms, ran them on our full data set, and compared their annotations against those of our manually produced gold standard. Table 13 shows the coverage, precision, recall, and F₁ for simplified Lesk (SL), simplified extended Lesk (SEL), simplified lexically expanded Lesk (SLEL), and the random and most frequent sense baselines; for SEL we also report results for each of our pun-specific tie-breaking strategies. (Here cluster₂ refers to the clustering produced with the two-way alignment to Omega-Wiki, and cluster₃ to that produced with the three-way alignment from Chapter 4.) All metrics are reported as percentages, and the highest score for each metric (excluding baseline coverage, which is always 100%) is highlighted in boldface.

Accuracy for the random baseline annotator was about 9%; for the MFS baseline it was just over 13%. These figures are considerably lower than what is typically seen with traditional WSD corpora, where random baselines achieve accuracies of 30 to 60%, and MFS baselines 65 to 80% (Palmer, Fellbaum, *et al.*, 2001; Snyder and Palmer, 2004; Navigli, Litkowski, *et al.*, 2007). Our baselines' low figures are the result of them having to consider senses from every possible lemmatization and part of speech of the target, and underscore the difficulty of our task.

The conceptually simplest knowledge-based algorithm we tested, simplified Lesk, was over twice as accurate as the random baseline

in terms of precision (19.74%), but predictably had very low coverage (35.52%), leading in turn to very low recall (7.01%). Manual examination of the unassigned instances confirmed that failure was usually due to the lack of any lexical overlap whatsoever between the context and definitions. The use of a tie-breaking strategy would not help much here, though some way of bridging the lexical gap would. This is, in fact, the strategy employed by the extended and lexically expanded variants of simplified Lesk, and we observed that both were successful to some degree. Simplified lexically expanded Lesk almost completely closed the lexical gap, with nearly complete coverage (98.69%), though this came at the expense of a large drop in precision (to 13.43%). Given the near-total coverage, use of a tie-breaking strategy here would have no appreciable effect on the accuracy.

Simplified extended Lesk, on the other hand, saw significant increases in coverage, precision, and recall (to 42.45%, 19.96%, and 8.47%, respectively). Its recall is statistically indistinguishable⁵ from the random baseline, though spot-checks of its unassigned instances show that the problem is very frequently not the lexical gap but rather multiple senses tied for the greatest overlap with the context. We therefore tested our two pun-specific backoff strategies to break this system's ties. Using the "pos" strategy increased coverage by 41%, relatively speaking, and gave us our highest observed precision of 21.21%. Our two "cluster" strategies effected a relative increase in coverage of about 60%, with "cluster₂" giving us the overall best recall (14.10%). The latter strategy also had the best tradeoff between precision and recall, with an F_1 of 16.77%.

Significance testing shows the recall scores for SLEL, SEL+POS, and both SEL+cluster approaches to be significantly better than the random baseline, and statistically indistinguishable from that of the most frequent sense baseline. This is excellent news, especially in light of the fact that supervised approaches (even baselines like MFS) usually outperform their knowledge-based counterparts. Though the three knowledge-based systems are not statistically distinguishable from each other in terms of recall, they do show a statistically significant improvement over SL and SEL, and the two implementing pun-specific tie-breaking strategies were markedly more accurate than SLEL for those targets where they attempted an assignment. These two systems would therefore be preferable for applications where precision is more important than recall.

We also examined the results of our generally best-performing system, SEL+cluster₂, to see whether there was any relationship with the targets' part of speech. We filtered the results according to whether both gold-standard meanings of the pun contain senses for nouns

⁵ All significance statements in this section are based on 's (1947) test at a confidence level of 5%.

POS	C	P	R	R _{rand}
noun	66.60	20.89	13.91	10.44
verb	65.61	14.54	9.54	5.12
adj.	68.87	39.73	27.36	16.84
adv.	100.00	75.00	75.00	46.67
pure	66.77	21.44	14.31	9.56
mult.	72.58	18.43	13.38	12.18

Table 14. Coverage, precision, and recall for *SEL+cluster₂*, and random baseline recall, according to part of speech

only, verbs only, adjectives only, or adverbs only; these amounted to 539, 346, 106, and 8 instances, respectively. These results are shown in Table 14. Also shown there is a row which aggregates the 999 targets with “pure” pos, and another for the remaining 608 instances (“mult.”), where one or both of the two meanings contain senses for multiple parts of speech, or where the two meanings have different parts of speech. The last column of each row shows the recall of the random baseline for comparison.

Accuracy was lowest on the verbs, which had the highest candidate polysemy (21.6) and are known to be particularly difficult to disambiguate even in traditional WSD. Still, as with all the other single parts of speech, performance of *SEL+cluster₂* exceeded the random baseline. While recall was lower on targets with mixed pos than those with pure pos, coverage was significantly higher. Normally such a disparity could be attributed to a difference in polysemy: Lesk-like systems are more likely to attempt a sense assignment for highly polysemous targets, since there is a greater likelihood of one of the candidate definitions matching the context, though the probability of the assignment being correct is reduced. In this case, however, the multi-pos targets actually had lower average polysemy than the single-pos ones (13.2 *vs.* 15.8).

5.5 PUN DETECTION

Pun detection is the task of determining whether or not any given context, or more precisely any given word in a context, contains a pun. For the case where a context is already known *a priori* to contain a pun, we can speak of the task of *pun location*, which is to determine which word or words within the context carry the intentional ambiguity.

More formally, the problems can be cast as binary classification tasks. Given a context of words $T = (w_1, w_2, \dots, w_n)$, the pun location task and the fine-grained variant of the pun detection task are to classify each $w \in T$ as either “pun” or “non-pun”. (The only difference between the two tasks is that in pun location, the system *must* apply a positive classification to exactly one of the $w \in T$, whereas in pun detection no such constraint is placed on the classifier.) In the coarse-grained variant of the pun detection task, a single “pun”/“non-pun” classification is applied to the entire context T .

In this section, we propose an evaluation methodology for the tasks of pun detection and location and present the results of some naïve baselines.

5.5.1 Evaluation methodology

Word sense disambiguation borrows its evaluation metrics from the field of information retrieval (IR), but modifies them slightly to account for disambiguators which decline to hazard a guess on certain instances (Cohn, 2003). In pun location and detection, however, the goal is to exhaustively classify all contexts, or all context words, as containing or not containing a pun. For these tasks we therefore use the IR metrics in their original forms (Manning, Raghavan, *et al.*, 2008, §8.3).

For a data set containing both positive and negative examples, we can score two variants of the pun detection task. In the coarse-grained variant, the system must simply mark all those instances it believes to contain a pun. In the fine-grained variant, the classification labels are applied to individual words rather than entire instances. The pun location task is the same as the fine-grained pun detection task, except that we exclude from the data set all those instances that do not contain any pun. For any of these tasks, we can construct a contingency matrix, as in Table 15 below, counting all possible classification outcomes:

PREDICTED CLASS	ACTUAL CLASS	
	PUN	NON-PUN
PUN	true positives (TP)	false positives (FP)
NON-PUN	false negatives (FN)	true negatives (TN)

Table 15. Classification contingency matrix

We can then define *precision* as the fraction of marked instances (or words) which actually do contain a pun:

$$P = \frac{TP}{TP + FP}. \quad (5.4)$$

Recall is the fraction of instances (or words) containing a pun which the system actually marked:

$$R = \frac{TP}{TP + FN}. \quad (5.5)$$

As in *WSD*, precision and recall measure orthogonal qualities of the classification task, so it is common to combine them into the F_1 score as in Formula 2.19. Note that in *IR*, unlike in *WSD*, *accuracy* is not a synonym for recall, but rather a metric of its own that reports the proportion of correctly classified items:

$$A = \frac{TP + TN}{TP + FP + FN + TN}. \quad (5.6)$$

Accuracy is not an appropriate measure for our tasks because of the extremely skewed data. That is, in our data set, as in real-world texts, non-puns outnumber puns by several orders of magnitude. It is therefore trivial to produce a high-accuracy classifier; it need only mark all words as non-puns.

5.5.2 Baselines and results

As in *WSD* and pun disambiguation, performance in the pun detection and location tasks can be measured with respect to a calculated random baseline. In the case of the coarse- and fine-grained pun detection tasks, this baseline simulates the effect of randomly deciding whether or not a given instance or word, respectively, contains a pun. Its recall is fixed at 0.5, and its precision is equal to the proportion of instances (or words) in the data set containing puns:

$$P_{\text{rand}} = \frac{TP + FN}{TP + TN + FP + FN}. \quad (5.7)$$

For the pun location task, where it is known *a priori* that every instance contains one and only one pun, the baseline approach randomly selects one of the words as the pun. For a set of n contexts $\{T_1, T_2, \dots, T_n\}$, each containing $|T_i|$ words, its precision and recall are therefore

$$P_{\text{rand}} = R_{\text{rand}} = \frac{1}{n} \sum_{i=1}^n \frac{1}{|T_i|}. \quad (5.8)$$

In this section we also test a slightly more sophisticated pun location baseline inspired by Mihalcea, Strapparava, and Pulman (2010). In that study, genuine joke punchlines are selected among several non-humorous alternatives by finding the candidate whose words have the highest mean polysemy. We adapt this technique by selecting as the pun the word with the highest polysemy (counting together

TASK	BASELINE	P	R	F ₁
pun detection (coarse)	random	57.99	50.00	53.70
pun detection (fine)	random	5.57	50.00	10.03
pun location	random	8.46	50.00	14.45
pun location	highest polysemy	17.98	17.98	17.98

Table 16. Baseline results for the pun detection and location tasks

senses from all parts of speech). In case of a tie, we choose the word nearest to the end of the context, as we had earlier observed (see §5.3.3) that puns tend to occur in terminal position.

Table 16 shows the results for our three baselines when run on our combined corpus of 1607 positive and 1164 negative instances. There is not much to be said about the scores for the coarse- and fine-grained pun detection tasks; they simply reflect the proportion of positive instances in the data set (considered as monolithic contexts or individual words, respectively). Nonetheless, they do give a rough idea of the difficulty of the task and set a lower bound for the performance of more sophisticated systems.

The pun location results are of slightly greater interest since we have two algorithms to compare. We can observe that the “highest polysemy” baseline already provides significant gains over random selection, with precision more than doubling.

5.6 CONCLUSION

In this chapter we have introduced the novel task of pun disambiguation and have proposed and evaluated several computational approaches for it. The major contributions of this work are as follows: First, we have produced a new data set consisting of manually sense-annotated homographic puns. The data set is large enough, and the manual annotations reliable enough, for a principled evaluation of automatic pun disambiguation systems. Second, we have shown how evaluation metrics, baselines, and disambiguation algorithms from traditional WSD can be adapted to the task of pun disambiguation, and we have tested these adaptations in a controlled experiment. The results show pun disambiguation to be a particularly challenging task for NLP, with baseline results far below what is commonly seen in traditional WSD. We showed that knowledge-based disambiguation algorithms naïvely adapted from traditional WSD perform poorly, but that extending them with strategies that rely on pun-specific features brings about dramatic improvements in accuracy: their recall

becomes comparable to that of a supervised baseline, and their precision greatly exceeds it.

We also introduced the novel task of pun detection, which would be a prerequisite for pun disambiguation in any real-world application. We have laid the groundwork for future research into this problem by constructing a data set of both positive and negative examples, and by providing an evaluation methodology and results of two simple baselines.

There are a number of avenues we intend to explore in future work on pun disambiguation. For one, we would like to try adapting and evaluating some additional wsd algorithms for use with puns. Though our data set is probably too small to use with machine learning-based approaches, we are particularly interested in testing knowledge-based disambiguators which rely on measures of graph connectivity rather than gloss overlaps. We would also like to investigate alternative tie-breaking strategies, such as the domain similarity measures used by Mihalcea, Strapparava, and Pulman (2010) in their work on incongruity detection.

With respect to pun detection, there are a number of approaches we would like to investigate. One knowledge-based approach we might try follows from our aforementioned technique for pun disambiguation: To determine whether or not a given word in context is a pun, run it through a high-precision wsd system and make a note of the differences in scores between the top two or three semantically dissimilar sense candidates. For unambiguous targets, we would expect the score for the top-chosen sense to greatly exceed those of the others. For puns, however, we would expect the two top-scoring dissimilar candidates to have similar scores, and the third dissimilar sense (if one exists) to score much lower. Given sufficient training data, it may also be possible to empirically determine the best score difference thresholds for discriminating puns from non-puns. (We hasten to note, however, that such an approach would not be able to distinguish between intentional and accidental puns. Whether this is a limitation or a feature would depend on the ultimate application of the pun detection system.)

Provided we are able to develop or further refine our pun detection and disambiguation methods such that they achieve sufficient precision and recall, the next steps would be to incorporate them in the higher-level NLP applications we covered in §5.1 and perform *in vivo* evaluations.

6.1 MOTIVATION

Despite the importance and popularity of word sense disambiguation as a subject of study, tools and resources supporting it have seen relatively little generalization and standardization. That is, most prior implementations of wsd systems have been hard-coded for particular algorithms, sense inventories, and data sets. This makes it difficult to compare systems or to adapt them to new scenarios without extensive reimplementation. The problem is by no means unique to wsd; Eckart de Castilho (2014) identifies the following barriers (among others) to the automatic analysis of language in general:

1. Existing automatic analysis components are not interoperable.
2. Implementing new automatic analysis components which *are* interoperable is too complex.
3. Assembling automatic analysis components into workflows is too complex.
4. The parameterization of analysis workflows is not readily supported.
5. Analysis workflows are not portable between computers.

In this chapter we present DKPro WSD, a modular and extensible processing framework for word sense disambiguation which aims to solve the above-noted problems. By *modular* we mean that it makes a logical separation between the *data sets* (e.g., the corpora to be annotated, the answer keys, manually annotated training examples, etc.), the *sense inventories* (i.e., the lexical-semantic resources enumerating the senses to which words in the corpora are assigned), and the *algorithms* (i.e., code which actually performs the sense assignments and prerequisite linguistic annotations), and provides a standard interface for each of these component types. Components can be assembled into a parameterizable workflow with relative ease, and components which provide the same functionality can be freely swapped. Thus one can easily run the same algorithm on different data sets (irrespective of which sense inventory they use), or test several different algorithms (or different parameterizations of the same algorithm) on the same data set.

While DKPro WSD ships with support for a number of common wsd algorithms, sense inventories, and data set formats, its *extensibility*

means that it is easy to adapt to work with new methods and resources. The system is written in Java and is based on UIMA (Ferrucci and Lally, 2004; Lally *et al.*, 2009), an industry-standard architecture for analysis of unstructured information. Support for new corpus formats, sense inventories, and wsd algorithms can be added by implementing new UIMA components for them, or more conveniently by writing UIMA wrappers around existing code. The framework and all existing components are released under the Apache License 2.0, a permissive free software licence.

DKPro WSD was designed primarily to support the needs of wsd researchers, who will appreciate the convenience and flexibility it affords in tuning and comparing algorithms and data sets. However, as a general-purpose toolkit it could also be used to implement a wsd module for a real-world natural language processing application. Its support for interactive visualization of the disambiguation process also makes it a powerful tool for learning or teaching the principles of wsd.

6.2 BACKGROUND AND RELATED WORK

6.2.1 Resources for wsd

In the early days of wsd research, electronic dictionaries and sense-annotated corpora tended to be small and hand-crafted on an ad-hoc basis. It was not until the growing availability of large-scale lexical resources and corpora in the 1990s that the need to establish a common platform for the evaluation of wsd systems was recognized. This led to the founding of the SENSEVAL (and later SemEval) series of competitions, the first of which was held in 1998 (see §2.4.4). Each competition defined a number of tasks with prescribed evaluation metrics, sense inventories, corpus file formats, and human-annotated test sets. For each task it was therefore possible to compare algorithms against each other.

However, sense inventories and file formats still vary across tasks and competitions. There are also a number of increasingly popular resources used outside SENSEVAL and SemEval, each with their own formats and structures: examples of sense-annotated corpora include SemCor (Miller, Chodorow, *et al.*, 1994), MASC (Passonneau, Baker, *et al.*, 2012), and webCAGE (Henrich, Hinrichs, and Vodolazova, 2012), and sense inventories include VerbNet (Kipper *et al.*, 2008), FrameNet (Ruppenhofer *et al.*, 2010), DANTE (Kilgarriff, 2010), BabelNet (Navigli and Ponzetto, 2013), and online community-produced resources such as Wiktionary and Wikipedia. So despite attempts at standardization, the canon of wsd resources remains quite fragmented.

6.2.2 Monolithic wsd systems

Most disambiguation systems described in the literature, including those entered into competition at the various SENSEVAL/SemEval campaigns, have been released in neither source nor binary form, thus precluding their reuse and modification by third parties. The few publically available implementations of individual disambiguation algorithms, such as SenseLearner (Mihalcea and Csomai, 2005), Sense-Relate::TargetWord (Patwardhan *et al.*, 2005), UKB (Agirre and Soroa, 2009), and IMS (Zhong and Ng, 2010), are all tied to a particular corpus and/or sense inventory, or define their own custom formats into which existing resources must be converted. Furthermore, where the algorithm depends on linguistic annotations such as part-of-speech tags, the users are expected to supply these themselves, or else must use the annotators built into the system. These annotators may not always be appropriate for a corpus language or domain other than the one the system was designed for.

Prior to the development of DKPro wsd, the only general-purpose dedicated wsd system we were aware of was I Can Sense It (Joshi *et al.*, 2012), a Web-based interface for running and evaluating various wsd algorithms. It includes i/o support for several corpus formats and implementations of a number of baseline and state-of-the-art disambiguation algorithms. However, as with previous single-algorithm systems, it is not possible to select the sense inventory, and the user is responsible for pre-annotating the input text with pos tags. The usability and extensibility of the system are greatly restricted by the fact that it is a proprietary, closed-source application fully hosted by the developers.

6.2.3 Processing frameworks

An alternative to constructing monolithic wsd systems is to build them using a *processing framework*—that is, a piece of software that invokes interoperable analysis components (Eckart de Castilho, 2014).

Natural language itself has several aspects—orthography, morphology, syntax, *etc.*—and as we mentioned in §2.2.2, specialized tools have been developed for processing each of these aspects. It is desirable in many text processing applications, including wsd, to run these tools in sequence, since some tools may benefit from the output of others. For example, a syntactic parser may require its input text to be first pos-tagged, a pos tagger may require its text to be first tokenized and segmented, and tokenizers and segmenters may require their input to be plain text rather than a computer markup language such as XML.

There already exists a wide variety of standalone NLP tools that could conceivably be used in the context of a wsd application. How-

ever, most of these tools do not directly interface with each other; the output from one is often in a format different from that expected as input by the next. Even where there exists a suite of interoperable tools, a given application may not wish to use them together. For example, one of the tools in the suite may have an incompatible alternative which is nonetheless more accurate or more specialized to the task at hand. In order to get incompatible tools to interoperate, it is necessary to write “glue code” to intermediate the transfer of data.

A processing framework eliminates the need to write glue code time and time again on an ad-hoc basis. It accomplishes this by adopting a standard data model through which tools communicate, and by wrapping the tools into modular *analysis components* whose activities are coordinated by the framework. In an NLP application, the framework invokes an instance of a tool with its desired parameterization, passes it the data in the format it expects, captures its output, and then shuts down the instance. The captured output is stored in the standard data model and, as and when necessary, converted into input for subsequent components in the toolchain, or formatted for final output to the user.

6.2.3.1 UIMA and DKPro

The Unstructured Information Management Architecture, or UIMA (Ferrucci and Lally, 2004), is an example of a processing framework which has become increasingly popular in the NLP community. UIMA, now a top-level project of the Apache Software Foundation, was originally developed at IBM and has had its specification published by the standards body OASIS (Lally *et al.*, 2009). The framework is not geared specifically towards language processing, but rather provides a general-purpose architecture for describing and running analysis components, and for storing and exchanging the data they use. Reference implementations exist in both Java and C++.

Applications built on UIMA can be thought of as factory assembly lines, where raw material in the form of unstructured data passes through a series of components that progressively structure it. The end-to-end process is modelled in a data structure known as an *aggregate analysis engine* (see Figure 9) which specifies a source-to-sink flow of data. At the beginning of the pipeline is a *collection reader* which iterates through a source collection and stores each document in a *common analysis structure* (CAS), a data structure for storing layers of data and stand-off metadata. The CAS is passed to one *analysis engine* (analysis component) after another. Each analysis engine derives a bit of structure from the data and records it in the CAS using user-defined data types known as *annotations*. At the end of the pipeline are *CAS consumers* which extract, analyze, display, or store annotations of interest.

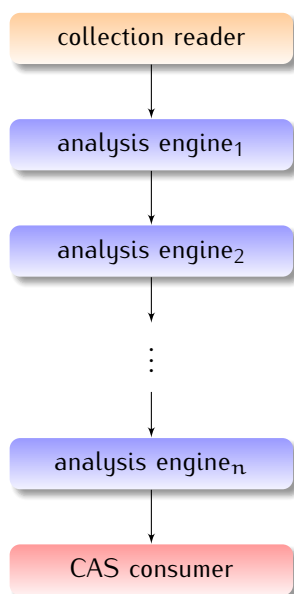


Figure 9. A UIMA aggregate analysis engine

A large number of NLP-specific tools have been wrapped or written from scratch as reusable UIMA collection readers, analysis engines, and CAS consumers. Furthermore, the UIMA framework itself has been extended to ease the assembly, configuration, and execution of language analysis workflows. Perhaps the largest collection of these components and extensions can be found in the Darmstadt Knowledge Processing Repository, or DKPro (Gurevych, Mühlhäuser, *et al.*, 2007). The DKPro family of projects includes DKPro Core (Eckart de Castilho and Gurevych, 2014), a collection of general-purpose UIMA-based NLP components; DKPro Similarity (Bär, Zesch, *et al.*, 2013), a UIMA-based framework for text similarity; and DKPro Lab (Eckart de Castilho and Gurevych, 2011), a UIMA framework for parameter sweeping experiments. DKPro also includes several tools not based on UIMA but still potentially useful for WSD, such as DKPro Statistics (Meyer, Mieskes, *et al.*, 2014) for computing correlation and interannotator agreement, and the unified lexical-semantic resources DKPro LSR (Garoufi *et al.*, 2008) and UBY (Gurevych, Eckle-Kohler, *et al.*, 2012).

6.3 SYSTEM DESCRIPTION

Though DKPro, not to mention other general-purpose NLP suites such as NLTK (Bird, 2006), provide frameworks and individual components potentially useful for WSD, they are not geared towards development and evaluation of WSD systems in particular. For instance, their type systems are not written with sense annotation in mind, they lack readers for some or all of the common sense-annotated data sets and corpus formats, and they do not provide ready-made components for

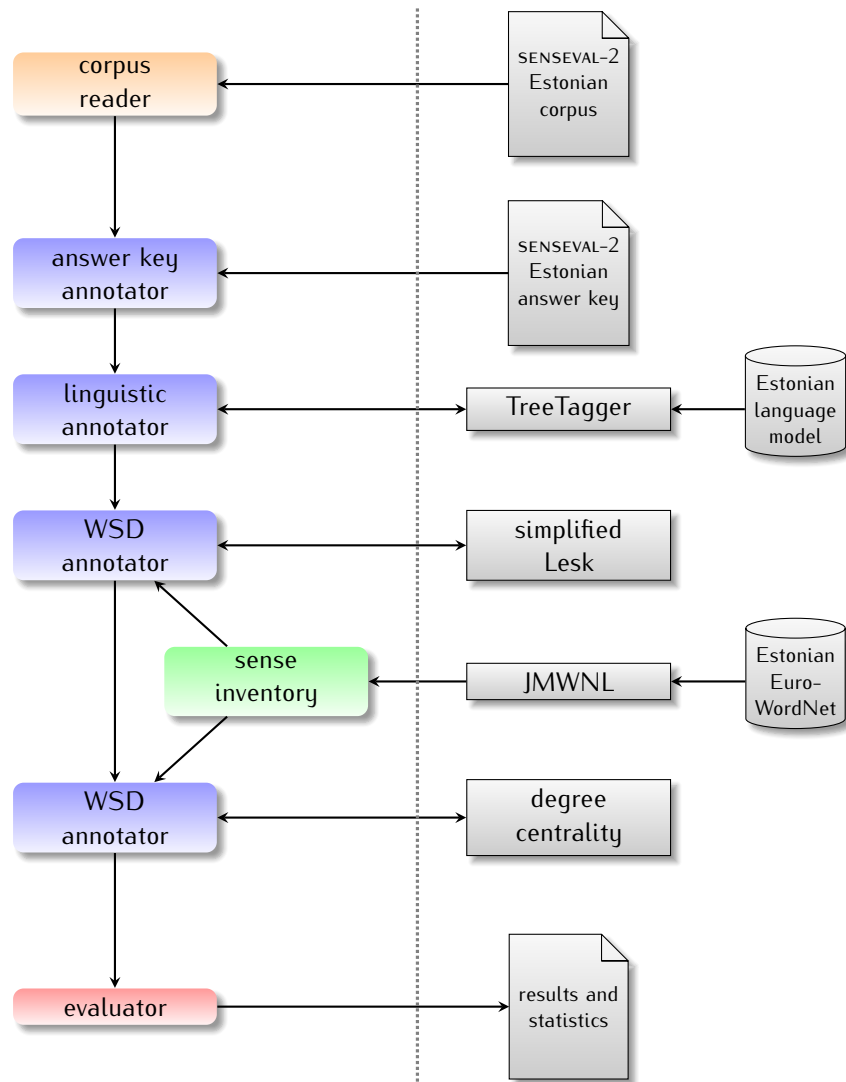


Figure 10. A sample DKPro WSD pipeline for the Estonian all-words data set from SENSEVAL-2

extending and evaluating disambiguation tools. This gap has led us to produce our own UIMA-based WSD-oriented processing framework, DKPro WSD .

Like other UIMA-based frameworks, DKPro WSD provides a collection of type systems, collection readers, annotators, CAS consumers, and other resources which the user combines into a data processing pipeline. We can best illustrate this with an example: Figure 10 shows a pipeline for running two disambiguation algorithms on the Estonian all-words task from SENSEVAL-2 (Kahusk *et al.*, 2001). UIMA components are the coloured, rounded boxes in the left half of the diagram, and the data and algorithms they encapsulate are the light grey shapes in the right half. The first component of the pipeline is a collection reader which reads the text of the XML-formatted corpus into a CAS and marks the words (instances) to be disambiguated with

their unique identifiers. The next component is an annotator which reads the answer key—a separate file which associates each instance ID with a sense ID from the Estonian EuroWordNet—and adds the gold-standard sense annotations to their respective instances in the CAS. Processing then passes to another annotator—in this case a UIMA wrapper for TreeTagger (Schmid, 1994)—which adds POS and lemma annotations to the instances. Then come the two disambiguation algorithms, also modelled as UIMA annotators wrapping non-UIMA-aware algorithms. Each WSD annotator iterates over the instances in the CAS and annotates them with sense IDs from EuroWordNet. (EuroWordNet itself is accessed via a UIMA resource which wraps the JMWNL library (Pazienza *et al.*, 2008) and which is bound to the two WSD annotators.) Finally, control passes to a CAS consumer which compares the WSD algorithms’ sense annotations against the gold-standard annotations produced by the answer key annotator, and outputs these sense annotations along with various evaluation metrics (precision, recall, *etc.*).

A pipeline of this sort can be written with just a few lines of code: one or two to declare each component and if necessary bind it to the appropriate resources, and a final one to string the components together into a pipeline. Moreover, once such a pipeline is written it is simple to substitute functionally equivalent components. For example, with only a few small changes the same pipeline could be used for SENSEVAL-3’s English lexical sample task (Mihalcea, Chklovski, and Kilgarriff, 2004), which uses a corpus and sense inventory in a different format and language. Specifically, we would substitute the collection reader with one capable of reading the SENSEVAL lexical sample format, we would pass an English instead of Estonian language model to TreeTagger, and we would substitute the sense inventory resource exposing the Estonian EuroWordNet with one for WordNet 1.7.1. Crucially, none of the WSD algorithms need to be changed.

The most important features of our system are as follows:

CORPORA AND DATA SETS. DKPro WSD currently has collection readers for most SENSEVAL and SemEval all-words and lexical sample tasks, the AIDA CONLL-YAGO data set (Hoffart *et al.*, 2011), the TAC KBP entity linking tasks (McNamee and Dang, 2009), and the aforementioned MASC, SemCor, and webCAGE corpora. Our prepackaged corpus analysis modules can compute statistics on monosemous terms, average polysemy, terms absent from the sense inventory, *etc.*

SENSE INVENTORIES. Sense inventories are abstracted into a system of types and interfaces according to the sort of lexical-semantic information they provide. There is currently support for WordNet 1.7 through 3.1 (Fellbaum, 1998), WN++-DC (Ponzetto and Navigli, 2010), EuroWordNet (Vossen, 1998), the Turk Bootstrap Word Sense Inven-

tory (Biemann, 2013), and UBY (Gurevych, Eckle-Kohler, *et al.*, 2012), which provides access to WordNet, Wikipedia, Wiktionary, Germanet, VerbNet, FrameNet, OmegaWiki, and various alignments between them. The system can automatically convert between various versions of WordNet using the UPC WN-Map mappings (Daudé *et al.*, 2003).

ALGORITHMS. As with sense inventories, WSD algorithms have a type and interface hierarchy according to what knowledge sources they require. Algorithms and baselines already implemented include the analytically calculated random sense baseline; the most frequent sense baseline; the original, simplified, extended, and lexically expanded Lesk variants (Miller, Biemann, *et al.*, 2012); various graph connectivity approaches from Navigli and Lapata (2010); Personalized PageRank (Agirre and Soroa, 2009); the supervised TWSI system (Biemann, 2013); and IMS (Zhong and Ng, 2010). Our open API permits users to program support for further knowledge-based and supervised algorithms.

LINGUISTIC ANNOTATORS. Many WSD algorithms require linguistic annotations from segmenters, lemmatizers, POS taggers, parsers, *etc.* Off-the-shelf UIMA components for producing such annotations, such as those provided by DKPro Core, can be used in a DKPro WSD pipeline with little or no adaptation.

VISUALIZATION TOOLS. We have enhanced some families of algorithms with animated, interactive visualizations of the disambiguation process. For example, Figure 11 shows part of a screenshot from the interactive running of the and 's (2010) degree centrality algorithm. The system is disambiguating the three content words in the sentence, "I drink milk with a straw." Red, green, and blue nodes represent senses (or more specifically, WordNet sense keys) of the words *drink*, *milk*, and *straw*, respectively; grey nodes are senses of other words discovered by traversing semantic relations (represented by arcs) in the sense inventory. The current traversal (toast%2:34:00:: to fuddle%2:34:00::) is drawn in a lighter colour. Mouseover tooltips provide more detailed information on senses. We have found such visualizations to be invaluable for understanding and debugging algorithms.

PARAMETER SWEEPING. The behaviour of many components (or entire pipelines) can be altered according to various parameters. For example, for the degree centrality algorithm one must specify the maximum search depth, the minimum vertex degree, and the context size. By interfacing with DKPro Lab, DKPro WSD can perform a parameter sweep, automatically running the pipeline once for every

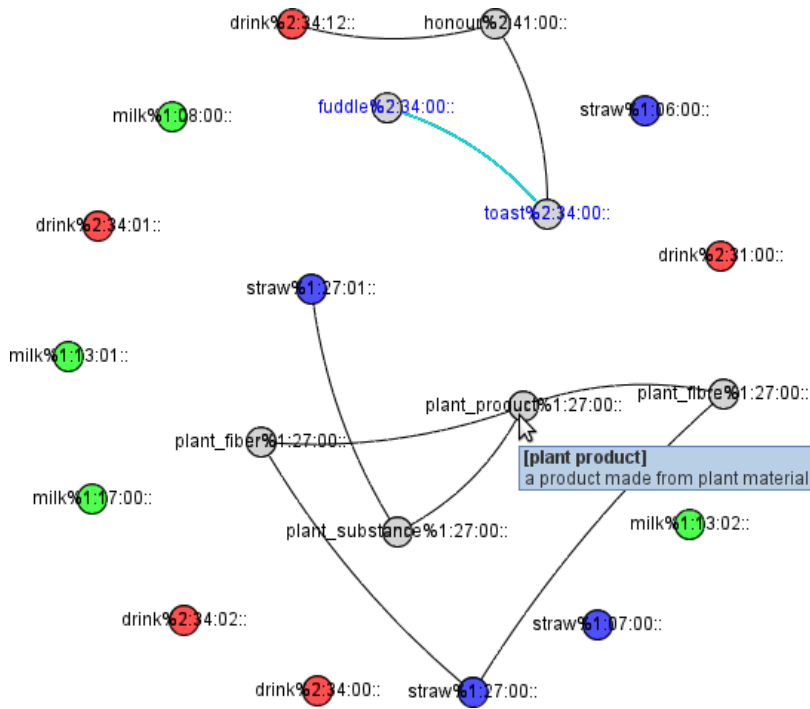


Figure 11. DKPro WSD's interactive visualization of a graph connectivity WSD algorithm

possible combination of parameters in user-specified ranges and concatenating the results into a table from which the optimal system configurations can be identified.

REPORTING TOOLS. There are several reporting tools to support evaluation and error analysis. Raw sense assignments can be output in a variety of formats (XML, HTML, CSV, SENSEVAL answer key, *etc.*), some of which support colour-coding to highlight correct and incorrect assignments. The system can also compute common evaluation metrics and plot precision–recall curves for each algorithm in the pipeline, as well as produce confusion matrices for algorithm pairs and calculate the statistical significance of the difference in accuracy. Users can specify backoff algorithms, or even entire chains thereof, and have the system compute results with and without the backoffs. Results can also be broken down by part of speech. Figure 12 shows an example of an HTML report produced by the system—on the left is the sense assignment table, in the upper right is a table of evaluation metrics, and in the lower right is a precision–recall graph.

DKPro WSD also has support for tasks closely related to word sense disambiguation:

ENTITY LINKING. Entity linking (EL) is the task of linking a named entity in a text (*e.g.*, *Washington*) to its correct representation in some knowledge base (*e.g.*, either *George Washington* or *Washington, DC* de-

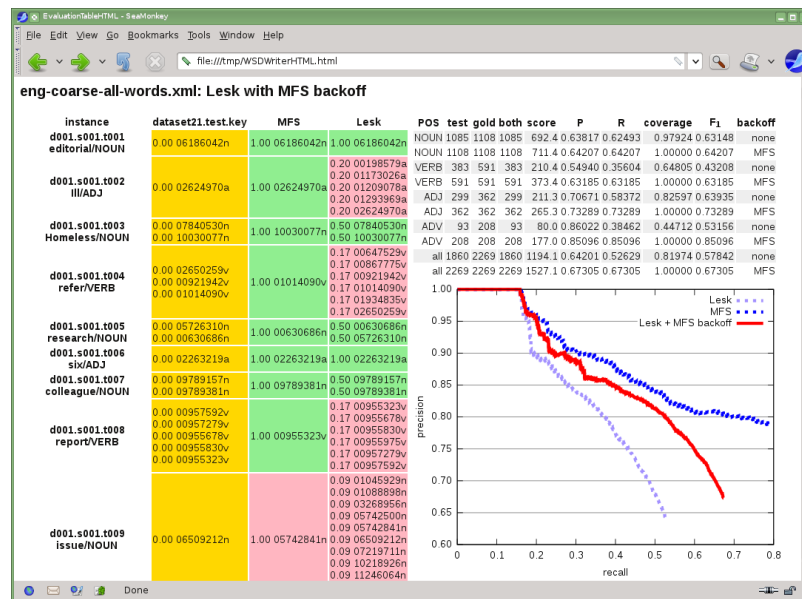


Figure 12. An HTML report produced by DKPro WSD

pending on the context). EL is very similar to WSD in that both tasks involve connecting ambiguous words in a text to entries in some inventory. DKPro WSD supports EL-specific sense inventories such as the list of Wikipedia articles used in the Knowledge Base Population workshop of the Text Analysis Conference (TAC KBP). This workshop, held annually since 2009, provides a means for comparing different EL systems in a controlled setting. DKPro WSD contains a reader for the TAC KBP data set, components for mapping other sense inventories to the TAC KBP inventory, and evaluation components for the official metrics. Researchers can therefore mitigate the entry barrier for their first participation at TAC KBP and experienced participants can extend their systems by making use of further wsd algorithms.

WORD SENSE INDUCTION. WSD is usually performed with respect to manually created sense inventories such as WordNet. In word sense induction (wsi) a sense inventory for target words is automatically constructed from an unlabelled corpus. This can be useful for search result clustering, or for general applications of wsd for languages and domains for which a sense inventory is not yet available. It is usually necessary to perform wsd at some point in the evaluation of wsi. DKPro WSD supports wsi by providing state-of-the art wsd algorithms capable of using arbitrary sense inventories, including induced ones. It also includes readers and writers for the SemEval-2007 and -2013 wsi data sets.

WORD SENSE CLUSTERING EVALUATION. In word sense clustering, the senses of an existing sense inventory are coarsened by automatically grouping (*clustering*) them according to their semantic similarity.

The resulting set of clusters is commonly evaluated by taking the raw sense assignments made by existing WSD systems and rescored them according to the coarsened sense inventory. To account for chance improvement, accuracy is measured relative to a random clustering of equivalent granularity. DKPro WSD contains modules for carrying out such evaluations of word sense clusterings, with random clustering scores computed as per the novel method we described in §4.5.1.

6.4 CONCLUSION

In this chapter we introduced DKPro WSD, a Java- and UIMA-based framework for word sense disambiguation. Its primary advantages over existing tools are its modularity, its extensibility, and its free licensing. By segregating and providing layers of abstraction for code, data sets, and sense inventories, DKPro WSD greatly simplifies the comparison of WSD algorithms in heterogeneous scenarios. Support for a variety of commonly used algorithms, data sets, and sense inventories has already been implemented.

Since its inception, the framework has seen use in a variety of research projects, including virtually all the experiments described in this thesis. Some further applications with no direct involvement by the present author are as follows:

- An entity linking system constructed with DKPro WSD is described by Erbs *et al.* (2012). It was evaluated competitively at the Knowledge Base Population track of the 2012 Text Analysis Conference at NIST (Mayfield *et al.*, 2012).
- Zorn and Gurevych (2013) used DKPro WSD to build a system for word sense induction and disambiguation. The system was entered in a SemEval shared task (Navigli and Vannella, 2013) where it achieved top performance according to one of the evaluation metrics.
- Del Corro *et al.* (2014) and Del Corro (2015) apply novel syntactic and semantic pruning techniques to boost accuracy in verb sense disambiguation. They tested their approaches on a variety of existing WSD implementations, including those provided by DKPro WSD.
- Wlotzka (2015) used the framework in a study of stream-based disambiguation methods.
- Heinzerling *et al.* (2015) used data types and interfaces from DKPro WSD to model entity mentions and links in an entity discovering and linking system. The system was evaluated at the Knowledge Base Population track of the 2015 Text Analysis Conference (Ji *et al.*, 2015).

DKPro WSD remains under active development, with work on several new features planned or in progress. These include implementations or wrappers for further algorithms and for the DANTE and BabelNet sense inventories. Source code, binaries, documentation, FAQs, an issue tracker, and community mailing lists are available on the project's website at <https://dkpro.github.io/dkpro-wsd/>.

7.1 MOTIVATION

7

Word sense disambiguation systems are commonly evaluated by having humans mark up the words in a text with their contextually appropriate meanings, as enumerated by a dictionary or other lexical-semantic resource, and then comparing these annotations against the ones supplied by the systems. Such “*in vitro*” evaluations are popular because they are straightforward to conduct, though they have the disadvantages of requiring considerable effort to produce the manually annotated gold standard data, and of requiring all human and machine annotators to use the same sense inventory.

A more recent and increasingly popular evaluation method is the *lexical substitution* task. Here the lexical annotations which are applied and compared are not sense labels, but rather lists of plausible synonyms. Because the human annotators’ substitutes are provided freely rather than selected from a fixed list, lexical substitution works around the second of the two problems mentioned above. It has been argued that, since the identification and ranking of the substitutions depends on a proper understanding of the word’s meaning in context, accuracy in this “*in vivo*” task is an indirect measure of WSD performance (McCarthy, 2002).

Partly because of the expense in producing it, however, sense- and substitution-annotated data remains scarce. Since the advent of organized evaluation competitions at SENSEVAL, the research community has published only about nine monolingual English evaluation sets for “pure” WSD¹ and only three data sets for monolingual English lexical substitution (McCarthy and Navigli, 2009; Biemann, 2013; Kremer *et al.*, 2014). The situation for other languages is even grimmer; German, for example, boasts just four sense-annotated corpora and only a single lexical substitution data set.

The principal contribution of ours discussed in this chapter is GLASS (German Lexemes Annotated with Senses and Substitutions), a new German-language sense-annotated data set. The GLASS data set aims to fill a gap in German-language resources by providing a lexical sample corpus that (i) features high-quality, manually applied sense annotations, (ii) is well balanced with respect to the target words’ frequency and part of speech, (iii) is of sufficient size to be useful for

¹ See Table 1 on p. 33. To these we might add a handful of obsolete data sets that predate SENSEVAL (Agirre and Edmonds, 2007, Appendix A.2.2), plus a few larger sense-tagged corpora that nevertheless were not specifically designed as evaluation sets (*e. g.*, Miller, Leacock, *et al.*, 1993; Mihalcea and Chklovski, 2003; Hovy *et al.*, 2006; Passonneau, Baker, *et al.*, 2012).

machine learning, and (iv) is distributed under a free content licence. Because GLASS extends an existing lexical substitution data set, it allows for *in vitro* and *in vivo* evaluations of WSD systems to be carried out on the same data. Moreover, GLASS is the first resource in any language to permit an empirical study of the relationship between manually annotated word senses and lexical substitutes.

7.2 BACKGROUND AND RELATED WORK

7.2.1 Sense-annotated data for German

There exist a handful of previously published sense-annotated data sets in the German language, all of which are of the lexical sample variety. The properties of these data sets, as well as those of our own GLASS, are summarized in Table 17.

The earliest of these resources, the MuchMore evaluation set (Rai-leanu *et al.*, 2002), has GermaNet annotations for 2421 occurrences of 25 nouns in a corpus of medical abstracts. Raw interannotator agreement was high ($A_o = 0.841$). Though the authors have made the data available for download, there is no explicit statement of the terms of use, so it cannot be presumed to be freely licensed.

The dewSD resource (Broscheit *et al.*, 2010) has manual sense annotations for 1154 occurrences of 40 lemmas (6 adjectives, 18 nouns, 16 verbs) in the deWaC corpus (Baroni *et al.*, 2009). While the lemmas were translated from an English WSD data set, some attempt was made to yield a good distribution across parts of speech, polysemy, and word frequency. No information is provided on the manual annotation process, including interannotator agreement. Though the original dewSD data set was annotated with senses from GermaNet 5.1, Henrich (2015) later updated these to GermaNet 9.0. As with the MuchMore data set, dewSD is available for download but with no specified licence.

webCAGE (Henrich, Hinrichs, and Vodolazova, 2012; Henrich, 2015) is a set of 10 750 occurrences of 2607 lemmas which have been semi-automatically tagged with senses from GermaNet 7.0 through 9.0. The source contexts are all Web-harvested, and include a mix of free and proprietary content. The portion of the data set derived from free sources (9376 tagged word tokens) is distributed under the terms of the Creative Commons Attribution-ShareAlike licence; as the full data set includes proprietary content, it is not publically available. A parallel project, wikiCAGE (Henrich, Hinrichs, and Suttner, 2012), applied GermaNet 6.0 sense annotations semi-automatically to a Wikipedia corpus containing 24 334 occurrences of 1030 lemmas. While it was

	WIKICAGE	WEBCAGE	MUCHMORE	DEWSD	TÜBA-D/Z	CLASS
TYPES { ADJECTIVES NOUNS VERBS TOTAL	0	211	0	6	0	51
	1 030	1 499	25	18	30	51
	0	897	0	16	79	51
	1 030	2 607	25	40	109	153
TOKENS	24 344	10 750	2 421	1 154	17 910	2 038
GERMANET VERSION	6.0	7.0–9.0	1.0(?)	5.1, 9.0	8.0	9.0
DOMAIN	open	open	medical	open	open	open
ANNOTATION METHOD	semi-automatic	semi-automatic	manual	manual	manual	manual
LICENCE	unpublished	CC BY-SA / proprietary	proprietary	proprietary	proprietary	CC BY-SA

Table 17. Comparison of sense-tagged corpora for German

intended to be released under a free content licence, it was never published.²

Recent versions of the TüBa-D/z treebank (Henrich and Hinrichs, 2013, 2014; Henrich, 2015) include manually applied GermaNet 8.0 annotations for 17 910 occurrences of 109 lemmas (30 nouns and 79 verbs). Lemmas were selected to ensure a good balance of word frequencies, number of distinct senses, and (for verbs) valence frames. Interannotator agreement was generally high (mean Dice coefficient of 0.964 for nouns and 0.937 for verbs). While the data is available for non-profit academic use, it is not released under a free content licence.

7.2.2 Lexical substitution

7.2.2.1 Task description

Lexical substitution is the task of identifying appropriate substitutes for a target word in a given context. For example, consider the following two German-language contexts (abridged from the Cholakov *et al.* (2014) data) containing the word *Erleichterung*:

- (22) In der Legislaturperiode 1998–2002 wurden einige Reformen des Staatsbürgerschaftsrechts bezüglich der *Erleichterung* von Einwanderung verabschiedet.
[In the legislative period of 1998–2002 a few reforms on citizenship law concerning the *easing* of immigration were passed.]
- (23) Vor allem auf dem Lande war die Umstellung aber schwer durchsetzbar und die *Erleichterung* groß, als 1802 der Sonntagsrhythmus und 1805 der vorrevolutionäre Kalender insgesamt wieder eingeführt wurden.
[The change was particularly difficult to enforce in the countryside, and there was great *relief* when in 1802 the Sunday routine and in 1805 the pre-revolutionary calendar were reintroduced.]

The word *Förderung* (meaning “facilitation”) would be an appropriate substitute for *Erleichterung* (meaning “easing”) in the first context, whereas the word *Freude* (meaning “delight”) would not be. Conversely, *Freude* would indeed be a valid substitute for *Erleichterung* (meaning “relief”) in the second context, whereas *Förderung* would not be.

Lexical substitution is a relatively easy task for humans, but potentially very challenging for machines because it relies—explicitly or implicitly—on word sense disambiguation. In fact, lexical substitution was originally conceived as a method for evaluating word sense

² Personal communication with V. Henrich, 7 September 2015.

		SEMEVAL	EVALITA	SEMDIS	GERMEVAL
TYPES	ADJECTIVES	≈ 58	58	10	51
	ADVERBS	≈ 35	36	—	—
	NOUNS	≈ 62	75	10	51
	VERBS	≈ 54	63	10	51
	TOTAL	197	232	30	153
TOKENS	ADJECTIVES	≈ 1 091	1 160	100	510
	ADVERBS	≈ 940	359	—	—
	NOUNS	≈ 901	728	100	510
	VERBS	≈ 587	620	100	1 020
	TOTAL	2 010	2 283	300	2 040
A_o		0.278	?	0.258	0.166*
A_{Mo}		0.507	?	0.73	?
LANGUAGE		en	it	fr	de
TRAINING:TEST		1:5.7	1:7.4	0:1	2:1

* On the subset using trained annotators

Table 18. Lexical substitution data sets from evaluation campaigns

disambiguation systems which is independent of any one sense inventory. However, it also has a number of uses in real-world NLP tasks, such as text summarization, question answering, paraphrase acquisition, text categorization, information extraction, text simplification, lexical acquisition, and text watermarking.

7.2.2.2 Evaluation and data sets

As with WSD, evaluation of automated lexical substitution systems is effected by applying them on a large number of word–context combinations and then comparing the substitutions they propose to those made by human annotators. The lack of a reference inventory of substitutes makes it impossible to directly apply traditional WSD metrics such as precision and recall; however, there are various nontrivial ways in which these metrics have been adapted (McCarthy and Navigli, 2009).

To date there have been four organized evaluation campaigns for lexical substitution: an English-language task at SemEval-2007 (McCarthy and Navigli, 2007, 2009), an Italian task at EVALITA 2009 (Toral, 2009), a French task at SemDis 2014 (Fabre *et al.*, 2014), and our own German task at GermEval 2015 (Miller, Benikova, *et al.*, 2015). Their respective data sets are summarized in Table 18.

The SemEval lexical substitution data set consists of 2010 sentences in which five trained annotators have provided substitutes for one

of the words. The lemmas were selected, some manually and some at random, such that they had more than one meaning and at least one synonym, as indicated by a variety of lexical resources. The sentences were selected, also partly manually and partly at random, from 's (2006) English Internet Corpus. The targets' part of speech was not explicitly marked in the raw data, nor were the annotators required to provide substitutions matching a particular part of speech, so for some of the 197 target lemmas, substitutes of different parts of speech were given. Interannotator agreement was measured by mean percentage agreement, for all instances ($A_o = 0.278$) and for all those instances which had a "mode", or single most frequently provided substitute ($A_{Mo} = 0.739$).

For the Italian task, 2283 instances of 232 lemmas were annotated by three annotators.³ The lemmas were carefully selected from various LSRS to guarantee a high level of polysemy, though some effort was made to ensure that the set was otherwise representative of the Italian language. The contexts consist of roughly ten sentences per lemma manually selected from a corpus of Italian newspapers and periodicals, plus another ten randomly selected from the same sources.

The data set for the French-language task at SemDis 2014 is comparatively small, with just 300 instances of 30 lemmas. The lemmas were manually selected to ensure an even distribution across part of speech (ten nouns, verbs, and adjectives), and also on the basis of their word frequency, polysemy, and "substitutability". Word frequency had a lower limit of 500 occurrences in the frWaC corpus (Baroni *et al.*, 2009); the later two criteria were based on polysemy and synonymy as indicated by a dictionary. Ten instances from each lemma were manually selected from frWaC, and up to three substitutes were provided by seven human annotators. Interannotator agreement was reported as $A_o = 0.258$, $A_{Mo} = 0.73$.

The German-language data set (Cholakov *et al.*, 2014) is similar in size to the English and Italian sets, with 2040 instances of 103 target words. The target words were randomly selected across an equal distribution of part of speech (nouns, verbs, and adjectives) and three frequency bands as measured by occurrence counts in the deWaC corpus (Baroni *et al.*, 2009); the data set's creators did not control for polysemy or synonymy as they did not wish to introduce a bias towards any one sense inventory. To similarly avoid biasing the selection of word senses, sentences containing the target lemmas were randomly selected from a source corpus. That corpus, the German edition of freely licensed Wikipedia, was chosen to ensure that the data set can also be freely distributed and modified. (All three of the

³ The data set described in Toral (2009) differs from the one distributed on the EVALITA 2009 workshop website. The instance and lemma counts reported here are from our own examination of the published data.

previously mentioned data sets are derived from proprietary sources, which limits their usability.)

To validate the annotation guidelines, part of the data set (200 instances of fifteen words) was manually annotated by four trained annotators; IAA was calculated as $A_o = 0.1695$. The remainder of the data was annotated by one trained annotator and five annotators per sentence recruited via crowdsourcing. No further IAA calculations were made on this portion of the data set, though it was post-processed by two professional annotators to normalize the annotations for spelling and POS and to remove contributions that were obviously copied and pasted from thesauri.

All four data sets described above are provided in the same format as XML and delimited text files. The XML files contain single-sentence instances enclosed in `instance` and `context` elements. Within each instance, the target word is enclosed in a `head` element. Instances with the same target lemma are grouped together in a `lexelt` element. The `lexelt` elements are grouped together in a top-level corpus element.

The gold-standard substitutions provided by the human judges are provided as stand-off annotations in the form of delimited text files. Each line has the format

```
lexelt id :: subs
```

where

lexelt is the unique identifier for the target lemma, corresponding to the `item` attribute of the `lexelt` element in the XML file;

id is the unique identifier for the instance, which matches the `id` attribute of the `instance` element; and

subs is a semicolon-delimited list of lemmatized substitutes. Following each substitute is its corresponding frequency count (indicating the number of annotators who provided that substitute).

The formats of the two file types are illustrated in Figures 13 and 14.

In addition to the above-noted data sets, there are two further English-language ones which have not been used in any organized evaluation campaigns: `TWSI` (Biemann, 2013) and `coInCo` (Kremer *et al.*, 2014). The former is a collection of 1012 target nouns in 183 398 sentential contexts, to which crowd-sourced annotators have applied lexical substitutions. Occurrences of the same target are then automatically clustered by substitution overlap to induce a sense inventory. Of greater relevance to the present thesis is `coInCo`, an all-words corpus to which crowd-sourced annotators applied substitutions for 15 629 instances of 3874 lemmas in 2474 sentences. Each instance's set of substitutes is then automatically mapped to a synset in WordNet 3.1.

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE corpus SYSTEM 'lexsub.dtd'>
<corpus lang="de">
  <lexelt item="Monarch.n">
    <instance id="Monarch_1">
      <context>
        Dies war die letzte britische Regierung, die ein
        <head>Monarch</head> ohne Mehrheit im Unterhaus
        ernannte, und scheiterte schon im April 1835.
      </context>
    </instance>
    :
  </lexelt>
  :
</corpus>

```

Figure 13. Format of the Cholakov *et al.* (2014) data set's XML files

```

Monarch.n Monarch_1 :: König 3; Herrscher 2; Adliger 1;
Staatsoberhaupt 1;

```

Figure 14. Sample line from the Cholakov *et al.* (2014) data set's stand-off annotation files

7.3 DATA SET CONSTRUCTION

7.3.1 Resource and data selection

Our decision to sense-annotate the existing lexical substitution data set by Cholakov *et al.* (2014) was motivated by three considerations. First, since it is based on text from Wikipedia, the data is modifiable and redistributable under a copyleft licence. There is therefore no legal barrier to our extending and republishing the data set, nor will there be such a barrier to others in the research or commercial communities who wish to do likewise with our own version. Second, having both sense and lexical substitution annotations conveniently allows for intrinsic and extrinsic evaluations of WSD systems to be carried out on the same data. Finally, the double annotations provide a rich resource for investigating the relationship between word senses and lexical substitutions. (The only previous study on this topic (Kremer *et al.*, 2014) uses automatically induced rather than manually applied word sense annotations.)

Our review of the original lexical substitution data set revealed that two of the lemmas had duplicate context sentences. We removed these, lowering the total number of contexts in the data set to 2038. We also corrected two types of inconsistencies in the segmentation of words in the XML files. First, in some cases inflections were included in the head elements, and in others they were not. Second, the posi-

tioning of the head element was inconsistent when the target lemma appeared as part of a compound word; sometimes the entire compound was marked and sometimes just the element of interest. Our normalized segmentation lumps inflections but splits compounds.

In line with the sense-annotated corpora discussed in §7.2.1, we chose GermaNet as our sense inventory. GermaNet (Hamp and Feldweg, 1997; Henrich and Hinrichs, 2010) is a lexical-semantic network that relates German-language nouns, verbs, and adjectives. Like its English analogue, the WordNet, GermaNet represents semantic concepts as synsets which are interlinked through labelled semantic relations. We used version 9.0 of the resource, which contains 93 246 synsets covering 121 810 lexical units.

As no published sense annotation tool supports using GermaNet as the sense inventory, we developed our own browser-based annotation tool, Ubyline (Miller, Khemakhem, *et al.*, 2016), which is not tied to any one sense inventory and whose interface is optimized for use with lexical sample data. Ubyline retrieves its senses from UBY (Gurevych, Eckle-Kohler, *et al.*, 2012), a unified lexical-semantic resource containing not only GermaNet but eleven other multilingual resources. It can therefore be readily adapted for lexical annotation tasks in various languages with different sense inventories.

7.3.2 Annotation process

We trained and engaged three human judges—all native German-speaking graduate students in computational linguistics—to produce our manually annotated data set. Two judges independently sense-annotated all 2038 instances in our data set, and the third served as an adjudicator who went through all occurrences where the two annotation sets differed and resolved the disagreements.

The two annotators were trained in the use of Ubyline and given oral and written annotation guidelines. We configured Ubyline such that, in addition to the senses from GermaNet, annotators had the option of applying two special senses: “proper name” (P) for senses not in GermaNet because they refer to a proper name, and “unassignable” (U) for senses not in GermaNet for any other reason. The annotators were free to consult outside sources, such as dictionaries, to help them understand the contexts, but they were not permitted to discuss cases with each other.

After annotating twenty lemmas each, the guidelines were revised to account for some anomalies in the data and in GermaNet itself. For instance, the annotators had discovered that for at least one lemma, *Korrektur*, the definition given by GermaNet was more specific than the hyponyms it lists. On further investigation this appeared to be an error introduced by a recent project to supplement GermaNet’s definitions with those semi-automatically extracted from Wiktionary (Hen-

rich, Hinrichs, and Vodolazova, 2014). We therefore instructed the annotators to resolve any apparent conflicts between GermaNet’s sense definitions and hypernym–hyponym taxonomy in favour of the latter.

For the adjudication phase, the adjudicator was provided with the original annotation guidelines as well as a set of adjudication guidelines. The latter basically instructed the adjudicator, for each instance on which the annotators disagreed, to accept one or the other set of annotations, or the union of the two. A custom browser-based interface was provided to effect the adjudications; this presented similar information as Ubyline, plus which senses (or sets thereof) were selected by the two annotators.

The final versions of the annotation and adjudication guidelines appear in Appendices C and D, respectively.

7.4 ANALYSIS

7.4.1 Interannotator agreement

Following Raileanu *et al.* (2002) and Henrich and Hinrichs (2013), we calculate interannotator agreement (IAA) using both raw percentage agreement and the Dice coefficient. Our mean raw agreement is high (0.861, 0.865, and 0.815 for adjectives, nouns, and verbs, respectively), and seemingly better than that reported for MuchMore (0.841 for nouns). The Dice coefficient, which awards credit for partial matches, gives us IAA scores of 0.873, 0.896, and 0.835 for adjectives, nouns, and verbs, respectively. These results are somewhat lower than those reported for TüBa-D/Z (0.964 for nouns and 0.937 for verbs), though it should be noted that unlike our annotators, theirs did not have the option of marking target words as unassignable or proper names. Furthermore, because these measures of IAA do not account for the sense distributions within and across data sets, they may not meaningfully reflect the relative reliabilities of the data sets.

Both Raileanu *et al.* (2002) and Henrich (2015) make further computations of IAA per lemma using Cohen’s κ , a chance-correcting measure of IAA (see §2.4.1.1). However, this metric is unable to cope with instances that receive multiple sense annotations (as happened in about 4.1% of our cases, as well as 3.3% and 0.4% of the MuchMore and TüBa-D/Z instances, respectively). Furthermore, neither Cohen’s κ , nor other IAA measures which do work with multiple labels (such as Krippendorff’s α), return meaningful results when all annotators apply the same sense annotation to all occurrences of a given lemma. This situation arises relatively often for our lemmas, which have lower average polysemy and much lower occurrence counts than those of TüBa-D/Z and MuchMore.

Raileanu *et al.* (2002) and Henrich and Hinrichs (2013) skirt both problems by simply excluding the affected instances from their κ calculations. With this expediency, the MuchMore lemmas yield κ scores ranging from 0.33 to 1.00, and the TüBa-D/Z ones from -0.00 to 1.00. When we do likewise, we observe a much wider range of κ scores, from -0.43 to 1.00. Since there were no obvious patterns of disagreement in the early phase of the annotation process, we suspect that this is a result of differences in our data set rather than an indicator of low quality. That is, the lemmas with systematic disagreement are indeed an artifact of their low polysemy and lower applicable occurrence counts. As further evidence of this, we observe a moderate negative correlation between lemma polysemy and (Dice) agreement for adjectives and nouns, with 's (1896) $r = -0.302$ and -0.333 , respectively. There is, however, no appreciable correlation for verbs ($r = -0.076$). A complete survey of interannotator agreement for this data set's lemmas, as well as the polysemy per lemma, can be found in Appendix E.

In the adjudication phase, a slight preference was expressed for annotations made by the first of the two annotators. Of the 328 items in disagreement, 200 (61%) were resolved in favour of the first annotator and 107 (33%) in favour of the second annotator. For the remaining 21 instances (6%), the adjudicator adopted the union of the two annotation sets.

Following adjudication, we are left with a data set in which 2079 sense annotations have been applied to 2038 instances, for an average of 1.02 senses per instance. This finding is in line with that of Henrich and Hinrichs (2014), who observe that the need to annotate more than one sense occurs infrequently. The special *r/u* senses were applied to 203 instances.

7.4.2 Characterizing lexical substitutions

As mentioned in §7.2.2, the constructors of the GermEval data set had made a conscious decision not to control for polysemy in order to avoid biasing their selection of lemmas to any one sense inventory. Perhaps as a result, GLASS does not exhibit as wide a range of sense coverage as other sense-annotated data sets. Table 19 shows the frequency of the lemmas in GLASS by part of speech and polysemy in GermaNet 9.0. About half the verbs, two thirds of the nouns, and nearly all the adjectives have only a single sense listed in GermaNet. However, the average number of senses per lemma, 1.40, is still higher than GermaNet's overall average of 1.31.

We next undertake an investigation to determine the sort of lexical-semantic relations that hold between a disambiguated target and its substitutes. A similar study had been conducted by Kremer *et al.* (2014), though the sense annotations in their data set were automat-

POS	POLYSEMY				TOTAL
	1	2	3	4	
adjectives	48	3	0	0	51
nouns	33	11	6	1	51
verbs	28	17	5	1	51
total	109	31	11	2	153

Table 19. Number of lemmas in GLASS by part of speech and polysemy in GermaNet

ically induced. Ours is therefore the first such study using manually applied sense annotations; it is also the first study using German-language data.

7.4.2.1 Substitute coverage

We first consider GermaNet’s coverage of the data set’s 4224 unique *substitute types*—that is, the union of all words and phrases suggested by *et al.*’s (2014) annotators, with duplicates removed. Of these types, only 3010 (71%) are found in GermaNet. Among the 1214 substitute types missing from GermaNet are many phrases or multiword expressions (38%), nominalizations of verbs which do occur in GermaNet (about 3%), and other derivations and compounds. There does not appear to be a great difference in lexical coverage for substitute types applied to items with successful versus unsuccessful sense annotations: 1081 of the 3887 unique substitute types applied to successfully sense-annotated items were not found in GermaNet (28%), as compared to 163 of the 667 types applied to the P/U items (24%).

7.4.2.2 Relating targets and substitutes

We next consider the semantic relations that link the successfully annotated target senses to their lexical substitutes. Recall that in GermaNet, words are grouped into structures known as synsets, where all words in the synset are synonymous. Synsets are in turn represented as vertices in a graph structure, with named semantic relations as the connecting edges. Table 20 shows the percentage of *substitute tokens* (*i. e.*, the individual words or phrases proposed as substitutes for each target, disregarding their frequency among annotators) which are synonyms, direct hypernyms, transitive hypernyms, direct hyponyms, or transitive hyponyms of any of its target’s annotated senses. (The figures for transitive hypernyms and hyponyms exclude the direct hypernyms and hyponyms—that is, the target synset and the synset containing the substitute are endpoints on a path of length 2 or greater.) The table also shows the percentage of substitutes di-

RELATION	ADJ.	NOUNS	VERBS	TOTAL
Synonym	7.5	6.6	4.6	5.9
Direct hypernym	7.1	7.5	6.5	6.9
Transitive hypernym	0.2	3.3	1.5	1.6
Direct hyponym	3.0	4.9	3.1	3.5
Transitive hyponym	1.5	0.7	0.8	0.6
Other direct relation	0.0	0.0	0.0	0.0
Otherwise reachable	60.4	58.9	71.2	65.4
Not in GermaNet	21.5	18.6	12.3	16.3

Table 20. Percentage of substitutes in successfully sense-annotated items in GLASS by their connection to the sense(s) through various semantic relations in GermaNet

rectly reachable by following any other type of semantic relation, the percentage of substitutes which exist in GermaNet but are not reachable from the target sense(s) via a path of uniform semantic relations, and the percentage of substitutes not covered by GermaNet at all.⁴

From these statistics we can make a number of observations and comparisons to the English-language study of Kremer *et al.* (2014). First, the proportion of substitute tokens not in GermaNet is slightly lower than the proportion of substitute types not in GermaNet (16% *vs.* 24%). That is, of all substitute types in the data set, the annotators were more likely to apply those in GermaNet. Nonetheless, GermaNet’s coverage of the substitutes in GLASS (84%) is significantly lower than WordNet’s coverage of the substitutes in coInCo (98%). Some of this difference must be due to how strictly each study’s annotation guidelines discouraged the use of phrases and multiword expressions, which are largely absent from both WordNet and GermaNet. Around 6% of the GLASS substitutes are not in GermaNet because they are phrases or multiword expressions; the same figure for coInCo cannot be more than 2%. The rest of the difference in substitute coverage may simply be a consequence of the size of the respective LSRs; WordNet 3.1 has about one and a third times the number of lemmas as GermaNet 9.0.

A second observation we can make is that the proportions of substitutes found in the synsets of the annotated senses and of those found in the synsets of the direct hypernyms are generally similar, while the proportion found in the synsets of transitive hypernyms is much lower. This is expected in light of the annotation instructions reported

⁴ The numbers in each column of Table 20 may sum to slightly more than 100%, since a few words appear multiple times in the same hypernymy-hyponymy taxonomy. For example, in GermaNet the word *Öl* is its own hypernym, because it is a synonym of synset 40402 (petroleum oil) and also of the hypernym synset 48480 (a viscous liquid not miscible with water). This also holds for the English word *oil* in WordNet.

in Cholakov *et al.* (2014), which encouraged annotators to choose “a slightly more general word” only if there was no one word or phrase which perfectly fit the target. What is particularly surprising, however, is the sizeable proportion of substitutes found in the synsets of direct and transitive hyponyms. This is because the annotation instructions did not make any provision for using more specific terms as substitutes. This anomaly was also observed in *coInCo*, where direct and transitive hyponyms account for 7.5 and 3.0% of the substitutes, respectively.

Our third observation is that in no case is a substitute found in a synset directly related to the target by any semantic relation other than hypernymy or hyponymy. (GermaNet provides twelve such relation types, which are all (sub)classes of meronymy/holonymy, entailment, causation, and association.) This finding is also surprising, since it is not uncommon for meronyms or holonyms to serve as substitutes in German (Schemann, 2011, pp. 39*–43* [*sic*]). For example, as in English, the word *Person* (“person”) can be substituted with its meronym *Kopf* (“head”) in many contexts:

- (24) Wir haben 8€ pro Person verdient.
[We earned €8 per person.]
- (25) Wir haben 8€ pro Kopf verdient.
[We earned €8 per head.]

It is unclear whether or not semantic relations besides hypernymy and hyponymy produced any valid substitutes in *coInCo*; Kremer *et al.* (2014) do not include them in their analysis.

Finally, we note that the majority of substitutes cannot be reached by following semantic relations of a single type. That is, some 60% of all substitutes exist as synonyms somewhere in the GermaNet graph, but are reachable from the target synset only by following semantic relations of at least two different types. This observation was also made for *coInCo*, where 69% of the substitutes exist outside the target’s hypernym/hyponym taxonomy..

7.4.2.3 Comparing paraset to synsets

Kremer *et al.* (2014) introduce the term *paraset* to refer to the set of substitutions produced for each target in its context, and investigate to what extent their paraset follows the boundaries of WordNet synsets. As their data set does not include manual sense annotations, they sense-annotate their targets heuristically by selecting the synset that has the greatest number of synonyms in common with the paraset. To overcome the lexical gap problem, they extend each synset’s synonyms with those of its immediate hypernyms and hyponyms.

To measure the extent to which the paraset contains substitutes from a single synset, one can compute the *cluster purity* (Manning, Raghavan, *et al.*, 2008, §16.3). This metric, borrowed from information

MEASURE	ADJ.	NOUNS	VERBS	TOTAL
cluster purity	0.774	0.795	0.824	0.801
mean paraset size	7.008	5.445	6.532	6.377
mean common core size	0.928	0.954	0.862	0.903
% common cores non-empty	62.963	72.727	42.254	58.639
% substitutes in common core	14.268	22.540	14.662	16.667

Table 21. Paraset purity and common core statistics for GLASS, by part of speech

retrieval, measures the accuracy of each cluster with respect to its best matching gold class:

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|, \quad (7.1)$$

where $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ is the set of clusters, $\mathbb{C} = \{c_1, c_2, \dots, c_J\}$ is the set of classes, and N is the number of objects being clustered. Purity values range from 0 to 1, where 0 is the absence of purity and 1 is total purity. For our purposes, Ω is the set of parasets in GLASS, \mathbb{C} is the set of synsets in GermaNet, and N is the total number of substitute tokens. Like Kremer *et al.* (2014), we consider only those substitutes that are found in the target’s synsets or those of its hypernyms and hyponyms, as it would otherwise be unclear whether low purity implies substitutes from a mixture of senses (which is what we are trying to measure) or simply a large number of substitutes reachable via relations other than hypernymy and hyponymy (which we already confirmed above). By necessity we also disregard those instances which our annotators tagged as P/U or with more than one sense.

Our overall purity is 0.801; the first line of Table 21 shows purity values broken down by part of speech. These results are comparable to those of Kremer *et al.* (2014), who report purities of 0.812 for nouns and 0.751 for verbs. This is good evidence that our substitutes—or at least, the ones which are synonyms or direct hyper-/hyponyms of the target—tend to follow the boundaries of single GermaNet synsets.

7.4.2.4 Similarity between same-sense parasets

In the previous section, we analyzed those substitutes found in the immediate semantic neighbourhood of the target sense. However, because the majority of our substitutes are found outside this neighbourhood, we now perform an investigation which includes these more distant relatives. In particular, we are interested in determining the similarity of parasets representing the same word sense.

Paraset similarity can be quantified as the number and proportion of their substitutes in the *common core*—that is, the intersection of all

paraset for targets tagged with the same sense. As Table 21 shows, the paraset in our data set (again, excluding those for P/U and multiply tagged instances) have about 6.4 substitutes on average, with adjectives being slightly more substitutable and nouns slightly less. Most of these paraset—about 59%—have a non-empty common core. The average common core size across all parts of speech is slightly less than one. This means that about one sixth to one fifth of the paraset’s substitutes are shared among all occurrences of the same target–sense combination.

Though it is reassuring that a common core exists more often than not, the fact that our same-sense paraset have more non-shared than shared substitutes is interesting. Part of the explanation for this is that some of the substitutes proposed by the annotators are highly context-specific, and do not apply to other instances even when used in the same word sense. For example, one of the contexts for *Athlet* (“athlete”) is as follows:

- (26) Seine eher mäßige schauspielerische Begabung rechtfertigte Weissmüller mit den Worten: „Das Publikum verzeiht meine Schauspielerei, weil es weiß, dass ich ein *Athlet* bin.“
[Weissmuller justified his rather modest acting talent by saying, “The public will forgive my acting because they know that I’m an *athlete*.”]

Here the paraset was {*Wettkämpfer*, *Sportler*, *Muskelprotz*, *Olympionike*, *Herkules*}, but the common core included only *Wettkämpfer* and *Sportler*. While the other three terms are plausible synonyms for this broad sense of *Athlet*, they would not necessarily fit every context. In particular, *Olympionike* (“Olympian”) suggests that one of the annotators has exploited his or her real-world knowledge of the context’s subject (in this case, Hollywood actor and competitive swimmer Johnny Weissmuller).

Another factor contributing to the low proportion of common-core substitutes is the sample size. As Kremer *et al.* (2014) observe, even six annotators cannot be expected to exhaust all possible substitutes for a given context. In fact, our common core statistics are only slightly lower than ones reported for comCo. In that data set, only about a quarter to a third of paraset substitutes were found in their respective common cores.

7.5 CONCLUSION

In this chapter, we have presented GLASS, a manually sense- and substitution-annotated German-language data set. GLASS is unique in providing both sense and lexical substitution annotations for the same targets. Our intention in doing this was to enable the data set to be used for both *in vitro* and *in vivo* evaluations of word sense

disambiguation systems. Though many of the lemmas in GLASS are monosemous in GermaNet, our data is still useful for intrinsic evaluations where systems must distinguish not only between senses provided by the inventory but the special “unassignable”/“proper name” tags that indicate a sense is missing from the inventory. GLASS has the further advantage of having the greatest lemma coverage of any manually sense-annotated data set for German. And unlike some other data sets which lack verbs or adjectives, it features an equal distribution across parts of speech (as well as lemma frequency).

The two annotation layers in GLASS have enabled us to conduct the first known empirical study of the relationship between manually applied word senses and lexical substitutions. Contrary to expectations, we found that synonymy, hypernymy, and hyponymy are the only semantic relations directly linking targets to their substitutes. Moreover, the substitutes in the target’s hypernymy/hyponymy taxonomy tend to closely align with the synonyms of a single synset in GermaNet. Despite this, these substitutes account for a minority of those provided by the annotators. Nearly two thirds of the substitutes exist somewhere in GermaNet but cannot be reached by traversing the target’s hypernymy/hyponymy taxonomy, and a sixth of the substitutes are not covered by GermaNet at all. These findings could be used to inform the design of future automatic lexical substitution systems.

The results of our analysis accord with those of a previous study on English-language data, but where the sense annotations were induced from the substitution sets by a fully automatic process. From this we can draw a couple of tentative conclusions. First, the relations the two studies discovered between word senses and lexical substitutions may prove to be of a universal nature, holding for other data sets and languages. Second, we have gathered good (albeit indirect) evidence that lexical substitution data can be used as a knowledge source for automatic wsd. This finding suggests that training data annotated with respect to a fine-grained sense inventory such as WordNet could be produced semi-automatically, by deriving it from relatively cheap, manually applied lexical substitution annotations.

One of Ubyline’s more innovative features is its ability to record timestamps for all annotator activity. One possible direction for future work, then, would be to analyze the timing data we have recorded in the production of GLASS. This would reveal whether there are any correlations between annotation time and the various properties of the target word or its contexts. Not only could this help predict annotation time for future data sets, but it may also be useful for assessing text difficulty in a readability setting.

Word sense disambiguation is a core research problem in computational linguistics, with applications in machine translation, information retrieval, and information extraction. Despite decades of research, how to build accurate general-purpose disambiguation systems remains an elusive open problem. Part of the difficulty lies in the myriad ways in which the task of wsd can be framed; an approach that works well in one scenario may fall completely flat in other. Another source of difficulty lies in the lexical-semantic knowledge sources that disambiguation systems depend on: in some cases the knowledge sources lack the necessary quantity or quality of information about words or senses, while in others this information may be present but too finely categorized to be immediately useful. This dissertation has concerned itself with ways in which these problems can be mitigated. In the following sections, we summarize our contributions and identify some applications and areas for future development of our ideas.

8.1 SUMMARY

In Chapter 3, we described how techniques from distributional semantics can be used to overcome the lexical gap in knowledge-based wsd. We began by obtaining a distributional thesaurus—a lexical-semantic resource which groups words according to their semantic similarity, and which was constructed using a large, automatically annotated background corpus. The entries in this thesaurus were then used to enrich the context and sense representations consulted by gloss overlap-based disambiguators. The improvement in accuracy resulted in state-of-the-art performance for knowledge-based wsd, exceeding even a contemporary approach based on word embeddings learned from neural networks, and in some cases even a naïve supervised baseline. Since our approach requires little more than a machine-readable dictionary and a raw text corpus, it is particularly attractive for wsd in under-resourced languages and domains, where the sense-annotated data and wordnets required by more complex approaches may not exist.

In Chapter 4, we presented another approach for closing the lexical gap in wsd. Specifically, we enriched the sense representations of WordNet by merging them with those of two complementary resources, Wikipedia and Wiktionary. We did this by inducing an alignment between the senses of the three resources. Rather than developing an alignment algorithm customized for the three resources, we

employed a general-purpose method for merging arbitrary numbers of pre-existing pairwise alignments. Use of our aligned resource led to significantly higher accuracy with gloss overlap-based disambiguation.

Another potential application of our approach to aligning complementary resources is word sense clustering. That is, the technique can be used to coarsen an existing sense inventory whose distinctions are too fine-grained for a given application. Though the particular clustering induced by our alignment did not prove helpful for word sense disambiguation, we nonetheless advanced the state of the art in this task by developing a more accurate cluster evaluation metric.

Chapter 5 built upon the previous two chapters by applying their techniques to the novel task of pun disambiguation. Traditional approaches to word sense disambiguation assume that words are used more or less unambiguously; they are unable to cope with puns, where lexical-semantic ambiguity is used to deliberate effect. We explained how evaluation metrics, baselines, and algorithms from wsd can be adapted to identifying the double meanings of puns, and tested these adaptations in a controlled setting. For this purpose we constructed a large, manually annotated data set of homographic puns. Our experiments showed pun disambiguation to be a particularly challenging task, though dramatic improvements were made possible through the use of our lexical expansion and sense coarsening techniques.

Our final two contributions were to produce and describe enabling technology and resources for word sense disambiguation. In Chapter 6, we introduced DKPro WSD, a modular framework for rapid development and evaluation of wsd systems. DKPro WSD mitigates the fragmentation in data formats, language resources, software components, and workflows which has long obstructed the interoperability and extensibility of disambiguation systems and the reproducibility of their results. We implemented the framework in UIMA, an industry-standard information processing architecture, which allows it to reuse existing language processing components with little or no modification.

Our final contribution is GLASS, a German-language lexical sample data set. The GLASS data set is unique in featuring high-quality, manually applied annotations for both word senses and lexical substitutions. Unlike many previously published data sets, it is well balanced with respect to the target words' frequency and part of speech, and is of sufficient size to be useful for machine learning approaches. It has the greatest lemma coverage of any manually annotated data set for German, and is the only one to be freely licensed in full.

We availed ourselves of the two annotation layers in GLASS to conduct the first known empirical study of the relationship between manually applied word senses and substitutions. We discovered that syn-

onymy and hyponymy/hypernymy are the only semantic relations directly linking targets to their substitutes, and that substitutes found in the target’s hyponymy/hypernymy taxonomy closely align with the synonyms of a particular sense in GermaNet. These findings accord with those of a previous English-language study using an automatically annotated data set. This suggests that our findings may hold across all data sets and languages, and that lexical substitution data is valuable knowledge source for automatic wsd.

8.2 OUTLOOK

In this section, we discuss the limitations of the research we have described here and identify some opportunities for future work.

In Chapters 3 and 5, we demonstrated how lexical expansions from distributional thesauri (DTs) can dramatically improve the accuracy of knowledge-based word sense disambiguation systems. Nonetheless, we observed that many of the expansions generated by the method were spurious. The problem can be traced to the fact that DTs are static knowledge sources which reflect distributions in the background corpus rather than the context of the word to be disambiguated. One way this could be addressed is to alter the lexical expansion mechanism to be sensitive to the context—something that is captured, for example, in LDA sampling. Another possible improvement would be to weight the expansions according to the DT similarity score.

Chapter 4 presented our approach to automatically merging arbitrary numbers of pairwise alignments of lexical-semantic resources, and used this approach to produce a three-way alignment of WordNet, Wikipedia, and Wiktionary. Manual examination of the merged synonym sets revealed the larger ones to be particularly noisy. Future work could therefore be directed to refinement of the approach to reduce the incidence of error cascades. One way of doing this would be to extend the alignment technique to filter outlier senses. Alternatively, the original technique could be retained, but applied only to existing pairwise alignments which favour precision over recall.

Despite its shortcomings, the aligned resource we produced was shown to significantly increase the accuracy of gloss overlap-based disambiguation. However, in that evaluation we made use of the resources’ glosses only. Each of the resources we aligned provides much richer lexical and semantic information, such as synonyms and semantic relations, which could also be exploited. It would be interesting to test our aligned resource with other wsd algorithms which make use of such information and see if there is any improvement in accuracy.

In Chapter 5 we introduced the novel tasks of pun detection and interpretation, and conducted some pioneering experiments using

methods adapted from traditional knowledge-based word sense disambiguation. The difficulty in collecting sufficient numbers of manually annotated puns rules out the use of most supervised disambiguation algorithms, at least for the task of pun interpretation. However, we are interested in testing further knowledge-based disambiguation algorithms for this task, particularly ones which rely on knowledge sources other than sense glosses. We are also interested in investigating alternative tie-breaking strategies, such as the domain similarity measures used by Mihalcea, Strapparava, and Pulman (2010) in their study on incongruity detection.

Computational processing of puns has heretofore been concerned almost exclusively with pun generation, or with their interpretation in highly structured forms such as knock-knock jokes. By contrast, our work is aimed at producing general-purpose methods for detection and interpretation of puns in arbitrary running text. Though we have restricted our experiments to homographic puns, we believe the methods we have developed would also work on imperfect puns. To validate this hypothesis, we could extend our data set to include imperfect puns and adapt our methods to use phonological theories of punning (Hempelmann, 2003a) to identify the target sense candidates.

Chapter 5 also discussed a number of possible applications of pun detection and interpretation—namely, to facilitate human–computer interaction in user interfaces implementing computational humour, to assist sentiment analysis in the advertising domain, to assist human translators in the identification and translation of ambiguous wordplay in comedic works, and to assist scholars in the digital humanities to identify and classify puns in literary *œuvres*. Future work could therefore be directed towards *in vivo* evaluation of pun detection and interpretation systems in these applications.

DKPro WSD, the software framework we covered in Chapter 6, has remained under application development since its initial release, with several new features planned or in progress. Implementations of or interfaces to further disambiguation algorithms, sense inventories, and data sets are underway. Perhaps the largest architectural question for future development is whether and how best to extend support for supervised algorithms. DKPro WSD’s current facilities for supervised WSD are focused on running existing classifiers; there is comparatively little direct support for engineering features and training classifiers on new data. Rather than writing such functionality from scratch, it may be possible to adapt or interface with general-purpose text classification frameworks such as DKPro TC (Daxenberger *et al.*, 2014).

Finally, our construction and analysis of the GLASS data set in Chapter 7 have opened up a number of possible avenues for further research. First, and most obviously, the data set can be used to evaluate German-specific and language-independent approaches to WSD.

Our own experiments with the knowledge-based systems presented in Chapter 3 are already underway. Second, we found that our data set’s parasetts tend to map to single senses in GermaNet, but that the majority of substitutions still come from outside the target’s hypernym/hyponym hierarchy. Further examination of these more distantly related senses could provide better insight into the nature of lexical substitutes and their relationship to target senses. Third, recall that Ubyline, the sense annotation tool we used to construct GLASS, recorded timestamps for all annotator activity. We intend to analyze this timing data to determine whether there are any correlations between annotation time and various properties of the target word or its contexts. This could help predict annotation time for constructing future data sets, and may also be useful for assessing text difficulty in a readability setting.

PUN ANNOTATION GUIDELINES

BACKGROUND AND SCOPE

In this task you are asked to annotate lexical units with their senses. However, unlike in a typical word sense annotation task, where each lexical unit to be annotated usually carries a single distinct meaning, the lexical units of interest in this study are puns. A `PUN` is a lexical unit which a speaker or writer uses in an intentionally ambiguous way, so that it conveys more than one distinct meaning.

By `LEXICAL UNITS` (also known as “lexical items” or “lexical entries”) we mean the single words or chains of words which form the basic elements of the vocabulary of English, and by `WORD`, we mean a sequence of letters bounded by space or by punctuation. A lexical unit is something whose lemma you would reasonably expect to find listed and defined in an English dictionary. A lexical unit is often a `SINGLE WORD`, such as “car” or “beach”. However, it can also be a (possibly discontinuous) `CHAIN OF WORDS`—examples of these include phrasal verbs such as “put up with” and “turn down”, or polywords such as “ice cream”, “motor vehicle”, and “run-through”.

In this study, we are studying puns which meet the following `INCLUSION CRITERIA`:

- The pun must be the *only* pun in the document in which it appears.
- The pun must consist of, or contain, only a *single* `CONTENT WORD` (*i.e.*, a noun, verb, adjective, or adverb). (Adverbial particles of phrasal verbs can be disregarded for the purposes of this criterion.) For example:
 - “car” meets this criterion because it has a single word which is a content word (a noun).
 - “to” does not meet the criterion. It has a single word, but it is neither a noun, verb, adjective, nor adverb.
 - “put up with” meets the criterion. Although it has several words, only one (“put”) is a content word (a verb). The other two words (“up” and “with”) are particles.
 - “ice cream” does not meet the criterion. It has two words, both of which are content words.
- The pun must have exactly *two* distinct meanings. (Note that it is extremely rare for a pun to have more than two distinct meanings.)



- The lexical units for the two distinct meanings must be *spelled exactly the same way*, except that you should disregard particles and/or inflections from one or both lexical units if doing so would make the spellings match. For example:
 - In the document, “The singing pony was a little hoarse today,” there is a pun on “hoarse” but it does not meet this criterion because the two meanings (“experiencing a dry voice” and “equine”) correspond to lexical units which are spelled differently, even after disregarding inflections and particles (“hoarse” versus “horse”).
 - In the document, “The marquis and the earl are duking it out,” there is a pun on “duking it out” which meets this criterion. The two meanings’ (“to fight” and “the ruler of a duchy”) lexical units share the same spelling (“duke”) after inflections and particles are disregarded.
 - In the document, “‘You have the right to remain silent,’ said Tom arrestingly,” there is a pun on “arrestingly”, but it does not meet this criterion because the two meanings (“to apprehend” and “sensationally”) correspond to lexical units which are spelled differently, even after disregarding inflections and particles (“arrest” or “arresting” versus “arrestingly”). (Recall that the suffix *-ly* is derivational, not inflectional.)

Note that on rare occasions, the two meanings of a lexical unit might correspond to words which are spelled the same but pronounced differently, such as “aks-IZ” versus “aks-EEZ” in the sentence, “A lumberjack’s world revolves on its axes.” Such cases are perfectly acceptable.

INSTRUCTIONS

To produce the annotations you will use the Punnotator, an online tool which you access through your web browser at <http://moe.ukp.informatik.tu-darmstadt.de:53280/~miller/Punnotator/start.php>

Main menu

When you first point your browser to the Punnotator, you will see a page with two sections:

ANNOTATE CORPUS. In this section you will see a list of all the corpora to be annotated in this study. Normally there will be only

Select all puns:
(Annotating sentence: 0, from file: 1 example.txt)

" Where do river otters keep their money " " At the bank ! "

NO PUN
☐

MULTIPLE PUNS
☐

OTHER
☐

Annotator: Continue

Figure 15. Punnotator pun selection page

one corpus listed here. To begin annotating a corpus, or to resume annotating from where you left off, simply click on the corpus name.

EDIT ANNOTATIONS. This section contains a link which you can follow to revise the annotations you have already made.

Annotating the corpus

When you choose to annotate a corpus, the Punnotator will first display a web page containing a document from the corpus. (See Figure 15.) These “documents” are always very short jokes or slogans, and have already been tentatively identified by another annotator as meeting the inclusion criteria for our study. On this web page, you need to verify that the document contains a pun which meets the inclusion criteria for our study, and if so, to mark it. For this you should use the following workflow:

1. Read the document.
2. Identify in your mind which lexical unit(s) in the document are puns, and very roughly what their meanings are.
You are free to consult any external sources of knowledge (dictionaries, encyclopedias, *etc.*) to help understand the document and the pun.
3. Eliminate any non-puns or puns not meeting the inclusion criteria using the following procedure:
 - a) If you judge that *none* of the lexical units in the document are puns, then check the “NO PUN” box and press the “Continue” button to skip ahead to the next document.
 - b) If you judge that *more than one* lexical unit in the document is a pun, then check the “MULTIPLE PUNS” box and press the “Continue” button to skip ahead to the next document.

- c) If you don't know whether any of the lexical units in the document are puns (for example, if you don't understand the document well enough to make a determination), then check the "OTHER" box and press the "Continue" button to skip ahead to the next document.
 - d) If you judge that exactly one lexical unit in the document is a pun, but that it has *more than two* distinct meanings, then check the "OTHER" box and press the "Continue" button to skip ahead to the next document.
 - e) If you judge that exactly one lexical unit in the document is a pun, but that it does not contain exactly one content word, then check the "OTHER" box and press the "Continue" button to skip ahead to the next document.
 - f) If you judge that exactly one lexical unit in the document is a pun, but that the spellings for the two meanings do not match (even after disregarding particles and inflections), then check the "OTHER" box and press the "Continue" button to skip ahead to the next document.
4. Select the single content word in the lexical unit you identified as a pun by clicking on it. For example, if you identified "put up with" as the pun, then click only on "put". If you click on the wrong word, you can de-select it by clicking on it again. Once you have selected the correct word, press the "Continue" button to proceed to the sense annotation page.

When you select a pun meeting the inclusion criteria and click on the "Continue" button, the Punnotator will take you to a new page where you can annotate its senses. (See Figure 16.) This page shows the document and pun followed by a list of senses. The list will contain senses for the single content word you selected, grouped by part of speech. Following this there may also be senses for some multiword lexical units containing the content word you selected, also grouped by part of speech. Senses are identified by their definitions, synonyms, and example sentences, as extracted from WordNet. To the left of each sense are two checkboxes, in columns labelled "s1" and "s2".

To annotate senses, you should use the following workflow:

1. Identify in your mind one of the two meanings of the pun. Then, read through the list of senses presented to see which of them match this meaning.
2. If *none* of the senses listed match the first meaning of the pun because the pun is a proper name, check only the first column's "Proper Name" box near the bottom of the list.

"Where do river otters keep their money" "At the **bank**!"

Select the two sense sets of the word **bank**: **bank (noun)** & **bank (verb)**

S1	S2	noun definitions
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => the high land (especially the slope beside a body of water) (e.g. they pulled the canoe up on the bank • he sat on the bank of the river and watched the currents)
<input type="checkbox"/>	<input type="checkbox"/>	bank (depository financial institution, bank, banking company, banking concern) => a financial institution that accepts deposits and channels the money into lending activities (e.g. he cashed a check at the bank • that bank holds the mortgage on my home)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => a long ridge or pile (e.g. a huge bank of earth)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => an arrangement of similar object in a row or in tiers (e.g. he operated a bank of switches)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => a supply or stock held in reserve for future use (especially in emergencies)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => the funds held by a gambling house or the dealer in some gambling games (e.g. he tried to break the bank at Monte Carlo)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank, counter, curb) => a slope in the turn of a road or track the outside is higher than the inside in order to reduce the effect of centrifugal force
<input type="checkbox"/>	<input type="checkbox"/>	bank (savings bank, money box, bank, coin bank) => a container (usually with a slot in the top) for keeping money at home (e.g. the coin bank was empty)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank, bank building) => a building in which the business of banking transacted (e.g. the bank is on the corner of Nassau and Wallenpoort)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => a flight maneuver; aircraft tip laterally & enter a longitudinal axis (especially in turning) (e.g. the plane went into a steep bank)
S1	S2	verb definitions
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => tip laterally (e.g. the pilot had to bank the aircraft)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => enclose with a bank (e.g. bank roads)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => do business with a banker or keep an account at a bank (e.g. Where do you bank in this town?)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => act as the banker in a game or in gambling
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => be in the banking business
<input type="checkbox"/>	<input type="checkbox"/>	bank (deposit, bank) => put into a bank account (e.g. She deposits her paycheck every month)
<input type="checkbox"/>	<input type="checkbox"/>	bank (bank) => cover with ashes so to control the rate of burning (e.g. bank a fire)
<input type="checkbox"/>	<input type="checkbox"/>	bank (count, calculate, bet, bank, depend, reckon, rely, swear, look) => have further confidence in (e.g. you can count on me to help you any time • Look to your friends for support • You can bet on that! • Depend on your family in times of crisis)
S1	S2	Proper name - Unassigned
<input type="checkbox"/>	<input type="checkbox"/>	Proper Name
<input type="checkbox"/>	<input type="checkbox"/>	Unassigned - Unknown

Submit & Next Editor

Figure 16. Punnotator pun annotation page

3. If *none* of the senses listed match the first meaning of the pun for any other reason, check only the first column's "Unassigned – Unknown" box near the bottom of the list.
4. Otherwise, in the first column of checkboxes, check all those senses which match this meaning. Determining what constitutes a "matching" sense is somewhat subjective, though the following guidelines should help:

- Reviewing the example sentences next to a sense can often help determine whether it is a good match.
- The candidate senses presented may be overly fine-grained. If the meaning you have in mind is more general, or if it is not clear which of the various fine-grained senses is the one you intend, then check *all* the fine-grained senses which your meaning plausibly subsumes.

For example, in the document, "Where do river otters keep their money? 'At the bank!'" there is a pun on "bank". The sense list for "bank" presented by Punnotator includes both "a financial institution that accepts deposits and channels the money into lending activities" and "a building in which the business of banking [is] transacted". One of the pun's meanings clearly refers to some sort of financial entity, though not with sufficient precision to distinguish between the banking organization itself and the building it occupies. So in this case both of these senses should be checked.

- Usually all the senses for the meaning you have in mind will be from the same part of speech. However, some puns evoke a meaning in a way which transcends or obscures the part-of-speech distinctions among closely related senses. (This is particularly common with gerunds and past participles, which can often be analyzed as being nouns or adjectives in addition to verbs. It can also occur with words such as "pin" which have closely related noun and verb senses.) In such cases you should select *all* the senses which could plausibly be considered to be correspond to the meaning you have in mind.
- If the lexical unit for the meaning you have in mind contains multiple words, you should generally restrict yourself to selecting senses from the multiword lexical units near the end of the list. Avoid selecting from the single-word lexical units unless the meaning is not represented among the multiword lexical units.

Note that Punnotator's selection of multiword lexical units at the bottom of the list may be overly broad. Do not select a sense

for a multiword lexical unit unless *all* the words in the lexical unit are contained, in some form, in the document. For example, if in the sentence “A lumberjack’s world revolves on its axes” you select “axes” as the pun word, Punnotator might show you (among many other senses) the sense for the lexical unit “axis of rotation”. You should not select this sense, because the words “of” and “rotation” (or any inflected forms of these words) do not appear in the sentence.

5. Repeat all the steps above for the second meaning of the pun, this time selecting senses from the second column. Note that the Punnotator will not permit you to select the same sense in both columns.
6. If at any time you realize that you were mistaken in thinking that the current document contains a pun which meets the inclusion criteria, or if you realize that you selected the wrong content word for annotation, you can always use your browser’s Back button to return to the pun selection page and fix your mistake.
7. Once you are satisfied with your selections for both columns, press the “Submit” button to proceed to the next document.

Interrupting, resuming, and editing annotations

You can interrupt your annotation session at any time, and return to it by visiting the Punnotator home page and clicking on the name of the corpus you were annotating. Punnotator will then return you to the document you were annotating (or were about to annotate) when you interrupted your session.

From the Punnotator home page you can also revise the annotations you have already applied to documents in the corpus. To do this, click on the “Edit annotations” link. You will be taken to a page listing all the documents and the annotations you have applied to them. Next to each document is an “Edit” link which you can use to revise your annotations.

GLOSSARY

CONTENT WORD A noun, verb, adjective, or adverb.

INFLECTION The modification of a word to express different grammatical categories. Recall that in English there are only eight inflectional affixes: the plural *-(e)s* and possessive *-(’s)* for nouns, the comparative *-(e)r* and superlative *-(e)st* for adjectives, and

the 3rd singular present *-(e)s*), past tense *-(e)d*), past participle *-(e)n*), and present participle *-(ing)*. Inflection is sometimes marked irregularly, such as through the vowel change of “see” to its past tense “saw”.

LEXICAL UNIT One of the single words or chains of words which form the basic elements of the vocabulary of English. A lexical unit is something you would reasonably expect to find listed and defined in an English dictionary.

PARTICLE A preposition, adverb, or other function word which forms part of a phrasal verb.

PHRASAL VERB A lexical unit consisting of a verb plus one or more particles. Examples: “look after”, “sit in for”.

WORD A sequence of letters bounded by space or by punctuation. For example, the string “setup” is one word, “set-up” is two words, and “put up with” is three words.

BACKGROUND AND SCOPE

In this task you are asked to adjudicate between two sets of conflicting sense annotations for English puns.

A *pun* is the deliberately ambiguous use of a word, usually for humorous purposes. In an earlier phase of this project, we constructed a data set consisting of short texts supposedly containing a single pun, and for each text we asked two human annotators to identify which word (if any) was the pun. If the annotator selected a pun word, we then presented them with all its senses (as extracted from WordNet, an electronic dictionary and semantic network). For each of the two meanings of the pun, the annotator selected the WordNet sense(s) corresponding to it. (Because WordNet's senses tend to be overly fine-grained, and because puns often transcend part-of-speech distinctions, annotators had the option of selecting more than one WordNet sense for each meaning.) If none of the senses from WordNet reflected the pun meaning, the annotators could mark the meaning as a "proper name" or as "unknown/unassignable".

In many cases our two annotators fundamentally disagreed on the sense annotations in one of the following ways:

1. One annotator thought the text contained a single pun, whereas the other annotator thought the text contained zero puns or more than one pun.
2. The two annotators identified a different word in the text as being the pun.
3. The two annotators selected the same word as the pun, but the WordNet senses they selected for the meanings were contradictory or disjoint.

Some or all of these disagreements may have arisen from one or both of the annotators misunderstanding the pun text, overlooking the correct senses, and/or mistakenly selecting the wrong senses. However, some or all of the disagreements may rather have arisen from the annotators having different but equally valid interpretations of the pun.

Your role as the adjudicator is to review the sense annotation pairs where the annotators disagreed, and, where possible, to resolve the disagreement in favour of one of the annotators.

B

[Index](#)

Text ID: pun_7

they stand best who kneel most

	ANNOTATOR 1	ANNOTATOR 2
PUN WORD	stand	stand
SENSE 1	stand (<i>verb</i>) stand, stand up ▶ be standing, be upright, "We had to stand for the entire performance!"	stand (<i>verb</i>) stand, stand up ▶ be standing, be upright, "We had to stand for the entire performance!"
SENSE 2	stand (<i>verb</i>) stand ▶ occupy a place or location, also metaphorically, "We stand on common ground" stand (<i>verb</i>) stand ▶ be in some specified state or condition, "I stand corrected"	stand (<i>verb</i>) digest, endure, stick out, stomach, bear, stand, tolerate, support, brook, abide, suffer, put up ▶ put up with something or somebody unpleasant, "I cannot bear his constant criticism", "The new secretary had to endure a lot of unprofessional remarks", "he learned to tolerate the heat", "She stuck out two years in a miserable marriage"
SHOW UNUSED		
ADJUDICATION	<input type="button" value="Annotator 1"/> <input type="button" value="Can't decide"/> <input type="button" value="Annotator 2"/>	

Figure 17. Pun annotation adjudication page

INSTRUCTIONS

First of all, read through the attached “Pun annotation guidelines” manual given to the two annotators, so that you understand exactly what they were asked to do.

To make your adjudications you will use an online tool which you access through your web browser at <http://moe.ukp.informatik.tu-darmstadt.de:53280/~miller/ita/>

Adjudication index

When you point your browser at the adjudication tool, you will be presented with a list of all the adjudications you have made so far. The first time you visit, the list will be empty, but on subsequent visits the list will show a unique ID for each pun text along with your decision.

To begin adjudicating, or to resume from where you left off, follow the **CONTINUE** link at the bottom of the page.

Alternatively, you can follow a link to any of the pun texts you have previously adjudicated, and this will allow you to revise your decision for that text.

Performing adjudications

A separate adjudication page is shown for each pun text, as shown in Figure 17. At the top of the page is a link which takes you back to the adjudication index. Underneath, on the left, you can see the unique ID for this pun text. Lower still, in large type centered on the

page, is the pun text, followed by a table showing the two annotators' annotations. Excluding the headers, the table has two columns (one for each annotator) and five rows:

- **PUN WORD** shows which word from the text the annotator identified as being the pun, or "no valid pun" if the annotator indicated that the text contains no puns or contains more than one pun.
- **SENSE 1** shows all the WordNet senses selected by the annotator as corresponding to one of the meanings of the pun. (For each sense, we show the following information from WordNet: the lemma, part of speech, synonyms, definition, and example sentences.) If none of the senses in WordNet fit the meaning, the cell shows "PROPER" (if the sense corresponded to a proper name) or "UNKNOWN". If the annotator did not select a pun word for this text, this cell contains "N/A".
- **SENSE 2** shows all the WordNet senses selected by the annotator as corresponding to the other meaning of the pun. As above, this row may also contain "PROPER", "UNKNOWN", or "N/A".
- If you want to see a list of all the WordNet senses *not* selected by each annotator, follow the **SHOW UNUSED** link in the fourth row.
- Finally, the **ADJUDICATION** row contains three buttons which you can use to record your adjudication decision.

To perform the adjudications, please use the following workflow:

1. Read the pun text.
2. Identify in your mind which word(s) (if any) in the text are puns, and very roughly what their meanings are.
3. Read the pun word annotations and sense annotations made by the two annotators. Consider which of the two annotators has more correctly identified the pun word (if any) and its meanings. (In some cases—particularly when some of the sense annotations are marked as "UNKNOWN" or "PROPER"—it may help to use the **SHOW UNUSED** link to show the senses not selected by the annotators.)
4. If after reading the pun text and the annotations, you do not understand the pun, then press the **CAN'T DECIDE** button.
5. If you believe both annotators' annotations are about equally correct or equally incorrect—for example, you think the annotators had different but valid interpretations of the pun, or you think both annotators misunderstood or improperly annotated the pun—then press the **CAN'T DECIDE** button.

6. Otherwise, press the button of the annotator whose annotations you consider to be more correct.

INTERRUPTING, RESUMING, AND EDITING ADJUDICATIONS

You can interrupt your adjudication session at any time, and return to it by visiting the index page and following the `CONTINUE` link. The adjudication tool will then return you to the text you were about to adjudicate.

From the index page you can also revise the adjudications you have already applied. Simply follow the link corresponding to the `ID` of the text you want to re-adjudicate.

SENSE LINKING GUIDELINES

BACKGROUND AND SCOPE

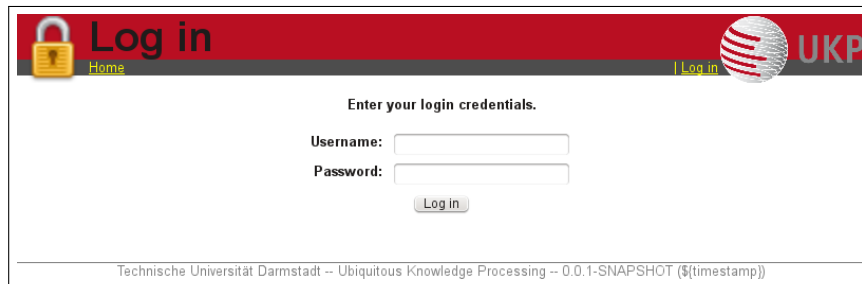
Most words take on different meanings according to the context they're used in. In this task you are asked to indicate the meaning of certain German words when they are used in different sentences.

There are 153 different target words in this study—51 nouns, 51 verbs, and 51 adjectives. For each target word, we will show you a list of different sentences containing the word, along with a list of possible senses for the word. Your job is to link each sentence to the sense(s) corresponding to how the target word is used. There are 2040 different sentences in all (10 each for the nouns and adjectives, and 20 each for the verbs).

INTRODUCTION TO UBYLINE

To produce the sense links you will use Ubyline, an online annotation tool which you access through your web browser at <http://ubyl ine.ukp.informatik.tu-darmstadt.de/ubyl ine/>.

Log in



The screenshot shows the Ubyline login interface. At the top, there is a red navigation bar containing a yellow padlock icon, the text 'Log in', a 'Home' link, a 'Log in' link, and the UKP logo. Below this bar, the main content area is white and contains the instruction 'Enter your login credentials.' followed by two input fields: 'Username:' and 'Password:'. A 'Log in' button is positioned below the password field. The footer of the page displays the text 'Technische Universität Darmstadt -- Ubiquitous Knowledge Processing -- 0.0.1-SNAPSHOT (\$){timestamp})'.

Figure 18. Ubyline login page

First, log in using the username and password provided to you by the experimenter.

Home page

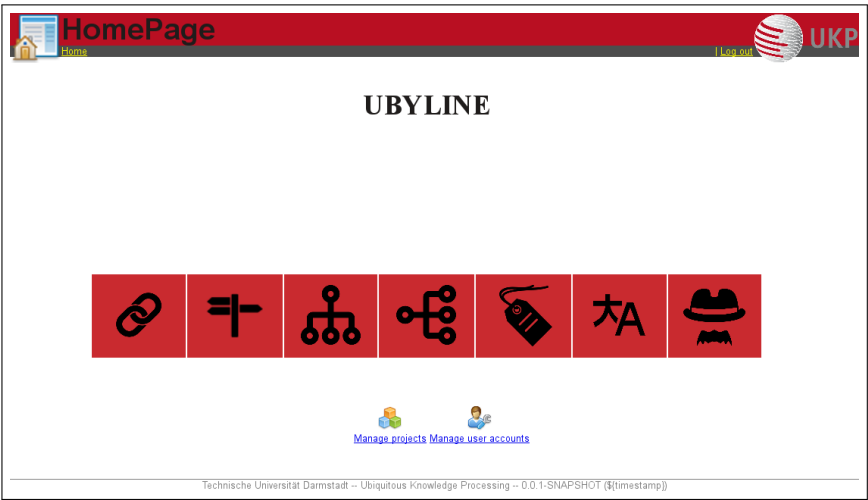


Figure 19. Ubyline home page

After logging in you will be taken to the Ubyline home page, which shows an icon-based menu of various annotation tools. Click on the “Link examples to senses” icon to be taken to the sense linking overview page.

Sense linking overview page

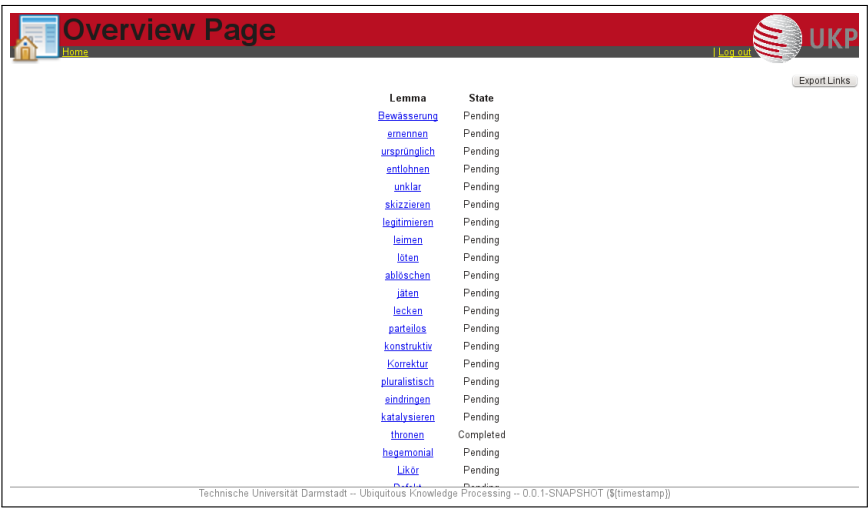


Figure 20. Ubyline sense linking overview page

The overview page shows the list of 153 target words (“lemmas”) in the task. Target words whose sentences you have already linked are marked as “Completed”, and those which you haven’t yet processed

are marked as “Pending”. You can click on any word to add or modify its sense links.

Also on the overview page, in the top right corner, is an “Export links” button which you can use to save your sense links to an XML file.

Sense example linking page

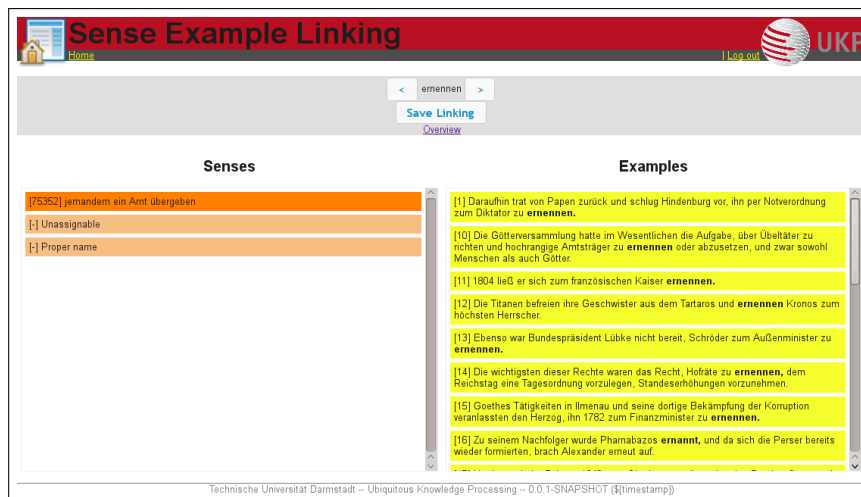


Figure 21. Ubyline sense example linking page

Each of the 153 target words has its own sense example linking page. The top of this page is labelled with the target word itself. To the word’s left and right are two navigation buttons which take you to the previous and next target word, respectively, and under the word is a link which returns you to the overview page.

The rest of the page is split into two scrolling areas: on the left is the “Senses” list, which lists one or more *senses* (meanings) of the target word, and on the right is the “Examples” list, which lists 10 or 20 numbered sentences containing the target word (marked up in boldface).

The senses have the following three-part format:

[sense_ID] (list_of_synonyms) definition

In addition, if you hover your mouse pointer over a sense, Ubyline will show you the sense’s hypernyms and hyponyms. A *hypernym* is a more general term which includes specific instances of the sense, and conversely a *hyponym* is a more specific term. A hyponym shares a *kind-of* relationship with its hypernym. For example, one of the senses of the word *Hund* has the hypernym *Tier* and the hyponym *Dackel*, because a *Hund* is a kind of *Tier*, and a *Dackel* is a kind of *Hund*.

Every sense will have a Sense ID, but not every sense will have a list of synonyms or a definition. For some senses—particularly those missing a definition—you will need to examine its hypernyms and hyponyms in order to understand it.

You may very occasionally find that the definition provided by GermaNet is fairly specific, but the hyponyms clearly refer to more general concepts. In such cases, you can assume the sense is actually the broader one implied by the hyponyms.

In addition to the regular senses, the “Senses” list contains two special senses: “Proper name” and “Unassignable”. Their use is explained in the next section.

To link an example sentence to a sense, drag it from the “Examples” list to a position underneath the appropriate sense in the “Senses” list. If you make a mistake, you can relink the example sentence by dragging it within the “Senses” list, or drag it back to the “Examples” list to remove the link altogether.

Every example sentence must be linked to at least one sense. It is possible to link an example sentence to more than one sense.

When you are done linking every example to at least one sense, save your links by pressing the “Save Linking” button near the top of the page. Ubyline will check that you have linked every example to at least one sense—if you forgot to link an example, Ubyline will warn you about this, and will refuse to save your links until you have corrected the problem.

SENSE LINKING INSTRUCTIONS

For this task, please produce sense links for all 2040 sentences. Use the following procedure for each target word:

1. Read the target word.
2. Read through all the senses in the “Senses” list. If the meaning of any sense is not clear to you (which may be the case particularly when the sense lacks synonyms or definitions), examine its hypernyms and hyponyms by hovering your mouse pointer over the sense.
3. For each example sentence in the “Examples” list:
 - a) Read the sentence and try to understand the meaning of the target word it contains. (Note that the target word may be part of a compound—if so, try to understand the specific meaning of the target word within this compound.)
 - b) If the word’s meaning corresponds to a sense in the “Senses” list, drag the example sentence to a position under that sense.

It is possible that more than one of the senses in the “Senses” list is a good match for the word. (For example, there may be two senses which are very specific but the word is used in a general way which would cover both senses.) In such cases, drag a separate copy of the example sentence from the “Examples” list to each of the matching senses in the “Senses” list.

- c) If the word’s meaning is missing from the “Senses” list because it refers to a proper name, then drag the example sentence underneath the special “Proper name” sense. (For example, the target word *Essen* should be linked to the “Proper name” sense for an example sentence such as *Essen ist eine Großstadt im Zentrum des Ruhrgebiets*, but not for *Ungarisches Essen ist nicht prinzipiell scharf gewürzt*.)
- d) If the word’s meaning is missing from the “Senses” list for any other reason, or if you don’t understand the meaning of the word, then drag the example sentence underneath the special “Unassignable” sense.

Every example sentence *must* be linked to at least one sense, though it is possible that some senses will not have any linked example sentences.

IMPORTANT NOTE: The Ubyline system keeps track of the time you spend performing sense links for each target word. We are not doing this to test your abilities, but rather to give us an idea of how difficult each target word is to understand! Do not feel that you need to rush through the task. However, it is very important that you do not get interrupted on the sense linking page. If you need to take a break, please finish sense-linking all the examples of the current target word first, and then return to the Overview page.

You are free to go back and revise your sense links at any time. (Return to the Overview page to see the full list of target words.) Remember to use the “Save Linking” button to save your changes.

Once you have finished linking all the example sentences for all the target words (that is, all target words are marked as “Completed” on the overview page), use the “Export links” button on the overview page to download an XML file containing your sense links. Send this XML file by e-mail to the experimenter.

SENSE LINKING ADJUDICATION GUIDELINES

BACKGROUND AND SCOPE

In this task you are asked to adjudicate between two sets of conflicting sense annotations for German words.

In an earlier phase of this project, we constructed a data set consisting of 2040 short texts containing a target word, and for each text we asked two human annotators to identify the meaning of that target. The annotators selected these meanings from those listed in GermaNet, a semantic network for the German language. Sometimes words are used in a vague or underspecified manner, so annotators had the option of assigning more than one possible “correct” sense for a given target. The annotators also had two special sense annotations they could apply (either alone or in combination with the ones from GermaNet):

- If GermaNet did not contain the sense of the target word because it was being used as a proper noun, they could annotate it as a “proper noun”.
- If GermaNet did not contain the sense of the target word for any other reason, or if they did not understand the meaning of the word, they could annotate it as “unassignable”.

In 329 cases, our two annotators disagreed on the sense annotation. Your role as adjudicator is to review these disputed sense annotations and to resolve the disagreement in favour of one or both of the annotators.

INSTRUCTIONS

First of all, read through the attached “Sense linking guidelines” manual given to the two annotators, so that you understand exactly what they were asked to do.

To make your adjudications, you will use an online tool which you can access your web browser at <http://www.nothingisreal.com/adjudication/>.

When you first point your browser at the adjudication tool, you will be presented with an index of all the lemmas whose annotations need adjudication. To begin or resume your adjudications, follow the link to the first lemma you haven’t yet completed.

There is a separate adjudication page for each lemma. At the top of the page is a heading showing the current lemma. Immediately below

D

this is a table showing the GermaNet senses for this lemma. For each sense, the table shows the following information:

- a numeric ID
- a list of the sense's synonyms (other than the lemma itself)
- a list of the sense's definitions, a list of example sentences showing the lemma being used in this sense
- the synonyms of the sense's hypernyms
- the synonyms of the sense's hyponyms

Note that, except for the ID, not every type of information may be available for every sense.

Below the sense table, the page is divided into sections, one for each text. There is a heading showing a unique identifier for the text, followed by the text itself, with the target word highlighted in bold-face. Below this is a table showing the annotations made by the two annotators (referred to as "Annotator A" and "Annotator B"), as well as their union ("Both"). The annotations are shown as numeric IDs corresponding to the ones in the sense table at the top of the page, or "U" or "P" for "unassignable" and "proper noun" senses, respectively.

For each text, compare the senses assigned by the two annotators to decide which of them better capture the meaning of the target word in this text. (It may be necessary, particularly when encountering "P" or "U" annotations, to refer back to the "Sense linking guidelines" to satisfy yourself that the annotators correctly followed the annotation instructions.) Then apply your adjudication as follows:

- If the senses assigned by Annotator A better match the meaning of the target word in this text, click on the "Annotator A" radio button.
- If the senses assigned by Annotator B better match the meaning of the target word in this text, click on the "Annotator B" radio button.
- If all the senses assigned by Annotator A and all the senses assigned by Annotator B are an equally good match for the meaning of the target word in this text, click on the "Both" radio button.

Once you have adjudicated every text on the page, press the "Submit adjudications" button. Your adjudications will be recorded and you will be forwarded to the next lemma adjudication page (or back to the index, if there are no further lemmas to adjudicate). Note that the adjudication tool does not warn you if you forgot to make a decision for every text, so make sure you have made all adjudications before submitting!

If, after submitting the adjudications for a lemma, you change your mind and want to revise them, you can always revisit the lemma page by using your browser's "Back" button, or by selecting the lemma from the tool's index page. However, this may cause all the radio buttons on the page to be reset—if this happens, then you must re-adjudicate *all* the texts for that lemma. Once you have made your revisions, press the "Submit adjudications" button again to record them; your previous adjudications for this lemma will be overwritten.

You may interrupt your adjudications at any time. You can resume them by pointing your browser to the tool's index page and following the link to the lemma where you left off.

GLASS INTERANNOTATOR AGREEMENT

The following table shows various statistics for the GLASS data set (Chapter 7). For each lemma, we show its part of speech (POS), its number of occurrences in the data set (#), its number of senses in GermaNet 9.0 (δ), and various measures interannotator agreement: the raw percentage agreement (%), the mean Dice coefficient (Dice), Cohen's κ (κ), and Krippendorff's α using the MASI distance metric (α). The figures for Cohen's κ exclude items for which one or both annotators applied multiple sense annotations. Lemmas for which such items exist are marked with a + in the occurrences column.

LEMMA	POS	#	δ	%	DICE	κ	α
Abbuchung	n	10	2	0.90	0.90	0.00	0.00
ablöschen	v	20	1	1.00	1.00	1.00	1.00
abmalen	v	20	1	0.90	0.90	-0.05	-0.03
abstürzen	v	20	2	0.90	0.90	0.75	0.76
abwechselnd	a	10	1	1.00	1.00	—	—
alkoholsüchtig	a	10	1	1.00	1.00	—	—
angestammt	a	10	1	0.30	0.30	0.00	-0.46
anklagen	v	20	2	0.80	0.83	0.28	0.19
anprobieren	v	20	1	0.95	0.95	0.00	0.00
anspitzen	v	20	1	1.00	1.00	—	—
Antrag	n	10+	3	0.90	0.97	0.00	0.75
Aristokratie	n	10	2	0.80	0.80	0.58	0.59
asexuell	a	10	1	0.90	0.90	0.00	0.00
assoziiieren	v	20	1	0.90	0.90	0.00	-0.03
Athlet	n	10	1	1.00	1.00	—	—
Auftrieb	n	10	3	1.00	1.00	1.00	1.00
ausführbar	a	10	1	1.00	1.00	—	—
ausgestattet	a	10	1	0.40	0.40	0.00	-0.36
aussagekräftig	a	10	1	1.00	1.00	1.00	1.00
Ballade	n	10	2	0.60	0.60	0.38	0.39
Bandit	n	10	1	0.80	0.80	0.44	0.44
Befestigung	n	10	3	0.80	0.93	0.70	0.94
befördern	v	20	2	0.95	0.95	0.88	0.88
Bekanntheit	n	10	2	0.50	0.70	0.07	-0.05
Belagerung	n	10	1	1.00	1.00	1.00	1.00
belehren	v	20	2	0.95	0.98	0.00	0.00
Besiedlung	n	10	1	0.90	0.90	0.00	0.00

E

LEMMA	POS	#	δ	%	DICE	κ	α
Bewässerung	n	10	1	1.00	1.00	—	—
brutal	a	10	2	0.30	0.43	0.01	-0.12
Bunker	n	10	3	0.80	0.87	0.71	0.80
Chronologie	n	10	1	0.90	0.90	0.00	0.00
deckeln	v	20	1	0.95	0.95	0.00	0.00
Defekt	n	10	2	0.90	0.90	0.80	0.80
demobilisieren	v	20	1	0.85	0.85	0.35	0.33
Diktatur	n	10	1	1.00	1.00	—	—
Distrikt	n	10	2	0.70	0.90	0.12	0.61
diversifizieren	v	20	1	0.85	0.85	0.71	0.71
drillen	v	20	1	1.00	1.00	—	—
Druckerei	n	10	1	1.00	1.00	—	—
dünnhäutig	a	10	1	1.00	1.00	1.00	1.00
duplizieren	v	20	1	0.80	0.80	0.00	-0.08
dynamisieren	v	20	1	0.75	0.75	0.14	0.16
eindringen	v	20	2	0.60	0.60	0.30	0.21
Entdecker	n	10	1	1.00	1.00	—	—
entlohn	v	20	1	0.95	0.95	0.00	0.00
Epos	n	10	1	0.90	0.90	0.00	0.00
Erkennung	n	10	1	1.00	1.00	—	—
Erleichterung	n	10	3	0.60	0.67	0.43	0.51
erlöschen	v	20	3	0.85	0.85	0.76	0.77
ernennen	v	20	1	1.00	1.00	—	—
Erschießung	n	10	1	1.00	1.00	—	—
exaltiert	a	10	1	0.90	0.90	0.00	0.00
explodieren	v	20	3	0.40	0.80	0.25	0.74
extern	a	10	1	1.00	1.00	—	—
Farm	n	10	1	0.90	0.90	0.00	0.00
fegen	v	20	3	0.60	0.60	0.25	0.24
fesselnd	a	9	1	1.00	1.00	1.00	1.00
Fluglinie	n	10	1	1.00	1.00	—	—
fundieren	v	20	1	0.95	0.95	0.00	0.00
Garnison	n	10	1	0.90	0.90	0.00	0.00
gedeihen	v	20	2	0.10	0.10	0.00	-0.77
glückbringend	a	10	1	0.90	0.90	0.00	0.00
grüblerisch	a	10	1	1.00	1.00	—	—
hegemonial	a	9	1	1.00	1.00	—	—
humorvoll	a	10	1	1.00	1.00	—	—

LEMMA	POS	#	δ	%	DICE	κ	α
hüten	v	20	2	0.95	0.95	0.91	0.92
inakzeptabel	a	10	1	1.00	1.00	—	—
inventarisieren	v	20	1	1.00	1.00	—	—
jäten	v	20	1	1.00	1.00	1.00	1.00
kameradschaftlich	a	10	1	1.00	1.00	—	—
Kannibale	n	10	1	0.80	0.80	-0.05	-0.03
katalysieren	v	20	1	1.00	1.00	—	—
koalieren	v	20	1	1.00	1.00	—	—
konstruktiv	a	10	1	0.70	0.70	0.40	0.37
kontrollierbar	a	10	1	1.00	1.00	—	—
Korrektur	n	10	1	0.80	0.80	0.00	-0.06
korrespondieren	v	20	1	1.00	1.00	—	—
Krieg	n	10	1	0.70	0.70	0.00	-0.12
kryptisch	a	10	1	0.90	0.90	0.00	0.00
Kürzel	n	10	1	0.90	0.90	0.00	0.00
laben	v	20	1	0.95	0.95	0.00	0.00
lancieren	v	20	1	0.10	0.10	0.01	-0.76
langjährig	a	10	1	1.00	1.00	—	—
lecken	v	20	3	0.95	0.95	0.92	0.92
legitimieren	v	20	1	0.10	0.10	0.00	-0.77
leimen	v	20	2	0.85	0.85	0.37	0.35
Lieferung	n	10+	2	0.80	0.87	0.55	0.68
Likör	n	10	1	1.00	1.00	—	—
löten	v	20	1	1.00	1.00	—	—
Mächtigkeit	n	10	1	1.00	1.00	1.00	1.00
mahlen	v	20	1	0.90	0.90	-0.05	-0.03
mehrfarbig	a	10	1	1.00	1.00	—	—
Meisterin	n	10	3	0.40	0.53	0.23	0.38
Metallurgie	n	10	2	0.30	0.70	-0.09	-0.19
Millionär	n	10	1	1.00	1.00	—	—
mitgehen	v	20	3	0.40	0.50	0.28	0.32
Monarch	n	10	1	1.00	1.00	—	—
multilingual	a	10	1	1.00	1.00	—	—
mutwillig	a	10	1	0.90	0.90	0.00	0.00
nett	a	10	1	0.90	0.90	0.00	0.00
offenbaren	v	20	2	1.00	1.00	1.00	1.00
optimieren	v	20	1	1.00	1.00	—	—
panikartig	a	10	1	1.00	1.00	—	—

LEMMA	POS	#	δ	%	DICE	κ	α
parteilos	a	10	1	0.80	0.80	0.00	-0.06
phlegmatisch	a	10	1	0.70	0.70	0.00	-0.12
Plünderung	n	10	1	1.00	1.00	—	—
pluralistisch	a	10	1	0.90	0.90	0.00	0.00
postulieren	v	20	2	0.85	0.88	0.36	0.44
preisgegeben	a	10	1	1.00	1.00	—	—
Problem	n	10	1	1.00	1.00	—	—
provisorisch	a	10	1	1.00	1.00	—	—
rasant	a	10	1	1.00	1.00	—	—
religiös	a	10	1	1.00	1.00	—	—
Rendite	n	10	1	1.00	1.00	—	—
Rezeption	n	10	2	0.20	0.20	-0.43	-0.58
Rivalität	n	10	1	0.90	0.90	0.00	0.00
schauspielerisch	a	10	1	1.00	1.00	—	—
schmettern	v	20	2	1.00	1.00	1.00	1.00
selbstgemacht	a	10	1	1.00	1.00	—	—
Signal	n	10	2	1.00	1.00	—	—
skizzieren	v	20	2	0.80	0.83	0.63	0.69
spezifisch	a	10	1	1.00	1.00	—	—
Streuung	n	10	1	0.90	0.90	0.00	0.00
strukturiert	a	10	1	1.00	1.00	—	—
stumpf	a	10	2	0.70	0.70	0.45	0.44
synchronisieren	v	20	2	0.90	0.90	0.81	0.81
Terroristin	n	10	1	1.00	1.00	—	—
thronen	v	20	1	0.30	0.30	0.00	-0.50
turbulent	a	10	1	1.00	1.00	—	—
Übereinstimmung	n	10	2	0.70	0.90	0.52	0.88
unbehindert	a	10	1	0.90	0.90	0.00	0.00
unbesetzt	a	10	1	0.30	0.30	-0.21	-0.46
ungesetzlich	a	10	1	1.00	1.00	—	—
unkalkulierbar	a	10	1	0.00	0.00	0.00	-0.90
unklar	a	10	1	1.00	1.00	—	—
unmäßig	a	10+	2	0.20	0.67	-0.01	0.07
unvermittelt	a	10	1	0.50	0.50	0.00	-0.27
ursprünglich	a	10	1	0.90	0.90	0.00	0.00
Varietät	n	10	4	0.90	0.97	0.85	0.97
Verbannung	n	10	1	1.00	1.00	—	—
verblenden	v	20	2	0.80	0.80	0.65	0.65

LEMMA	POS	#	δ	%	DICE	κ	α
verdecken	v	20	1	0.70	0.70	0.18	0.09
Verfilmung	n	10	1	1.00	1.00	—	—
vergesslich	a	10	1	0.90	0.90	0.00	0.00
verlängern	v	20+	4	0.90	0.97	0.75	0.95
verletzend	a	10	1	1.00	1.00	—	—
Versuchung	n	10	1	1.00	1.00	—	—
vertauschen	v	20	1	0.95	0.95	0.77	0.78
verzaubern	v	20+	2	0.85	0.95	0.69	0.93
wackelig	a	10	1	1.00	1.00	—	—
wegschaffen	v	20+	2	0.30	0.53	-0.17	-0.36
zusammenkommen	v	20+	2	1.00	1.00	1.00	1.00
Zustellung	n	10	1	1.00	1.00	—	—

BIBLIOGRAPHY

- Agirre, Eneko and Philip Edmonds, eds. (2007). *Word Sense Disambiguation: Algorithms and Applications*. Vol. 33 of Text, Speech, and Language Technology. Springer. ISBN: 978-1-4020-6870-6 (cit. on pp. 2, 111).
- Agirre, Eneko and Oier Lopez de Lacalle (2003). Clustering WordNet Word Senses. In: *International Conference Recent Advances in Natural Language Processing: Proceedings [RANLP]*. Ed. by Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov. Sept. 2003, pp. 11–18. ISBN: 978-954-90906-6-6 (cit. on pp. 62, 67).
- Agirre, Eneko, Oier López de Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers (2010). Semeval-2010 Task 17: All-words Word Sense Disambiguation on a Specific Domain. In: *SemEval 2010: 5th International Workshop on Semantic Evaluation: Proceedings of the Workshop*. Ed. by Katrin Erk and Carlo Strapparava. Stroudsburg, PA: Association for Computational Linguistics, July 2010, pp. 75–80. ISBN: 978-1-932432-70-1 (cit. on p. 32).
- Agirre, Eneko, Lluís Màrquez, and Richard Wicentowski, eds. (2007). *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Stroudsburg, PA: Association for Computational Linguistics, June 2007 (cit. on p. 31).
- Agirre, Eneko and David Martinez (2001). Knowledge Sources for Word Sense Disambiguation. In: *Text, Speech and Dialogue: 4th International Conference, TSD 2011*. Ed. by Václav Matoušek, Pavel Mautner, Roman Mouček, and Karel Taušer. Vol. 2166 of Lecture Notes in Computer Science. Berlin: Springer, pp. 1–10. ISSN: 0302-9743. ISBN: 978-3-540-42557-1. DOI: 10.1007/3-540-44805-5_1 (cit. on p. 14).
- Agirre, Eneko and David Martínez (2001). Learning Class-to-class Selectional Preferences. In: *Proceedings of CONLL-2001*. Ed. by Walter Daelemans and Rémi Zajac. New Brunswick, NJ: Association for Computational Linguistics, May 2001, pp. 15–22 (cit. on p. 19).
- Agirre, Eneko, David Martínez, Oier López de Lacalle, and Aitor Soroa (2006). Evaluating and Optimizing the Parameters of an Unsupervised Graph-based wsd Algorithm. In: *TextGraphs: Graph-based Algorithms for Natural Language Processing: Proceedings of the Workshop*. Ed. by Dragomir Radev and Rada Mihalcea. Stroudsburg, PA: Association for Computational Linguistics, pp. 89–96 (cit. on p. 40).
- Agirre, Eneko and German Rigau (1996). Word Sense Disambiguation Using Conceptual Density. In: *COLING-96: The 16th International Conference on Computational Linguistics: Proceedings*. Vol. 1. Aug. 1996, pp. 16–22 (cit. on p. 19).
- Agirre, Eneko and Aitor Soroa (2007). SemEval-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 7–12 (cit. on p. 31).
- (2009). Personalizing PageRank for Word Sense Disambiguation. In: *Proceedings of the 12th Conference of the European Chapter of the Association*

- for *Computational Linguistics* [EACL]. Stroudsburg, PA: Association for Computational Linguistics, Mar. 2009, pp. 33–41 (cit. on pp. 101, 106).
- Agirre, Eneko and Mark Stevenson (2007). Knowledge Sources for wsd. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Technology. Springer. Chap. 8, pp. 217–251. ISBN: 978-1-4020-6870-6 (cit. on p. 14).
- Anaya-Sánchez, Henry, Aurora Pons-Porrata, and Rafael Berlanga-Llavori (2007). tKB-UO: Using Sense Clustering for wsd. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 322–325 (cit. on p. 49).
- Artstein, Ron and Massimo Poesio (2008). Inter-coder Agreement for Computational Linguistics. In: *Computational Linguistics* 34(4) (Dec. 2008), pp. 555–596. ISSN: 0891-2017. DOI: 10.1162/coli.07-034-R2 (cit. on pp. 25, 26).
- Atkins, B. T. Sue (1992). Tools for Computer-aided Corpus Lexicography: The Hector Project. In: *Papers in Computational Lexicography: COMPLEX '92*. Ed. by Ferenc Kiefer, Gábor Kiss, and Júlia Pajzs. Budapest: Linguistics Institute, Hungarian Academy of Sciences, pp. 1–59. ISBN: 978-963-8461-67-4 (cit. on p. 30).
- Attardo, Salvatore (1994). *Linguistic Theories of Humor*. Berlin: Mouton de Gruyter. ISBN: 978-3-11-014255-6. DOI: 10.1515/9783110219029 (cit. on pp. 82, 88, 90).
- Banerjee, Satanjeev and Ted Pedersen (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002*. Ed. by Alexander Gelbukh. Vol. 2276 of Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, Feb. 2002, pp. 136–145. ISSN: 0302-9743. ISBN: 978-3-540-43219-7. DOI: 10.1007/3-540-45715-1_11 (cit. on pp. 38, 72).
- Bär, Daniel, Chris Biemann, Iryna Gurevych, and Torsten Zesch (2012). UKP: Computing Semantic Textual Similarity by Combining Multiple Content Similarity Measures. In: **SEM 2012: The First Joint Conference on Lexical and Computational Semantics: Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Ed. by Suresh Manandhar and Deniz Yuret. Stroudsburg, PA: Association for Computational Linguistics, June 2012, pp. 435–440. ISBN: 978-1-937284-22-0 (cit. on p. 41).
- Bär, Daniel, Torsten Zesch, and Iryna Gurevych (2013). DKPro Similarity: An Open-source Framework for Text Similarity. In: *ACL 2013: 51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: System Demonstrations*. Stroudsburg, PA: Association for Computational Linguistics, Aug. 2013, pp. 121–126 (cit. on p. 103).
- Bar-Hillel, Yehoshua (1960). Automatic Translation of Languages. In: *Advances in Computers*. Ed. by Franz Alt, A. Donald Booth, and R. E. Meagher. New York, NY: Academic Press (cit. on p. 1).
- Baroni, Marco, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-crawled Corpora. In: *Language Resources and Evaluation* 43(3) (Sept. 2009), pp. 209–226. ISSN: 1574-020X. DOI: 10.1007/s10579-009-9081-4 (cit. on pp. 15, 112, 116).

- Basile, Pierpaolo, Annalina Caputo, and Giovanni Semeraro (2014). An Enhanced Lesk Word Sense Disambiguation Algorithm Through a Distributional Semantic Model. In: *The 25th International Conference on Computational Linguistics: Proceedings of COLING 2014: Technical Papers*. Aug. 2014, pp. 1591–1600. ISBN: 978-1-941643-26-6 (cit. on p. 55).
- Bekinschtein, Tristan A., Matthew H. Davis, Jennifer M. Rodd, and Adrian M. Owen (2011). Why Clowns Taste Funny: The Relationship Between Humor and Semantic Ambiguity. In: *The Journal of Neuroscience* 31(26) (June 2011), pp. 9665–9671. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5058-10.2011 (cit. on p. 78).
- Bell, Nancy D., Scott Crossley, and Christian F. Hempelmann (2011). Wordplay in Church Marquees. In: *Humor: International Journal of Humor Research* 24(2) (Apr. 2011), pp. 187–202. ISSN: 0933-1719. DOI: 10.1515/HUMR.2011.012 (cit. on pp. 78, 81).
- Bhagwani, Sumit, Shrutiranjana Satapathy, and Harish Karnick (2013). Merging Word Senses. In: *Graph-based Methods for Natural Language Processing: Proceedings of the Workshop [TextGraphs]*. Stroudsburg, PA: Association for Computational Linguistics, Oct. 2013, pp. 11–19. ISBN: 978-1-937284-97-8 (cit. on pp. 62, 69).
- Biemann, Chris (2013). Creating a System for Lexical Substitutions from Scratch Using Crowdsourcing. In: *Language Resources and Evaluation* 47(1) (Mar. 2013), pp. 97–122. ISSN: 1574-020X. DOI: 10.1007/s10579-012-9180-5. (cit. on pp. 40, 106, 111, 117).
- Biemann, Chris, Gerhard Heyer, Uwe Quasthoff, and Matthias Richter (2007). The Leipzig Corpora Collection: Monolingual Corpora of Standard Size. In: *Proceedings of the Corpus Linguistics Conference: CL2007*. Ed. by Matthew Davies, Paul Rayson, Susan Hunston, and Pernilla Danielsson. July 2007. ISSN: 1747-9398 (cit. on p. 44).
- Biemann, Chris and Martin Riedl (2013). Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. In: *Journal of Language Modelling* 1(1), pp. 55–95. ISSN: 2299-856X. DOI: 10.15398/jlm.v1i1.60 (cit. on pp. 8, 44).
- Binsted, Kim and Graeme Ritchie (1994). An Implemented Model of Punning Riddles. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence: AAAI-94*. Palo Alto, CA: AAAI Press, pp. 633–638. ISBN: 978-0-262-51078-3 (cit. on p. 75).
- (1997). Computational Rules for Generating Punning Riddles. In: *Humor: International Journal of Humor Research* 10(1) (Jan. 1997), pp. 25–76. ISSN: 0933-1719. DOI: 10.1515/humr.1997.10.1.25 (cit. on p. 75).
- Bird, Steven (2006). NLTK: The Natural Language Toolkit. In: *COLING-ACL 2006: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Interactive Presentation Sessions*. Stroudsburg, PA: Association for Computational Linguistics, July 2006, pp. 69–72. DOI: 10.3115/1225403.1225421 (cit. on p. 103).
- Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). Latent Dirichlet Allocation. In: *Journal of Machine Learning Research* 3 (Jan. 2003), pp. 993–1022. ISSN: 1532-4435 (cit. on pp. 49, 56).
- Booth, Pat F. (2001). *Indexing: The Manual of Good Practice*. Munich: K. G. Saur. ISBN: 978-3-598-11536-3 (cit. on p. 66).

- Brants, Thorsten and Alex Franz (2006). *Web 1T 5-gram Version 1*. Philadelphia, PA: Linguistic Data Consortium, Sept. 2006. ISBN: 978-1-58563-397-5 (cit. on p. 16).
- Brin, Sergey and Lawrence Page (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. In: *Computer Networks and ISDN Systems* 30(1-7) (Apr. 1998), pp. 107-117. ISSN: 0169-7552. DOI: 10.1016/S0169-7552(98)00110-X (cit. on p. 20).
- Briscoe, Ted (1991). Lexical Issues in Natural Language Processing. In: *Natural Language and Speech: Symposium Proceedings*. Ed. by Ewan Klein and Frank Veltman. ESPRIT Basic Research Series. Berlin/Heidelberg: Springer, Nov. 1991, pp. 39-68. ISBN: 978-3-642-77191-0. DOI: 10.1007/978-3-642-77189-7_4 (cit. on p. 57).
- Broscheit, Samuel, Anette Frank, Dominic Jehle, Simone Paolo Ponzetto, Danny Rehl, Anja Summa, Klaus Suttner, and Saskia Vola (2010). Rapid Bootstrapping of Word Sense Disambiguation Resources for German. In: *Semantic Approaches in Natural Language Processing: Proceedings of the Conference on Natural Language Processing 2010 [KONVENS]*. Ed. by Manfred Pinkal, Ines Rehbein, Sabine Schulte im Walde, and Angelika Storrer. universaar, pp. 19-27. ISBN: 978-3-86223-004-4 (cit. on p. 112).
- Bucaria, Chiara (2004). Lexical and Syntactic Ambiguity as a Source of Humor: The Case of Newspaper Headlines. In: *Humor: International Journal of Humor Research* 17(3) (June 2004), pp. 279-309. ISSN: 0933-1719. DOI: 10.1515/humr.2004.013 (cit. on p. 81).
- Budanitsky, Alexander and Graeme Hirst (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. In: *Computational Linguistics* 32(1) (Mar. 2006), pp. 13-47. ISSN: 0891-2017. DOI: 10.1162/coli.2006.32.1.13 (cit. on p. 20).
- Buitelaar, Paul (2000). Reducing Lexical Semantic Complexity with Systematic Polysemous Classes and Underspecification. In: *NAACL-ANLP Workshop: Syntactic and Semantic Complexity in Natural Language Processing Systems*. Ed. by Amit Bagga, James Pustejovsky, and Wlodek Zadrozny. New Brunswick, NJ: Association for Computational Linguistics, Apr. 2000, pp. 14-19 (cit. on pp. 3, 61).
- Burnard, Lou, ed. (2007). *Reference Guide for the British National Corpus (XML Edition)*. British National Corpus Consortium. Feb. 2007 (cit. on p. 15).
- Cai, Jun Fu, Wee Sun Lee, and Yee Whye Teh (2007). NUS-ML: Improving Word Sense Disambiguation Using Topic Features. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 249-252 (cit. on p. 40).
- Carpuat, Marine and Dekai Wu (2007). Improving Statistical Machine Translation Using Word Sense Disambiguation. In: *EMNLP-CONLL: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 61-72 (cit. on p. 23).
- Chan, Yee Seng, Hwee Tou Ng, and David Chiang (2007). Word Sense Disambiguation Improves Statistical Machine Translation. In: *ACL 2007: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 33-40 (cit. on p. 23).

- Chen, Jen Nan and Jason S. Chang (1998). Topical Clustering of MRD Senses Based on Information Retrieval Techniques. In: *Computational Linguistics* 24(1), pp. 61–95. ISSN: 0891-2017 (cit. on p. 61).
- Chen, Xinxiong, Zhiyuan Liu, and Maosong Sun (2014). A Unified Model for Word Sense Representation and Disambiguation. In: *The 2014 Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference [EMNLP]*. Stroudsburg, PA: Association for Computational Linguistics, Oct. 2014, pp. 1025–1035. ISBN: 978-1-937284-96-1 (cit. on pp. 40, 49, 55).
- Chklovski, Timothy and Rada Mihalcea (2003). Exploiting Agreement and Disagreement of Human Annotators for Word Sense Disambiguation. In: *International Conference Recent Advances in Natural Language Processing: Proceedings [RANLP]*. Ed. by Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov. Sept. 2003. ISBN: 978-954-90906-6-6 (cit. on p. 62).
- Chklovski, Timothy, Rada Mihalcea, Ted Pedersen, and Amruta Purandare (2004). The SENSEVAL-3 Multilingual English–Hindi Lexical Sample Task. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Philip Edmonds. New Brunswick, NJ: Association for Computational Linguistics, July 2004, pp. 5–8 (cit. on p. 23).
- Cholakov, Kostadin, Chris Biemann, Judith Eckle-Kohler, and Iryna Gurevych (2014). Lexical Substitution Dataset for German. In: *LREC 2014, Ninth International Conference on Language Resources and Evaluation*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis. European Language Resources Association, May 2014, pp. 2524–2531. ISBN: 978-2-9517408-8-4 (cit. on pp. 114, 116, 118, 122, 124).
- Clear, Jeremy H. (1993). The British National Corpus. In: *The Digital Word: Text-based Computing in the Humanities*. Ed. by George P. Landow and Paul Delany. Technical Communication and Information Systems. Cambridge, MA: MIT Press, pp. 163–187. ISBN: 978-0-262-12176-7 (cit. on p. 66).
- Cohen, Jacob (1960). A Coefficient of Agreement for Nominal Scales. In: *Educational and Psychological Measurement* 20(1) (Apr. 1960), pp. 37–46. ISSN: 0013-1644. DOI: 10.1177/001316446002000104 (cit. on p. 26).
- Cohn, Trevor (2003). Performance Metrics for Word Sense Disambiguation. In: *Proceedings of the Australasian Language Technology Summer School (ALTSS) and Australasian Language Technology Workshop (ALTW) 2003*. Ed. by Catherine Bow and Baden Hughes. Dec. 2003, pp. 86–93. ISBN: 978-0-9751687-1-4 (cit. on p. 95).
- Collin, Peter H. (1980). *Harrap's Easy English Dictionary*. London: George G. Harrap & Co. Ltd (cit. on p. 3).
- Crosbie, John S. (1977). *Crosbie's Dictionary of Puns*. New York, NY: Harmony. ISBN: 978-0-517-53124-2 (cit. on p. 81).
- Culler, Jonathan D., ed. (1988). *On Puns: The Foundation of Letters*. Oxford: Basil Blackwell. ISBN: 978-0-631-15893-6 (cit. on p. 78).
- Curran, James R. and Marc Moens (2002). Improvements in Automatic Thesaurus Extraction. In: *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*. New Brunswick, NJ: Association for Computational Linguistics, July 2002, pp. 59–66. DOI: 10.3115/1118627.1118635 (cit. on p. 40).

- Czuczor, Gergely and János Fogarasi, eds. (1874). *A magyar nyelv szótára*. Vol. 6. Budapest: Athenaeum Irodalmi és Nyomdai R.-Társulat (cit. on p. 13).
- Dandala, Bharath, Rada Mihalcea, and Razvan Bunescu (2013). Word Sense Disambiguation Using Wikipedia. In: *The People's Web Meets NLP: Collaboratively Constructed Language Resources*. Ed. by Iryna Gurevych and Jungi Kim. Theory and Applications of Natural Language Processing. Berlin/Heidelberg: Springer, pp. 241–262. ISSN: 2192-032X. ISBN: 978-3-642-35084-9. DOI: 10.1007/978-3-642-35085-6_9 (cit. on p. 60).
- Daudé, Jordi, Lluís Padró, and German Rigau (2003). Validation and Tuning of WordNet Mapping Techniques. In: *International Conference Recent Advances in Natural Language Processing: Proceedings [RANLP]*. Ed. by Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov. Sept. 2003, pp. 117–123. ISBN: 978-954-90906-6-6 (cit. on pp. 70, 106).
- Daxenberger, Johannes, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch (2014). DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations [ACL]*. Stroudsburg, PA: Association for Computational Linguistics, June 2014, pp. 61–66. ISBN: 978-1-941643-00-6 (cit. on p. 132).
- de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In: *5th Edition of the International Conference on Language Resources and Evaluation*. European Language Resources Association, May 2006, pp. 449–454. ISBN: 978-2-9517408-2-2 (cit. on p. 44).
- de Melo, Gerard and Gerhard Weikum (2009). Towards a Universal Wordnet by Learning from Combined Evidence. In: *CIKM '09: Proceedings of the 18th ACM Conference on Information and Knowledge Management*. New York, NY: ACM, pp. 513–522. ISBN: 978-1-60558-512-3. DOI: 10.1145/1645953.1646020 (cit. on p. 60).
- Decadt, Bart, Véronique Hoste, Walter Daelemans, and Antal van den Bosch (2004). GAMBL, Genetic Algorithm Optimization of Memory-based wsd. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Philip Edmonds. New Brunswick, NJ: Association for Computational Linguistics, July 2004, pp. 108–112 (cit. on p. 70).
- Del Corro, Luciano (2015). *Methods for Open Information Extraction and Sense Disambiguation on Natural Language Text*. Dr.-Ing. thesis. Universität des Saarlandes (cit. on p. 109).
- Del Corro, Luciano, Rainer Gemulla, and Gerhard Weikum (2014). Werdy: Recognition and Disambiguation of Verbs and Verb Phrases with Syntactic and Semantic Pruning. In: *The 2014 Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference [EMNLP]*. Stroudsburg, PA: Association for Computational Linguistics, Oct. 2014, pp. 374–385. ISBN: 978-1-937284-96-1 (cit. on p. 109).
- Delabastita, Dirk (1997a). Introduction. In: *Traductio: Essays on Punning and Translation*. Ed. by Dirk Delabastita. Manchester: St. Jerome, pp. 1–22. ISBN: 978-1-900650-06-9 (cit. on pp. 75, 78).
- ed. (1997b). *Traductio: Essays on Punning and Translation*. Manchester: St. Jerome. ISBN: 978-1-900650-06-9 (cit. on p. 78).

- Dice, Lee R. (1945). Measures of the Amount of Ecologic Association Between Species. In: *Ecology* 26(3), pp. 297–302. ISSN: 0012-9658. DOI: 10.2307/1932409 (cit. on p. 25).
- Dijkstra, Edsger W. (1959). A Note on Two Problems in Connexion with Graphs. In: *Numerische Mathematik* 1(1) (Dec. 1959), pp. 269–271. ISSN: 0029-599X. DOI: 10.1007/BF01386390 (cit. on p. 61).
- Dolan, William B. (1994). Word Sense Ambiguation: Clustering Related Senses. In: *COLING 94: The 15th International Conference on Computational Linguistics: Proceedings*. Vol. 2. Aug. 1994, pp. 712–716 (cit. on p. 61).
- Dunning, Ted (1993). Accurate methods for the statistics of surprise and coincidence. In: *Computational Linguistics* 19(1) (Mar. 1993), pp. 61–74. ISSN: 0891-2017 (cit. on p. 44).
- Eckart de Castilho, Richard (2014). *Natural Language Processing: Integration of Automatic and Manual Analysis*. Dr.-Ing. thesis. Technische Universität Darmstadt, Feb. 2014 (cit. on pp. 99, 101).
- Eckart de Castilho, Richard and Iryna Gurevych (2011). A Lightweight Framework for Reproducible Parameter Sweeping in Information Retrieval. In: *Proceedings of the 2011 Workshop on Data Infrastructures for Supporting Information Retrieval Evaluation (DESIRE '11)*. Ed. by Maristella Agosti, Nicola Ferro, and Costantino Thanos. New York, NY: ACM, Oct. 2011, pp. 7–10. ISBN: 978-1-4503-0952-3. DOI: 10.1145/2064227.2064248 (cit. on p. 103).
- (2014). A Broad-coverage Collection of Portable NLP Components for Building Shareable Analysis Pipelines. In: *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT [OIAF4HLT]*. Aug. 2014, pp. 1–11 (cit. on p. 103).
- Edmonds, Philip (2005). Lexical Disambiguation. In: *Encyclopedia of Language and Linguistics*. Ed. by Keith Brown. 2nd edition. Oxford: Elsevier Science, pp. 607–623. ISBN: 978-0-08-044299-0 (cit. on pp. 2, 30).
- Edmonds, Philip and Adam Kilgarriff (2002a). Introduction to the Special Issue on Evaluating Word Sense Disambiguation Systems. In: *Natural Language Engineering* 8(4), pp. 279–291. ISSN: 1351-3249 (cit. on pp. 23, 30).
- eds. (2002b). *Natural Language Engineering* 8(4) (Dec. 2002): *Special Issue on Evaluating Word Sense Disambiguation Systems*. ISSN: 1351-3249 (cit. on p. 2).
- Edmundson, Harold P. (1967). Axiomatic Characterization of Synonymy and Antonymy. In: *2ème Conférence internationale sur le traitement automatique des langues [COLING]* (cit. on p. 63).
- Erbs, Nicolai, Eneko Agirre, Aitor Soroa, Ander Barrena, Ugaitz Etxebarria, Iryna Gurevych, and Torsten Zesch (2012). UKP-UBC Entity Linking at TAC-KBP. In: *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*. Gaithersburg, MD: National Institute of Standards and Technology, Nov. 2012 (cit. on p. 109).
- Erk, Katrin and Carlo Strapparava, eds. (2010). *SemEval 2010: 5th International Workshop on Semantic Evaluation: Proceedings of the Workshop*. Stroudsburg, PA: Association for Computational Linguistics, July 2010. ISBN: 978-1-932432-70-1 (cit. on p. 32).
- Fabre, Cécile, Nabil Hathout, Lydia-Mai Ho-Dac, François Morlane-Hondère, Philippe Muller, Franck Sajous, Ludovic Tanguy, and Tim Van de Cruys (2014). Présentation de l’atelier SemDis 2014: sémantique distributionnelle pour la substitution lexicale et l’exploration de corpus spécialisés. In: *Actes des Ateliers TALN 2014*. Ed. by Brigitte Bigi. Paris: Association

- pour le traitement automatique des langues, July 2014, pp. 196–205. ISBN: 978-2-9518233-6-5 (cit. on p. 115).
- Fellbaum, Christiane, ed. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press. ISBN: 978-0-262-06197-1 (cit. on pp. 15, 16, 59, 105).
- Ferrucci, David and Adam Lally (2004). UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. In: *Natural Language Engineering* 10(3&4) (Sept. 2004), pp. 327–348. ISSN: 1351-3249. DOI: 10.1017/S1351324904003523 (cit. on pp. 100, 102).
- Firth, J. R. (1957). A Synopsis of Linguistic Theory, 1930–1955. In: *Studies in Linguistic Analysis*. Oxford: Basil Blackwell, pp. 1–32 (cit. on p. 39).
- Fort, Karën, Adeline Nazarenko, and Sophie Rosset (2012). Modeling the Complexity of Manual Annotation Tasks: A Grid of Analysis. In: *Proceedings of COLING 2012: Technical Papers*. Ed. by Martin Kay and Christian Boitet. Dec. 2012, pp. 895–910 (cit. on p. 35).
- Gale, William (1992). Estimating Upper and Lower Bounds on the Performance of Word-sense Disambiguation Programs. In: *30th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. Morristown, NJ: Association for Computational Linguistics, June 1992, pp. 249–256. DOI: 10.3115/981967.981999 (cit. on pp. 29, 91).
- Gale, William A., Kenneth W. Church, and David Yarowsky (1992). One Sense per Discourse. In: *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York [HLT]*. San Mateo, CA: Morgan Kaufmann, Feb. 1992, pp. 233–237. ISBN: 978-1-55860-272-4 (cit. on p. 20).
- Garoufi, Konstantina, Torsten Zesch, and Iryna Gurevych (2008). Representational Interoperability of Linguistic and Collaborative Knowledge Bases. In: *Proceedings of the KONVENS Workshop on Lexical-semantic and Ontological Resources – Maintenance, Representation, and Standards*. Oct. 2008 (cit. on pp. 60, 103).
- Gliozzo, Alfio, Claudio Giuliano, and Carlo Strapparava (2005). Domain Kernels for Word Sense Disambiguation. In: *43rd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. New Brunswick, NJ: Association for Computational Linguistics, June 2005, pp. 403–410. DOI: 10.3115/1219840.1219890 (cit. on p. 40).
- Gonzalo, Julio and Felisa Verdejo (2007). Automatic Acquisition of Lexical Information and Examples. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Technology. Springer. Chap. 9, pp. 253–274. ISBN: 978-1-4020-6870-6 (cit. on p. 22).
- Goyal, Amit, Hal Daumé III, and Graham Cormode (2012). Sketch Algorithms for Estimating Point Queries in NLP. In: *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning: Proceedings of the Conference [EMNLP-CONLL]*. Stroudsburg, PA: Association for Computational Linguistics, July 2012, pp. 1093–1103. ISBN: 978-1-937284-43-5 (cit. on p. 44).
- Grefenstette, Gregory (1994). *Explorations in Automatic Thesaurus Discovery*. Vol. 278 of The Springer International Series in Engineering and Computer Science. New York, NY: Springer US. ISSN: 0893-3405. ISBN: 978-0-7923-9468-6. DOI: 10.1007/978-1-4615-2710-7 (cit. on p. 40).
- Gurevych, Iryna, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth (2012). UBY – A Large-

- scale Unified Lexical-semantic Resource. In: *13th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Conference [EACL]*. Stroudsburg, PA: Association for Computational Linguistics, Apr. 2012, pp. 580–590. ISBN: 978-1-937284-19-0 (cit. on pp. 56, 60, 103, 106, 119).
- Gurevych, Iryna, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch (2007). Darmstadt Knowledge Processing Repository Based on UIMA. In: *UIMA Workshop at the GLDV 2007*. Apr. 2007 (cit. on p. 103).
- Gurevych, Iryna and Elisabeth Wolf (2010). Expert-built and Collaboratively Constructed Lexical Semantic Resources. In: *Language and Linguistics Compass* 4(11) (Nov. 2010), pp. 1074–1090. ISSN: 1749-818X. DOI: 10.1111/j.1749-818X.2010.00251.x (cit. on p. 59).
- Hamp, Birgit and Helmut Feldweg (1997). GermaNet – A Lexical-semantic Net for German. In: *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications: ACL/EACL-97 Workshop Proceedings*. Ed. by Piek Vossen, Geert Adriaens, Nicolleta Calzolari, Antonio Sanfilippo, and Yorick Wilks. Somerset, NJ: Association for Computational Linguistics, July 1997, pp. 9–15 (cit. on pp. 16, 119).
- Hanks, Patrick (2000). Do Word Meanings Exist? In: *Computers and the Humanities* 34(1&2) (Apr. 2000), pp. 205–215. ISSN: 0010-4817. DOI: 10.1023/A:1002471322828 (cit. on p. 3).
- Hartmann, Silvana and Iryna Gurevych (2013). FrameNet on the Way to Babel: Creating a Bilingual FrameNet Using Wiktionary as Interlingual Connection. In: *ACL 2013: 51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: Volume 1: Long Papers*. Stroudsburg, PA: Association for Computational Linguistics, Aug. 2013, pp. 1363–1373. ISBN: 978-1-937284-50-3 (cit. on p. 60).
- Hearst, Marti A. (1998). Automated Discovery of WordNet Relations. In: *WordNet: An Electronic Lexical Database*. Ed. by Christiane Fellbaum. Cambridge, MA: MIT Press, pp. 131–152. ISBN: 978-0-262-06197-1 (cit. on p. 57).
- Heinzerling, Benjamin, Alex Judea, and Michael Strube (2015). HITS at TAC KBP 2015: Entity Discovering and Linking, and Event Nugget Detection. In: *Proceedings of the Eighth Text Analysis Conference (TAC 2015)*. Gaithersburg, MD: National Institute of Standards and Technology, Feb. 2015 (cit. on p. 109).
- Hempelmann, Christian F. (2003a). *Paronomasic Puns: Target Recoverability Towards Automatic Generation*. Ph.D. thesis. West Lafayette, IN: Purdue University, Aug. 2003 (cit. on pp. 75, 83, 132).
- (2003b). YPS – The Ynperfect Pun Selector for Computational Humor. In: *Proceedings of the CHI 2003 Workshop on Humor Modeling in the Interface*. Apr. 2003 (cit. on p. 75).
- (2008). Computational Humor: Beyond the Pun? In: *The Primer of Humor Research*. Ed. by Victor Raskin. Vol. 8 of Humor Research. Berlin: Mouton de Gruyter, pp. 333–360. ISSN: 1861-4116. ISBN: 978-3-11-018616-1. DOI: 10.1515/9783110198492.333 (cit. on p. 76).
- Henrich, Verena (2015). *Word Sense Disambiguation with GermaNet: Semi-automatic Enhancement and Empirical Results*. Ph.D. thesis. Eberhard Karls Universität Tübingen, Jan. 2015. DOI: 10.15496/publikation-4706 (cit. on pp. 20, 112, 114, 120).
- Henrich, Verena and Erhard Hinrichs (2010). GernEdiT – The GermaNet Editing Tool. In: *LREC 2010, Seventh International Conference on Language Re-*

- sources and Evaluation*. Ed. by Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias. European Language Resources Association, May 2010, pp. 2228–2235. ISBN: 978-2-9517408-6-0 (cit. on pp. 16, 119).
- Henrich, Verena and Erhard Hinrichs (2013). Extending the TüBa-D/z Treebank with GermaNet Sense Annotation. In: *Language Processing and Knowledge in the Web: 25th International Conference, GSCL 2013*. Ed. by Iryna Gurevych, Chris Biemann, and Torsten Zesch. Vol. 8105 of Lecture Notes in Artificial Intelligence. Berlin/Heidelberg: Springer, pp. 89–96. ISSN: 0302-9743. ISBN: 978-3-642-40721-5. DOI: 10.1007/978-3-642-40722-2_9 (cit. on pp. 114, 120, 121).
- (2014). Consistency of Manual Sense Annotation and Integration into the TüBa-D/z Treebank. In: *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*. Ed. by Verena Henrich, Erhard Hinrichs, Daniël de Kok, Petya Osenova, and Adam Przepiórkowski. Tübingen: Department of Linguistics (SfS), University of Tübingen, Dec. 2014, pp. 62–74. ISBN: 978-3-9809183-9-8 (cit. on pp. 114, 121).
- Henrich, Verena, Erhard Hinrichs, and Klaus Suttner (2012). Automatically Linking GermaNet to Wikipedia for Harvesting Corpus Examples for GermaNet Senses. In: *Journal for Language Technology and Computational Linguistics* 27(1), pp. 1–19. ISSN: 2190-6858 (cit. on p. 112).
- Henrich, Verena, Erhard Hinrichs, and Tatiana Vodolazova (2012). webCAGE – A Web-harvested Corpus Annotated with GermaNet Senses. In: *13th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Conference [EACL]*. Stroudsburg, PA: Association for Computational Linguistics, Apr. 2012, pp. 387–396. ISBN: 978-1-937284-19-0 (cit. on pp. 43, 100, 112).
- (2014). Aligning GermaNet Senses with Wiktionary Sense Definitions. In: *Human Language Technology Challenges for Computer Science and Linguistics: 5th Language and Technology Conference, LTC 2011*. Ed. by Zygmunt Vetulani and Joseph Mariani. Vol. 8387 of Lecture Notes in Artificial Intelligence. Springer International Publishing, pp. 329–342. ISSN: 0302-9743. ISBN: 978-3-319-08957-7. DOI: 10.1007/978-3-319-08958-4_27 (cit. on p. 119).
- Hindle, Donald (1990). Noun Classification from Predicate–Argument Structures. In: *28th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*. Morristown, NJ: Association for Computational Linguistics, June 1990, pp. 268–275 (cit. on p. 40).
- Hirschberg, Julia, ed. (1998). *Computational Linguistics* 24(1) (Mar. 1998): *Special Issue on Word Sense Disambiguation*. ISSN: 0891-2017 (cit. on p. 2).
- Hirst, Graeme (1987). *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge: Cambridge University Press. ISBN: 978-0-521-32203-4 (cit. on p. 1).
- Hirst, Graeme and David St-Onge (1998). Lexical Chains as Representations of Context in the Detection and Correction of Malapropisms. In: *WordNet: An Electronic Lexical Database*. Ed. by Christiane Fellbaum. Cambridge, MA: MIT Press, pp. 305–332. ISBN: 978-0-262-06197-1 (cit. on p. 19).
- Hoffart, Johannes, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum (2011). Robust Disambiguation of Named Entities in Text. In: *Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference [EMNLP]*. Stroudsburg, PA: Association for Computa-

- tional Linguistics, July 2011, pp. 782–792. ISBN: 978-1-937284-11-4 (cit. on p. 105).
- Hong, Bryan Anthony and Ethel Ong (2009). Automatically Extracting Word Relationships as Templates for Pun Generation. In: *Computational Approaches to Linguistic Creativity: Proceedings of the Workshop [CALC]*. Stroudsburg, PA: Association for Computational Linguistics, June 2009, pp. 24–31. ISBN: 978-1-932432-36-7 (cit. on pp. 75, 81).
- Hopcroft, John and Robert Tarjan (1973). Algorithm 447: Efficient Algorithms for Graph Manipulation. In: *Communications of the ACM* 16(6) (June 1973), pp. 372–378. ISSN: 0001-0782. DOI: 10.1145/362248.362272 (cit. on p. 64).
- Hoste, Véronique, Iris Hendrickx, Walter Daelemans, and Antal van den Bosch (2002). Parameter Optimization for Machine-learning of Word Sense Disambiguation. In: *Natural Language Engineering* 8(4) (Dec. 2002), pp. 311–325. ISSN: 1351-3249. DOI: 10.1017/S1351324902003005 (cit. on p. 21).
- Hovy, Eduard, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel (2006). OntoNotes: The 90% Solution. In: *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: Short Papers [HLT-NAACL]*. Ed. by Robert C. Moore, Jeff Bilmes, Jennifer Chu-Carroll, and Mark Sanderson. Stroudsburg, PA: Association for Computational Linguistics, June 2006, pp. 57–60 (cit. on pp. 32, 84, 111).
- Ide, Nancy (2006). Making Senses: Bootstrapping Sense-tagged Lists of Semantically-related Words. In: *Computational Linguistics and Intelligent Text Processing: 7th International Conference, CICLing 2006*. Ed. by Alexander Gelbukh. Vol. 3878 of Theoretical Computer Science and General Issues. Berlin/Heidelberg: Springer, Feb. 2006, pp. 13–27. ISBN: 978-3-540-32205-4. DOI: 10.1007/11671299_2 (cit. on p. 61).
- Ide, Nancy and Keith Suderman (2004). The American National Corpus First Release. In: *LREC 2004: Fourth International Conference on Language Resources and Evaluation*. Ed. by Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva. May 2004, pp. 1681–1684. ISBN: 978-2-9517408-1-5 (cit. on p. 15).
- Ide, Nancy and Jean Véronis (1998). Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. In: *Computational Linguistics* 24(1), pp. 1–40. ISSN: 0891-2017 (cit. on pp. 2, 23).
- Ide, Nancy and Yorick Wilks (2007). Making Sense About Sense. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Technology. Springer. Chap. 3, pp. 47–73. ISBN: 978-1-4020-6870-6 (cit. on pp. 3, 85).
- Iverson, Kenneth E. (1962). *A Programming Language*. New York, NY: Wiley. ISBN: 978-0-471-43014-8 (cit. on p. 11).
- Jaccard, Paul (1901). Distribution de la flore alpine dans le Bassin des Dranses et dans quelques régions voisines. In: *Bulletin de la Société vaudoise des sciences naturelles* 37(140), pp. 241–272. ISSN: 0037-9603. DOI: 10.5169/seals-266440 (cit. on p. 28).
- Jarmasz, Mario (2003). *Roget's Thesaurus as a Lexical Resource for Natural Language Processing*. Master's thesis. July 2003 (cit. on p. 15).
- Ji, Heng, Joel Nothman, Ben Hachey, and Radu Florian (2015). Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In: *Proceedings of*

- the Eighth Text Analysis Conference (TAC 2015)*. Gaithersburg, MD: National Institute of Standards and Technology, Feb. 2015 (cit. on p. 109).
- Jiang, Jay J. and David W. Conrath (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: *Proceedings of the ROCLING x (1997) International Conference: Research in Computational Linguistics*. Aug. 1997, pp. 19–33 (cit. on p. 19).
- Jorgensen, Julia C. (1990). The Psychological Reality of Word Senses. In: *Journal of Psycholinguistic Research* 19(3) (May 1990), pp. 167–190. ISSN: 0090-6905. DOI: 10.1007/BF01077415 (cit. on p. 3).
- Joshi, Salil, Mitesh M. Khapra, and Pushpak Bhattacharyya (2012). I Can Sense It: A comprehensive online system for wsd. In: *Proceedings of COLING 2012: Demonstration Papers*. Ed. by Martin Kay and Christian Boitet. Dec. 2012, pp. 247–254 (cit. on p. 101).
- Jurgens, David and Ioannis Klapaftis (2013). SemEval-2013 Task 13: Word Sense Induction for Graded and Non-graded Senses. In: **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics: Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Ed. by Suresh Manandhar and Deniz Yuret. Stroudsburg, PA: Association for Computational Linguistics, June 2013, pp. 290–299. ISBN: 978-1-937284-49-7 (cit. on p. 86).
- Kahusk, Neeme, Heili Orav, and Haldur Õim (2001). Sensiting Inflectionality: Estonian Task for SENSEVAL-2. In: *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*. Ed. by Judita Preiss and David Yarowsky. New Brunswick, NJ: Association for Computational Linguistics, July 2001, pp. 25–28 (cit. on p. 104).
- Kaplan, Nora and Teresa Lucas (2001). Comprensión del humorismo en inglés: Estudio de las estrategias de inferencia utilizadas por estudiantes avanzados de inglés como lengua extranjera en la interpretación de los retruécanos en historietas cómicas en lengua inglesa. In: *Anales de la Universidad Metropolitana* 1(2), pp. 245–258. ISSN: 1856-9811 (cit. on p. 81).
- Kawahara, Shigeto (2010). *Papers on Japanese Imperfect Puns*. Online collection of previously published journal and conference articles. <http://user.keio.ac.jp/~kawahara/pdf/punbook.pdf>. Accessed 17 June 2015. Jan. 2010 (cit. on p. 75).
- Keller, Stefan Daniel (2009). *The Development of Shakespeare's Rhetoric: A Study of Nine Plays*. Vol. 136 of Swiss Studies in English. Tübingen: Narr. ISSN: 0080-7214. ISBN: 978-3-7720-8324-2 (cit. on p. 77).
- Kilgarriff, Adam (1997). I Don't Believe in Word Senses. In: *Computers and the Humanities* 31(2) (Mar. 1997), pp. 91–113. ISSN: 0010-4817 (cit. on p. 3).
- (2001). English Lexical Sample Task Description. In: *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*. Ed. by Judita Preiss and David Yarowsky. New Brunswick, NJ: Association for Computational Linguistics, July 2001, pp. 17–20 (cit. on p. 46).
- (2010). A Detailed, Accurate, Extensive, Available English Lexical Database. In: *Human Language Technologies: the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstration Session [NAACL-HLT]*. Stroudsburg, PA: Association for Computational Linguistics, June 2010, pp. 21–24 (cit. on p. 100).
- Kilgarriff, Adam and Martha Palmer, eds. (2000). *Computers and the Humanities* 34(1&2) (Apr. 2000): *Special Issue on SENSEVAL*. ISSN: 0010-4817 (cit. on p. 2).

- Kilgarriff, Adam and Joseph Rosenzweig (2000). Framework and Results for English SENSEVAL. In: *Computers and the Humanities* 34(1&2) (Apr. 2000), pp. 15–48. ISSN: 0010-4817. DOI: 10.1023/A:1002693207386 (cit. on pp. 30, 37, 38, 71, 89).
- Kilgarriff, Adam and David Tugwell (2001). Word Sketch: Extraction and Display of Significant Collocations for Lexicography. In: *Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis and Exploitation*, pp. 32–38 (cit. on p. 40).
- Kipfer, Barbara A. (1984). Methods of Ordering Senses Within Entries. In: *LEXeter '83 Proceedings: Papers from the International Conference on Lexicography at Exeter*. Ed. by Reinhard R. K. Hartmann. Vol. 1. Lexicographica Series Maior: Supplementary Volumes to the International Annual for Lexicography. Tübingen: Max Niemeyer Verlag, pp. 101–108. ISSN: 0175-9264. ISBN: 3-484-30901-6 (cit. on p. 29).
- Kipper, Karin, Anna Korhonen, Neville Ryant, and Martha Palmer (2008). A Large-scale Classification of English Verbs. In: *Language Resources and Evaluation* 42(1) (Mar. 2008), pp. 21–40. ISSN: 1574-020X. DOI: 10.1007/s10579-007-9048-2 (cit. on p. 100).
- Knuth, Donald (1992). Two Notes on Notation. In: *American Mathematical Monthly* 99(5) (May 1992), pp. 403–422. ISSN: 0002-9890. DOI: 10.2307/2325085 (cit. on p. 11).
- Kohomban, Upali S. and Wee Sun Lee (2005). Learning Semantic Classes for Word Sense Disambiguation. In: *43rd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. New Brunswick, NJ: Association for Computational Linguistics, June 2005, pp. 34–41. DOI: 10.3115/1219840.1219845 (cit. on p. 62).
- Kremer, Gerhard, Katrin Erk, Sebastian Padó, and Stefan Thater (2014). What Substitutes Tell Us – Analysis of an “All-words” Lexical Substitution Corpus. In: *14th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Conference [EACL]*. Stroudsburg, PA: Association for Computational Linguistics, Apr. 2014, pp. 540–549. ISBN: 978-1-937284-78-7 (cit. on pp. 111, 117, 118, 121, 123–126).
- Krippendorff, Klaus (1980). *Content Analysis: An Introduction to Its Methodology*. Vol. 5 of The Sage COMMTEXT Series. Beverly Hills, CA: Sage Publications. ISBN: 0-8039-1497-0 (cit. on pp. 27, 86).
- (2004). Reliability in Content Analysis: Some Common Misconceptions and Recommendations. In: *Human Communication Research* 30(3) (July 2004), pp. 411–433. ISSN: 1468-2958. DOI: 10.1111/j.1468-2958.2004.tb00738.x (cit. on pp. 25, 26).
- Krovetz, Robert (1997). Homonymy and Polysemy in Information Retrieval. In: *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Conference [ACL-EACL]*. Association for Computational Linguistics, July 1997, pp. 72–79. DOI: 10.3115/976909.979627 (cit. on p. 2).
- Kurohashi, Sadao (2001). SENSEVAL-2 Japanese Translation Task. In: *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*. Ed. by Judita Preiss and David Yarowsky. New Brunswick, NJ: Association for Computational Linguistics, July 2001, pp. 37–40 (cit. on p. 23).
- Kwong, Oi Yee (2013). *New Perspectives on Computational and Cognitive Strategies for Word Sense Disambiguation*. SpringerBriefs in Speech Technology.

- New York, NY: Springer. ISSN: 2191-737X. ISBN: 978-1-4614-1319-6 (cit. on pp. 2, 3).
- Lagerwerf, Luuk (2002). Deliberate Ambiguity in Slogans: Recognition and Appreciation. In: *Document Design* 3(3), pp. 245–260. ISSN: 1388-8951. DOI: 10.1075/dd.3.3.07lag (cit. on p. 78).
- Lally, Adam, Karin Verspoor, and Eric Nyberg, eds. (2009). *Unstructured Information Management Architecture (UIMA) Version 1.0*. OASIS, Mar. 2009 (cit. on pp. 100, 102).
- Leacock, Claudia and Martin Chodorow (1998). Combining local context and WordNet similarity for word sense identification. In: *WordNet: An Electronic Lexical Database*. Ed. by Christiane Fellbaum. Cambridge, MA: MIT Press, pp. 265–283. ISBN: 978-0-262-06197-1 (cit. on p. 19).
- Lee, Yoong Keok and Hwee Tou Ng (2002). An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. In: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing [EMNLP]*. Ed. by Jan Hajič and Yuji Matsumoto. New Brunswick, NJ: Association for Computational Linguistics, July 2002, pp. 41–48. DOI: 10.3115/1118693.1118699 (cit. on p. 21).
- Lesk, Michael (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from a Ice Cream Cone. In: *SIGDOC '86: Proceedings of the 5th Annual International Conference on Systems Documentation*. Ed. by Virginia DeBuys. New York, NY: ACM Press, pp. 24–26. ISBN: 978-0-89791-224-2. DOI: 10.1145/318723.318728 (cit. on pp. 19, 36–38, 43, 80).
- Lessard, Greg, Michael Levison, and Chris Venour (2002). Cleverness Versus Funniness. In: *TWLT20: The April Fools' Day Workshop on Computation Humour: Proceedings of the Twentieth Twente Workshop on Language Technology*. Ed. by Oliviero Stock, Carlo Strapparava, and Anton Nijholt, pp. 137–145. ISSN: 0929-0672 (cit. on p. 81).
- Li, Linlin, Benjamin Roth, and Caroline Sporleder (2010). Topic Models for Word Sense Disambiguation and Token-based Idiom Detection. In: *48th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. Vol. 1. Stroudsburg, PA: Association for Computational Linguistics, July 2010, pp. 1138–1147. ISBN: 978-1-932432-66-4 (cit. on pp. 40, 49).
- Lin, Dekang (1998a). An Information Theoretic Definition of Similarity. In: *Machine Learning: Proceedings of the Fifteenth International Conference (ICML '98)*. Ed. by Jude W. Shavlik. San Francisco, CA: Morgan Kaufmann, pp. 296–304. ISBN: 978-1-55860-556-5 (cit. on p. 19).
- (1998b). Automatic Retrieval and Clustering of Similar Words. In: *COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics: Proceedings of the Conference*. Vol. 2. New Brunswick, NJ: Association for Computational Linguistics, Aug. 1998, pp. 768–774. DOI: 10.3115/980691.980696 (cit. on pp. 40, 44).
- Lin, Dekang, Shaojun Zhao, Lijuan Qin, and Ming Zhou (2003). Identifying Synonyms Among Distributionally Similar Words. In: *IJCAI-03: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*. Ed. by Georg Gottlob and Toby Walsh. San Francisco, CA: Morgan Kaufmann, Aug. 2003, pp. 1492–1493 (cit. on p. 40).
- Lippman, Louis G. and Mara L. Dunn (2000). Contextual Connections Within Puns: Effects on Perceived Humor and Memory. In: *Journal of Gen-*

- eral Psychology* 127(2) (Apr. 2000), pp. 185–197. ISSN: 0022-1309. DOI: 10.1080/00221300009598578 (cit. on p. 77).
- Lucas, Teresa (2004). *Deciphering the Meaning of Puns in Learning English as a Second Language: A Study of Triadic Interaction*. Ph.D. thesis. Florida State University (cit. on p. 81).
- Ludlow, Peter J. (1996). *Semantic Ambiguity and Underspecification* (review). In: *Computational Linguistics* 3(23), pp. 476–482. ISSN: 0891-2017 (cit. on p. 3).
- Mallery, John C. (1988). *Thinking About Foreign Policy: Finding an Appropriate Rule for Artificial Intelligence Computers*. Master's thesis. Cambridge, MA: Department of Political Science, Massachusetts Institute of Technology, Feb. 1988 (cit. on p. 2).
- Manandhar, Suresh and Deniz Yuret, eds. (2013). **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics: Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Stroudsburg, PA: Association for Computational Linguistics, June 2013. ISBN: 978-1-937284-49-7 (cit. on p. 32).
- Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). Cambridge: Cambridge University Press. ISBN: 978-0-521-86571-5 (cit. on pp. 95, 124).
- Manning, Christopher D. and Hinrich Schütze (1999). Foundations of Statistical Natural Language Processing. In: MIT Press. Chap. Word Sense Disambiguation, pp. 229–264. ISBN: 978-0-262-13360-9 (cit. on p. 2).
- Markert, Katja and Malvina Nissim (2007). SemEval-2007 Task 08: Metonymy Resolution at SemEval-2007. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 36–41 (cit. on p. 2).
- Màrquez, Lluís, Gerard Escudero, David Martínez, and German Rigau (2007). Supervised Corpus-based Methods for wsd. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Technology. Springer. Chap. 7, pp. 167–216. ISBN: 978-1-4020-6870-6 (cit. on p. 21).
- Martínez, David, Oier López de Lacalle, and Eneko Agirre (2008). On the Use of Automatically Acquired Examples for All-nouns Word Sense Disambiguation. In: *Journal of Artificial Intelligence Research* 33(1) (Sept. 2008), pp. 79–107. ISSN: 1076-9757. DOI: 10.1613/jair.2395 (cit. on p. 40).
- Matuschek, Michael and Iryna Gurevych (2013). Dijkstra-wsa: A Graph-based Approach to Word Sense Alignment. In: *Transactions of the Association for Computational Linguistics* 1 (May 2013), pp. 151–164. ISSN: 2307-387x (cit. on pp. 60, 61, 67).
- Matuschek, Michael, Tristan Miller, and Iryna Gurevych (2014). A Language-independent Sense Clustering Approach for Enhanced wsd. In: *Proceedings of the 12th Edition of the KONVENS Conference*. Ed. by Josef Ruppenhofer and Gertrud Faaß. Universitätsverlag Hildesheim, Oct. 2014, pp. 11–21. ISBN: 978-3-934105-46-1 (cit. on pp. 10, 90).
- Mayfield, James, Javier Artiles, and Hoa Trang Dang (2012). Overview of the TAC2012 Knowledge Base Population Track. In: *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*. Gaithersburg, MD: National Institute of Standards and Technology, Nov. 2012 (cit. on p. 109).
- McCarthy, Diana (2002). Lexical Substitution as a Task for wsd Evaluation. In: *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambigua-*

- tion: *Recent Successes and Future Directions*. Association for Computational Linguistics. New Brunswick, NJ: Association for Computational Linguistics, pp. 109–115. DOI: 10.3115/1118675.1118691 (cit. on p. 111).
- McCarthy, Diana (2006). Relating WordNet Senses for Word Sense Disambiguation. In: *Proceedings of the Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*. Stroudsburg, PA: Association for Computational Linguistics, Apr. 2006, pp. 17–24 (cit. on p. 62).
- McCarthy, Diana and Roberto Navigli (2007). SemEval-2007 Task 10: English Lexical Substitution Task. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 48–53 (cit. on pp. 31, 115).
- (2009). The English Lexical Substitution Task. In: *Language Resources and Evaluation* 43(2) (June 2009), pp. 139–159. ISSN: 1574-020X. DOI: 10.1007/s10579-009-9084-1 (cit. on pp. 23, 111, 115).
- McNamee, Paul and Hoa Trang Dang (2009). Overview of the TAC 2009 Knowledge Base Population Track. In: *Proceedings of the Second Text Analysis Conference (TAC 2009)*. Nov. 2009 (cit. on p. 105).
- McNemar, Quinn (1947). Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. In: *Psychometrika* 12(2) (June 1947), pp. 153–157. ISSN: 0033-3123. DOI: 10.1007/BF02295996 (cit. on pp. 71, 93).
- Meijssen, Gerard (2009). The Philosophy Behind OmegaWiki and the Visions for the Future. In: *Lexicography at a Crossroads: Dictionaries and Encyclopedias Today, Lexicographical Tools Tomorrow*. Ed. by Henning Bergenholtz, Sandro Nielsen, and Sven Tarp. Vol. 90 of Linguistic Insights: Studies in Language and Communication. Bern: Peter Lang, pp. 91–98. ISSN: 1424-8689. ISBN: 978-3-03911-799-4 (cit. on p. 90).
- Merriam-Webster's Collegiate Dictionary (2004). 11th edition. Springfield, MA: Merriam-Webster. ISBN: 978-0-87779-808-8 (cit. on p. 3).
- Meyer, Christian M. (2013). *Wiktionary: The Metalexicographic and Natural Language Processing Perspective*. Dr.-Ing. thesis. Technische Universität Darmstadt, Oct. 2013 (cit. on p. 15).
- Meyer, Christian M. and Iryna Gurevych (2010). How Web Communities Analyze Human Language: Word Senses in Wiktionary. In: *Proceedings of WebSci10: Extending the Frontiers of Society On-Line*. Apr. 2010 (cit. on p. 60).
- (2011). What Psycholinguists Know About Chemistry: Aligning Wiktionary and WordNet for Increased Domain Coverage. In: *Proceedings of the Fifth International Joint Conference on Natural Language Processing [IJCNLP]*. Asian Federation of Natural Language Processing, Nov. 2011, pp. 883–892. ISBN: 978-974-466-564-5 (cit. on pp. 60, 64, 67).
- Meyer, Christian M., Margot Mieskes, Christian Stab, and Iryna Gurevych (2014). DKPro Agreement: An Open-source Java Library for Measuring Inter-rater Agreement. In: *The 25th International Conference on Computational Linguistics: Proceedings of the Conference System Demonstrations [COLING]*. Ed. by Lamia Tounsi and Rafal Rak. Aug. 2014, pp. 105–109. ISBN: 978-1-941643-27-3 (cit. on p. 103).
- Mihalcea, Rada (2005). Unsupervised Large-vocabulary Word Sense Disambiguation with Graph-based Algorithms for Sequence Data Labeling. In: *Human Language Technology Conference and Conference on Empirical Methods*

- in *Natural Language Processing: Proceedings of the Conference [HLT-EMNLP]*. Stroudsburg, PA: Association for Computational Linguistics, Oct. 2005, pp. 411–418. DOI: 10.3115/1220575.1220627 (cit. on p. 20).
- (2006). Using Wikipedia for Automatic Word Sense Disambiguation. In: *Human Language Technologies 2007: the Conference of the North American Chapter of the Association for Computational Linguistics: Proceedings of the Main Conference [NAACL-HLT]*. Ed. by Candace Sidner, Tanja Schultz, Matthew Stone, and ChengXiang Zhai. Stroudsburg, PA: Association for Computational Linguistics, Apr. 2006, pp. 196–203 (cit. on p. 15).
- (2007). Knowledge-based Methods for wsd. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Technology. Springer. Chap. 5, pp. 107–131. ISBN: 978-1-4020-6870-6 (cit. on p. 18).
- (2011). Word Sense Disambiguation. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. New York, NY: Springer, pp. 1027–1030. ISBN: 978-0-387-30768-8 (cit. on p. 2).
- Mihalcea, Rada and Timothy Chklovski (2003). Open Mind Word Expert: Creating Large Annotated Data Collections with Web Users' Help. In: *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*. Ed. by Anne Abeillé, Silvia Hansen-Schirra, and Hans Uszkoreit. Stroudsburg, PA: Association for Computational Linguistics, Apr. 2003, pp. 53–60 (cit. on pp. 15, 21, 35, 84, 111).
- Mihalcea, Rada, Timothy Chklovski, and Adam Kilgarriff (2004). The SENSEVAL-3 English Lexical Sample Task. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Philip Edmonds. New Brunswick, NJ: Association for Computational Linguistics, July 2004, pp. 25–28 (cit. on pp. 31, 105).
- Mihalcea, Rada and Andras Csomai (2005). SenseLearner: Word Sense Disambiguation for All Words in Unrestricted Text. In: *Interactive Poster and Demonstration Sessions: Proceedings [ACL]*. New Brunswick, NJ: Association for Computational Linguistics, June 2005, pp. 53–56. DOI: 10.3115/1225753.1225767 (cit. on p. 101).
- Mihalcea, Rada and Philip Edmonds, eds. (2004). *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. New Brunswick, NJ: Association for Computational Linguistics, July 2004 (cit. on p. 31).
- Mihalcea, Rada and Ehsanul Faruque (2004). SenseLearner: Minimally Supervised Word Sense Disambiguation for All Words in Open Text. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Philip Edmonds. New Brunswick, NJ: Association for Computational Linguistics, July 2004, pp. 155–158 (cit. on p. 70).
- Mihalcea, Rada and Dan Moldovan (1997). A Method for Word Sense Disambiguation of Unrestricted Text. In: *37th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. Association for Computational Linguistics, June 1997, pp. 152–158. ISBN: 978-1-55860-609-8. DOI: 10.3115/1034678.1034709 (cit. on p. 19).
- Mihalcea, Rada and Dan I. Moldovan (2001). Automatic Generation of a Coarse Grained WordNet. In: *Proceedings of the SIGLEX Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations* (cit. on p. 61).

- Mihalcea, Rada and Carlo Strapparava (2005). Making Computers Laugh: Investigations in Automatic Humor Recognition. In: *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference [HLT-EMNLP]*. Stroudsburg, PA: Association for Computational Linguistics, Oct. 2005, pp. 531–538. DOI: 10.3115/1220575.1220642 (cit. on pp. 80, 83).
- (2006). Learning to Laugh (Automatically): Computational Models for Humor Recognition. In: *Computational Intelligence* 22(2) (May 2006), pp. 126–142. ISSN: 1467-8640. DOI: 10.1111/j.1467-8640.2006.00278.x (cit. on pp. 77, 80, 83).
- Mihalcea, Rada, Carlo Strapparava, and Stephen Pulman (2010). Computational Models for Incongruity Detection in Humour. In: *Computational Linguistics and Intelligent Text Processing: 11th International Conference, CICLING 2010*. Ed. by Alexander Gelbukh. Vol. 6008 of Theoretical Computer Science and General Issues. Berlin/Heidelberg: Springer, Feb. 2010, pp. 364–374. ISBN: 978-3-642-12115-9. DOI: 10.1007/978-3-642-12116-6_30 (cit. on pp. 80, 96, 98, 132).
- Miller, George A., Martin Chodorow, Shari Landes, Claudio Leacock, and Robert G. Thomas (1994). Using a Semantic Concordance for Sense Identification. In: *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey [HLT]*. San Francisco, CA: Morgan Kaufmann, Mar. 1994, pp. 240–243. ISBN: 978-1-55860-357-8 (cit. on pp. 29, 72, 100).
- Miller, George A., Claudia Leacock, Randee Teng, and Ross T. Bunker (1993). A Semantic Concordance. In: *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey [HLT]*. San Francisco, CA: Morgan Kaufmann, Mar. 1993, pp. 303–308. ISBN: 978-1-55860-324-0. DOI: 10.3115/1075671.1075742 (cit. on pp. 15, 47, 86, 111).
- Miller, Tristan, Darina Benikova, and Sallam Abualhaija (2015). GermEval 2015: LexSub – A Shared Task for German-language Lexical Substitution. In: *Proceedings of GermEval 2015: LexSub*. Sept. 2015, pp. 1–9 (cit. on pp. 10, 115).
- Miller, Tristan, Chris Biemann, Torsten Zesch, and Iryna Gurevych (2012). Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In: *Proceedings of COLING 2012: Technical Papers*. Ed. by Martin Kay and Christian Boitet. Dec. 2012, pp. 1781–1796 (cit. on pp. 8, 55, 72, 106).
- Miller, Tristan, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych (2013). DKPro WSD: A Generalized UIMA-based Framework for Word Sense Disambiguation. In: *ACL 2013: 51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: System Demonstrations*. Stroudsburg, PA: Association for Computational Linguistics, Aug. 2013, pp. 37–42 (cit. on p. 10).
- Miller, Tristan and Iryna Gurevych (2014). WordNet–Wikipedia–Wiktionary: Construction of a Three-way Alignment. In: *LREC 2014, Ninth International Conference on Language Resources and Evaluation*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis. European Language Resources Association, May 2014, pp. 2094–2100. ISBN: 978-2-9517408-8-4 (cit. on p. 8).
- (2015). Automatic Disambiguation of English Puns. In: *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federa-*

- tion of Natural Language Processing: Proceedings of the Conference [ACL-IJCNLP]*. Vol. 1. Stroudsburg, PA: Association for Computational Linguistics, July 2015, pp. 719–729. ISBN: 978-1-941643-72-3 (cit. on p. 10).
- Miller, Tristan, Mohamed Khemakhem, Richard Eckart de Castilho, and Iryna Gurevych (2016). Sense-annotating a Lexical Substitution Data Set with Ubyline. In: *LREC 2016, Tenth International Conference on Language Resources and Evaluation*. To appear. European Language Resources Association, May 2016 (cit. on pp. 10, 35, 119).
- Miller, Tristan and Mladen Turković (2016). Towards the Automatic Detection and Identification of English Puns. In: *European Journal of Humour Research* 4(1) (Jan. 2016), pp. 59–75. ISSN: 2307-700X (cit. on p. 10).
- Monnot, Michel (1981). *Selling America: Puns, Language and Advertising*. Washington, DC: University Press of America. ISBN: 978-0-8191-2002-1 (cit. on p. 81).
- (1982). Puns in Advertising: Ambiguity as Verbal Aggression. In: *Maledicta* 6, pp. 7–20. ISSN: 0363-3659 (cit. on pp. 76, 78).
- Morkes, John, Hadyn K. Kernal, and Clifford Nass (1999). Effects of Humor in Task-oriented Human–computer Interaction and Computer-mediated Communication: A Direct Test of SRCT Theory. In: *Human–Computer Interaction* 14(4) (Dec. 1999), pp. 395–435. ISSN: 0737-0024. DOI: 10.1207/S15327051HCI1404_2 (cit. on p. 76).
- Morris, Jane and Graeme Hirst (1991). Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. In: *Computational Linguistics* 17(1) (Mar. 1991), pp. 21–48. ISSN: 0891-2017 (cit. on p. 20).
- Murray, G. Craig and Rebecca Green (2004). Lexical Knowledge and Human Disagreement on a WSD Task. In: *Computer Speech and Language* 18, pp. 209–222. ISSN: 0885-2308. DOI: 10.1016/j.csl.2004.05.001 (cit. on p. 30).
- Naber, Daniel (2005). OpenThesaurus: ein offenes deutsches Wortnetz. In: *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung 2005 in Bonn*. Ed. by Bernhard Fisseni, Hans-Christian Schmitz, and Bernhard Schröder und Petra Wagner. Frankfurt: Peter Lang, pp. 422–433. ISBN: 978-3-631-53874-6 (cit. on p. 15).
- Nakov, Preslav, Torsten Zesch, Daniel Cer, and David Jurgens, eds. (2015). *The 9th International Workshop on Semantic Evaluation: Proceedings of SemEval-2015*. Stroudsburg, PA: Association for Computational Linguistics, June 2015. ISBN: 978-1-941643-40-2 (cit. on p. 32).
- Navigli, Roberto (2006). Meaningful Clustering of Senses Helps Boost Word Sense Disambiguation Performance. In: *COLING–ACL 2006: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: Volume 1*. Stroudsburg, PA: Association for Computational Linguistics, July 2006, pp. 105–112. ISBN: 978-1-932432-65-7. DOI: 10.3115/1220175.1220189 (cit. on pp. 58, 62, 67, 68, 71).
- (2009). Word Sense Disambiguation: A Survey. In: *ACM Computing Surveys* 41(2) (Feb. 2009), 10:1–10:69. ISSN: 0360-0300. DOI: 10.1145/1459352.1459355 (cit. on pp. 2, 25, 28).
- Navigli, Roberto, David Jurgens, and Daniele Vannella (2013). SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In: **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics: Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Ed. by Suresh Manandhar and Deniz Yuret. Stroudsburg, PA:

- Association for Computational Linguistics, June 2013, pp. 222–231. ISBN: 978-1-937284-49-7 (cit. on p. 55).
- Navigli, Roberto and Mirella Lapata (2010). An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(4) (Apr. 2010), pp. 678–692. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.36 (cit. on pp. 20, 106).
- Navigli, Roberto, Kenneth C. Litkowski, and Orin Hargraves (2007). SemEval-2007 Task 07: Coarse-grained English All-words Task. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 30–35 (cit. on pp. 31, 46, 92).
- Navigli, Roberto and Simone Paolo Ponzetto (2010). BabelNet: Building a Very Large Multilingual Semantic Network. In: *48th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. Vol. 1. Stroudsburg, PA: Association for Computational Linguistics, July 2010, pp. 216–225. ISBN: 978-1-932432-66-4 (cit. on p. 56).
- (2013). An Overview of BabelNet and Its API for Multilingual Language Processing. In: *The People’s Web Meets NLP: Collaboratively Constructed Language Resources*. Ed. by Iryna Gurevych and Jungi Kim. Theory and Applications of Natural Language Processing. Berlin/Heidelberg: Springer, pp. 177–197. ISSN: 2192-032X. ISBN: 978-3-642-35084-9. DOI: 10.1007/978-3-642-35085-6_7 (cit. on pp. 60, 100).
- Navigli, Roberto and Daniele Vannella (2013). SemEval-2013 Task 11: Word Sense Induction and Disambiguation Within an End-user Application. In: **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics: Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Ed. by Suresh Manandhar and Deniz Yuret. Stroudsburg, PA: Association for Computational Linguistics, June 2013, pp. 193–201. ISBN: 978-1-937284-49-7 (cit. on pp. 23, 55, 109).
- Navigli, Roberto and Paola Velardi (2005). Structural Semantic Interconnections: A Knowledge-based Approach to Word Sense Disambiguation. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(7), pp. 1075–1086. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.149 (cit. on p. 52).
- Ng, Hwee Tou (1997). Getting Serious About Word Sense Disambiguation. In: *Tagging Text with Lexical Semantics: Why, What, and How? Proceedings of the Workshop*. Somerset, NJ: Association for Computational Linguistics, Apr. 1997, pp. 1–7 (cit. on p. 21).
- Niemann, Elisabeth and Iryna Gurevych (2011). The People’s Web Meets Linguistic Knowledge: Automatic Sense Alignment of Wikipedia and WordNet. In: *Proceedings of the Ninth International Conference on Computational Semantics: iwcs 2011*. Ed. by Johan Bos and Stephen Pulman. Jan. 2011, pp. 205–214 (cit. on p. 60).
- Palmer, Martha (2000). Consistent Criteria for Sense Distinctions. In: *Computers and the Humanities* 34(1&2) (Apr. 2000), pp. 217–222. ISSN: 0010-4817. DOI: 10.1023/A:1002613125904 (cit. on p. 3).
- Palmer, Martha, Olga Babko-Malaya, and Hoa Trang Dang (2004). Different Sense Granularities for Different Applications. In: *2nd International Workshop on Scalable Natural Language Understanding (scanLU): Proceedings of the HLT-NAACL 2004 Workshop*. Ed. by Robert Porzel. Stroudsburg, PA: Association for Computational Linguistics, May 2004, pp. 49–56 (cit. on p. 62).

- Palmer, Martha, Hoa Trang Dang, and Christiane Fellbaum (2007). Making Fine-grained and Coarse-grained Sense Distinctions, Both Manually and Automatically. In: *Natural Language Engineering* 13(2) (June 2007), pp. 137–163. ISSN: 1351-3249. DOI: 10.1017/S135132490500402X (cit. on p. 62).
- Palmer, Martha, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang (2001). English Tasks: All-words and Verb Lexical Sample. In: *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*. Ed. by Judita Preiss and David Yarowsky. New Brunswick, NJ: Association for Computational Linguistics, July 2001, pp. 21–24 (cit. on pp. 46, 92).
- Palmer, Martha, Hwee Tou Ng, and Hoa Trang Dang (2007). Evaluation of wsd Systems. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Technology. Springer. Chap. 4, pp. 75–106. ISBN: 978-1-4020-6870-6 (cit. on pp. 23, 25, 28, 90).
- Passonneau, Rebecca J. (2006). Measuring Agreement on Set-valued Items (MASI) for Semantic and Pragmatic Annotation. In: *5th Edition of the International Conference on Language Resources and Evaluation*. European Language Resources Association, May 2006, pp. 831–836. ISBN: 978-2-9517408-2-2 (cit. on pp. 27, 86).
- Passonneau, Rebecca J., Collin Baker, Christiane Fellbaum, and Nancy Ide (2012). The MASC Word Sense Sentence Corpus. In: *LREC 2012, Eighth International Conference on Language Resources and Evaluation*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. European Language Resources Association, May 2012, pp. 3025–3030. ISBN: 978-2-9517408-7-7 (cit. on pp. 84, 100, 111).
- Passonneau, Rebecca J., Nizar Habash, and Owen Rambow (2006). Inter-annotator Agreement on a Multilingual Semantic Annotation Task. In: *5th Edition of the International Conference on Language Resources and Evaluation*. European Language Resources Association, May 2006, pp. 1951–1956. ISBN: 978-2-9517408-2-2 (cit. on p. 86).
- Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen (2005). Sense-Relate::TargetWord – A Generalized Framework for Word Sense Disambiguation. In: *Interactive Poster and Demonstration Sessions: Proceedings [ACL]*. New Brunswick, NJ: Association for Computational Linguistics, June 2005, pp. 73–76. DOI: 10.3115/1225753.1225772 (cit. on p. 101).
- Pazienza, Maria Teresa, Armando Stellato, and Alexandra Tudorache (2008). JMWNL: An Extensible Multilingual Library for Accessing Wordnets in Different Languages. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias. European Language Resources Association, May 2008, pp. 2074–2078. ISBN: 978-2-9517408-4-6 (cit. on p. 105).
- Pearson, Karl (1896). Mathematical Contributions to the Theory of Evolution—III. Regression, Heredity, and Panmixia. In: *Philosophical Transactions of the Royal Society of London: Series A* 187, pp. 253–318. ISSN: 1364-503X. DOI: 10.1098/rsta.1896.0007 (cit. on p. 121).
- Pedersen, Ted (2007). Unsupervised Corpus-based Methods for wsd. In: *Word Sense Disambiguation: Algorithms and Applications*. Ed. by Eneko Agirre and Philip Edmonds. Vol. 33 of Text, Speech, and Language Tech-

- nology. Springer. Chap. 6, pp. 133–166. ISBN: 978-1-4020-6870-6 (cit. on p. 22).
- Pedersen, Ted, Satanjeev Banerjee, and Siddharth Patwardhan (2005). *Maximizing Semantic Relatedness to Perform Word Sense Disambiguation*. Research Report UMSI 2005/25. University of Minnesota Supercomputing Institute, Mar. 2005 (cit. on p. 20).
- Peters, Wim, Ivonne Peters, and Piek Vossen (1998). Automatic Sense Clustering in EuroWordNet. In: *Proceedings of the First International Conference on Language Resources and Evaluation*. European Language Resources Association, May 1998, pp. 409–416 (cit. on p. 61).
- Ponzetto, Simone Paolo and Roberto Navigli (2010). Knowledge-rich Word Sense Disambiguation Rivaling Supervised Systems. In: *48th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. Vol. 1. Stroudsburg, PA: Association for Computational Linguistics, July 2010, pp. 1522–1531. ISBN: 978-1-932432-66-4 (cit. on pp. 20, 39, 43, 49, 68, 89, 105).
- Pradhan, Sameer S., Edward Loper, Dmitriy Dligach, and Martha Palmer (2007). Semeval-2007 Task 17: English Lexical Sample, SRL and All Words. In: *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*. Ed. by Eneko Agirre, Lluís Màrquez, and Richard Wicentowski. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 87–92 (cit. on p. 31).
- Preiss, Judita, Jon Dehdari, Josh King, and Dennis Mehay. Refining the Most Frequent Sense Baseline. In: *SEW-2009: Semantic Evaluations: Recent Achievements and Future Directions: Proceedings of the Workshop*. Stroudsburg, PA: Association for Computational Linguistics, pp. 10–18. ISBN: 978-1-932432-31-2 (cit. on p. 29).
- Preiss, Judita and Mark Stevenson, eds. (2004). *Computer Speech and Language* 18(3) (July 2004): *Special Issue on Word Sense Disambiguation*. ISSN: 0885-2308 (cit. on p. 2).
- Raileanu, Diana, Paul Buitelaar, Spela Vintar, and Jörg Bay (2002). Evaluation Corpora for Sense Disambiguation in the Medical Domain. In: *LREC 2002: Third International Conference on Language Resources and Evaluation*. Ed. by Manuel González Rodríguez, Suárez Araujo, and Carmen Paz. Vol. 2. European Language Resources Association, May 2002, pp. 609–612. ISBN: 978-2-9517408-0-8 (cit. on pp. 112, 120, 121).
- Rapp, Reinhard (2004). A Freely Available Automatically Generated Thesaurus of Related Words. In: *LREC 2004: Fourth International Conference on Language Resources and Evaluation*. Ed. by Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva. May 2004, pp. 395–398. ISBN: 978-2-9517408-1-5 (cit. on p. 41).
- Raskin, Victor (1985). *Semantic Mechanisms of Humor*. Vol. 24 of *Studies in Linguistics and Philosophy*. Springer Netherlands. ISSN: 0924-4662. ISBN: 978-90-277-1821-1. DOI: 10.1007/978-94-009-6472-3 (cit. on p. 90).
- Redfern, Walter (1984). *Puns*. Oxford: Basil Blackwell. ISBN: 978-0-631-13793-1 (cit. on p. 82).
- Resnik, Philip (1995). Using Information Content to Evaluate Semantic Similarity. In: *IJCAI 95: International Joint Conference on Artificial Intelligence: Proceedings Volume 1*. San Francisco, CA: Morgan Kaufmann, pp. 448–453. ISBN: 978-1-558-60363-9 (cit. on p. 19).
- Resnik, Philip and David Yarowsky (1999). Distinguishing Systems and Distinguishing Senses: New Evaluation Methods for Word Sense Disam-

- biguation. In: *Natural Language Engineering* 5(2) (June 1999), pp. 113–133. ISSN: 1351-3249. DOI: 10.1017/S1351324999002211 (cit. on p. 62).
- Ritchie, Graeme D. (2004). *The Linguistic Analysis of Jokes*. London: Routledge. ISBN: 978-0-415-30983-7 (cit. on p. 77).
- (2005). Computational Mechanisms for Pun Generation. In: *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*. Ed. by Graham Wilcock, Kristiina Jokinen, Chris Mellish, and Ehud Reiter, pp. 125–132 (cit. on p. 75).
- Roget, Peter Mark (1852). *Thesaurus of English Words and Phrases Classified and Arranged so as to Facilitate the Expression of Ideas and Assist in Literary Composition*. Harlow: Longman (cit. on p. 15).
- Rothe, Sascha and Hinrich Schütze (2015). AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In: *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: Proceedings of the Conference [ACL-IJCNLP]*. Vol. 1. Stroudsburg, PA: Association for Computational Linguistics, July 2015, pp. 1793–1803. ISBN: 978-1-941643-72-3 (cit. on pp. 40, 55).
- Rubinstein, Frankie (1984). *A Dictionary of Shakespeare's Sexual Puns and Their Significance*. London: Macmillan. ISBN: 978-0-333-34308-1 (cit. on p. 77).
- Ruiz-Casado, Maria, Enrique Alfonseca, and Pablo Castells (2005). Automatic Assignment of Wikipedia Encyclopedic Entries to WordNet Synsets. In: *Advances in Web Intelligence: Third International Atlantic Web Intelligence Conference, AWIC 2005*. Ed. by Piotr S. Szczepaniak and Adam Niewiadomski. Vol. 3528 of Lecture Notes in Artificial Intelligence. Berlin/Heidelberg: Springer, pp. 380–386. ISSN: 0302-9743. ISBN: 978-3-540-26219-0. DOI: 10.1007/11495772_59 (cit. on p. 60).
- Ruppenhofer, Josef, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk (2010). *FrameNet II: Extended Theory and Practice*. Berkeley, CA: International Computer Science Institute, Sept. 2010 (cit. on p. 100).
- Saussure, Ferdinand de (1916). *Cours de linguistique générale*. Paris: Librairie Payot & Cie (cit. on p. 41).
- Schemann, Hans (2011). *Deutsche Idiomatik: Wörterbuch der deutschen Redewendungen im Kontext*. 2nd edition. Berlin/Boston, MA: Walter de Gruyter. ISBN: 978-3-11-025940-7. DOI: 10.1515/9783110217896 (cit. on p. 124).
- Schmid, Helmud (1994). Probabilistic Part-of-speech Tagging Using Decision Trees. In: *New Methods in Language Processing*. Ed. by Daniel B. Jones and Harold L. Somers. London/New York, NY: Routledge, pp. 154–164. ISBN: 978-1-85728-711-0 (cit. on p. 105).
- Schröter, Thorsten (2005). *Shun the Pun, Rescue the Rhyme? The Dubbing and Subtitling of Language-play in Film*. Ph.D. thesis. Karlstad University. ISSN: 1403-8099. ISBN: 978-91-85335-50-3 (cit. on p. 76).
- Schütze, Hinrich and Jan O. Pedersen (1995). Information Retrieval Based on Word Senses. In: *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval [SDAIR]*, pp. 161–175 (cit. on p. 23).
- Schwab, Didier, Jérôme Goulian, and Andon Tchechmedjiev (2013). Désambiguïsation lexicale de textes: efficacité qualitative et temporelle d'un algorithme à colonies de fourmis. In: *Traitement Automatique des Langues* 54(1), pp. 99–138. ISSN: 1965-0906 (cit. on p. 55).

- Sharoff, Serge (2006). Open-source Corpora: Using the Net to Fish for Linguistic Data. In: *International Journal of Corpus Linguistics* 11(4), pp. 435–462. ISSN: 1384-6655. DOI: 10.1075/ijcl.11.4.05sha (cit. on p. 116).
- Sharp, Harold S. (1984). *Advertising Slogans of America*. Metuchen, NJ: Scarecrow Press. ISBN: 978-0-8108-1681-7 (cit. on p. 81).
- Snow, Rion, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng (2007). Learning to Merge Word Senses. In: *EMNLP-CONLL: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Stroudsburg, PA: Association for Computational Linguistics, June 2007, pp. 1005–1014 (cit. on pp. 62, 68–71).
- Snyder, Benjamin and Martha Palmer (2004). The English All-words Task. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Philip Edmonds. New Brunswick, NJ: Association for Computational Linguistics, July 2004, pp. 41–43 (cit. on pp. 31, 46, 62, 70, 92).
- Soanes, Catherine and Angus Stevenson, eds. (2003). *Oxford Dictionary of English*. Oxford University Press. ISBN: 978-0-19-861347-3 (cit. on p. 62).
- Stokoe, Christopher, Michael P. Oakes, and John Tait (2003). Word Sense Disambiguation in Information Retrieval Revisited. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, July 2003, pp. 159–166. ISBN: 978-1-58113-646-3. DOI: 10.1145/860435.860466 (cit. on p. 23).
- Taylor, Julia M. and Lawrence J. Mazlack (2004). Computationally Recognizing Wordplay in Jokes. In: *Proceedings of the Twenty-sixth Annual Conference of the Cognitive Science Society [CogSci]*. Ed. by Kenneth Forbus, Dedre Gentner, and Terry Regier. Cognitive Science Society, Aug. 2004, pp. 1315–1320. ISSN: 1047-1316. ISBN: 978-0-9768318-0-8 (cit. on p. 80).
- Thompson, Della, ed. (1993). *The Oxford Dictionary of Current English*. 2nd edition. Oxford University Press. ISBN: 978-0-19-283127-9 (cit. on p. 3).
- Tomuro, Noriko (2001). Tree-cut and a Lexicon Based on Systematic Polysemy. In: *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics [NAACL]*. Association for Computational Linguistics. DOI: 10.3115/1073336.1073346 (cit. on p. 61).
- Toral, Antonio (2009). The Lexical Substitution Task at EVALITA 2009. In: *Poster and Workshop Proceedings of the 11th Conference of the Italian Association for Artificial Intelligence [EVALITA]*. Dec. 2009. ISBN: 978-88-903581-1-1 (cit. on pp. 115, 116).
- Toutanova, Kristina, Dan Klein, Christopher Manning, and Yoram Singer (2003). Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics [HLT-NAACL]*. Association for Computational Linguistics, pp. 252–259 (cit. on p. 89).
- Tugwell, David and Adam Kilgarrieff (2001). wasp-Bench: A Lexicographic Tool Supporting Word Sense Disambiguation. In: *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems*. Ed. by Judita Preiss and David Yarowsky. New Brunswick, NJ: Association for Computational Linguistics, July 2001, pp. 151–154 (cit. on p. 40).
- Valitutti, Alessandro, Carlo Strapparava, and Oliviero Stock (2008). Textual Affect Sensing for Computational Advertising. In: *Creative Intelligent Sys-*

- tems: *Papers from the AAAI Spring Symposium*. Ed. by Dan Ventura, Mary Lou Maher, and Simon Colton. Technical Report ss-08-03. Menlo Park, CA: AAAI Press, Mar. 2008, pp. 117–122. ISBN: 978-1-57735-359-1 (cit. on p. 76).
- Vasilescu, Florentina, Philippe Langlais, and Guy Lapalme (2004). Evaluating Variants of the Lesk Approach for Disambiguating Words. In: *LREC 2004: Fourth International Conference on Language Resources and Evaluation*. Ed. by Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa, and Raquel Silva. May 2004, pp. 633–636. ISBN: 978-2-9517408-1-5 (cit. on p. 38).
- Venour, Chris (1999). *The Computational Generation of a Class of Puns*. Master's thesis. Kingston, ON: Queen's University (cit. on p. 81).
- Véronis, Jean (1998). A Study of Polysemy Judgements and Inter-annotator Agreement. In: *Programme and Advanced Papers of the SENSEVAL Workshop*. Sept. 1998 (cit. on p. 25).
- Vickrey, David, Luke Biewald, Marc Teyssier, and Daphne Koller (2005). Word-sense Disambiguation for Machine Translation. In: *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing: Proceedings of the Conference [HLT-EMNLP]*. Stroudsburg, PA: Association for Computational Linguistics, Oct. 2005, pp. 771–778. DOI: 10.3115/1220575.1220672 (cit. on p. 23).
- Vossen, Piek, ed. (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Springer, Oct. 1998. ISBN: 978-0-7923-5295-2 (cit. on p. 105).
- Weaver, Warren (1955). Translation. In: *Machine Translation of Languages: Fourteen Essays*. Ed. by William N. Locke and A. Donald Booth. Cambridge, MA/New York, NY: Massachusetts Institute of Technology, John Wiley, and Sons (cit. on p. 1).
- Weeds, Julie Elizabeth (2003). *Measures and Applications of Lexical Distributional Similarity*. D.Phil. thesis. University of Sussex, Sept. 2003 (cit. on p. 40).
- Winchester, Simon (2011). A Verb for Our Frantic Times. In: *The New York Times* (May 2011), wk9. ISSN: 0362-4331 (cit. on p. 1).
- Wlotzka, Marcel (2015). *Context-sensitive Lookup of Lexical Information with Stream-based Disambiguation Methods*. M.Sc. thesis. Technische Universität Darmstadt, June 2015 (cit. on p. 109).
- Wu, Zhibiao and Martha Palmer (1994). Verb Semantics and Lexical Selection. In: *32nd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. New Brunswick, NJ: Association for Computational Linguistics, June 1994, pp. 133–138. DOI: 10.3115/981732.981751 (cit. on p. 19).
- Wurth, Leopold (1895). *Das Wortspiel bei Shakspeare*. Vienna: Wilhelm Braumüller (cit. on p. 77).
- Yarowsky, David (1993). One Sense per Collocation. In: *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey [HLT]*. San Francisco, CA: Morgan Kaufmann, Mar. 1993, pp. 266–271. ISBN: 978-1-55860-324-0. DOI: 10.3115/1075671.1075731 (cit. on pp. 21, 39).
- (1994). Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In: *32nd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference [ACL]*. New Brunswick, NJ: Association for Computational Linguistics, June 1994, pp. 88–95. DOI: 10.3115/981732.981745 (cit. on p. 2).

- Yarowsky, David (2010). Word Sense Disambiguation. In: *Handbook of Natural Language Processing*. Ed. by Nitin Indurkha and Fred J. Damerau. 2nd edition. Chapman and Hall/CRC, pp. 315–338. ISBN: 978-1-4200-8592-1 (cit. on p. 2).
- Yarowsky, David and Radu Florian (2002). Evaluating Sense Disambiguation Across Diverse Parameter Spaces. In: *Natural Language Engineering* 8(4) (Dec. 2002), pp. 293–310. ISSN: 1351-3249. DOI: 10.1017/S135132490200298X (cit. on p. 21).
- Yimam, Seid Muhie, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann (2013). WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In: *ACL 2013: 51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: System Demonstrations*. Stroudsburg, PA: Association for Computational Linguistics, Aug. 2013, pp. 1–6 (cit. on p. 84).
- Yokogawa, Toshihiko (2002). Japanese Pun Analyzer Using Articulation Similarities. In: *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems: FUZZ 2002*. Vol. 2. Piscataway, NJ: IEEE, May 2002, pp. 1114–1119. ISSN: 1098-7584. ISBN: 978-0-7803-7280-1. DOI: 10.1109/FUZZ.2002.1006660 (cit. on p. 80).
- Yuret, Deniz (2004). Some Experiments with a Naive Bayes wsd System. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Ed. by Rada Mihalcea and Philip Edmonds. New Brunswick, NJ: Association for Computational Linguistics, July 2004, pp. 265–268 (cit. on p. 70).
- Zesch, Torsten, Iryna Gurevych, and Max Mühlhäuser (2007). Analyzing and Accessing Wikipedia as a Lexical Semantic Resource. In: *Datenstrukturen für linguistische Ressourcen und ihre Anwendungen / Data Structures for Linguistic Resources and Applications: Proceedings of the Biennial GLDV Conference 2007*. Tübingen: Narr, Apr. 2007, pp. 197–205. ISBN: 978-3-8233-6314-9 (cit. on pp. 15, 59).
- Zesch, Torsten, Christof Müller, and Iryna Gurevych (2008). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias. European Language Resources Association, May 2008, pp. 1646–1652. ISBN: 978-2-9517408-4-6 (cit. on p. 59).
- Zhao, Ying and George Karypis (2003). Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering. In: *Machine Learning* 55(3) (June 2003), pp. 311–331. ISSN: 0885-6125. DOI: 10.1023/B:MACH.0000027785.44527.d6 (cit. on p. 67).
- Zhong, Zhi and Hwee Tou Ng (2010). It Makes Sense: A Wide-coverage Word Sense Disambiguation System for Free Text. In: *48th Annual Meeting of the Association for Computational Linguistics: Proceedings of System Demonstrations [ACL]*. Stroudsburg, PA: Association for Computational Linguistics, July 2010, pp. 78–83 (cit. on pp. 101, 106).
- (2012). Word Sense Disambiguation Improves Information Retrieval. In: *50th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: Volume 1: Long Papers [ACL]*. Stroudsburg, PA: Association for Computational Linguistics, July 2012, pp. 273–282. ISBN: 978-1-937284-24-4 (cit. on p. 23).

- Zipf, George Kingsley (1949). *Human Behaviour and the Principle of Least Effort*. Cambridge, MA: Addison–Wesley (cit. on pp. 1, 66).
- Zorn, Hans-Peter and Iryna Gurevych (2013). UKP-WSI: UKP Lab SemEval Task 11 System Description. In: **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics: Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Ed. by Suresh Manandhar and Deniz Yuret. Stroudsburg, PA: Association for Computational Linguistics, June 2013, pp. 212–216. ISBN: 978-1-937284-49-7 (cit. on pp. 55, 109).
- Zwicky, Arnold M. and Elizabeth D. Zwicky (1986). Imperfect Puns, Markedness, and Phonological Similarity: With Fronds Like These, Who Needs Anemones? In: *Folia Linguistica* 20(3&4), pp. 493–503. ISSN: 0165-4004. DOI: 10.1515/flin.1986.20.3-4.493 (cit. on p. 81).

CURRICULUM VITÆ

EDUCATION

- 2000–2003 M.Sc. in Computer Science
University of Toronto
- 1995–1999 B.Sc. Hons. in Computer Science
University of Regina

SELECTED PROFESSIONAL EXPERIENCE

- 2011– Research scientist
Ubiquitous Knowledge Processing Lab
Technische Universität Darmstadt
- 2003–2005 Research scientist
Knowledge Management Department
German Research Center for Artificial Intelligence
- 1999–2000 Research assistant
School of Computing and Information Technology
Griffith University
- 1998 Research assistant
Department of Computer Science
University of Regina

PUBLICATION RECORD

- Yang Xiang and Tristan Miller (1999). A Well-behaved Algorithm for Simulating Dependence Structures of Bayesian Networks. In: *International Journal of Applied Mathematics* 1(8), pp. 923–932. ISSN: 1311-1728.
- Michael J. Maher, Allan Rock, Grigoris Antoniou, David Billington, and Tristan Miller (2000). Efficient Defeasible Reasoning Systems. In: *Proceedings: 12th IEEE International Conference on Tools with Artificial Intelligence: ICTAI 2000*. Piscataway, NJ: IEEE Press, Nov. 2000, pp. 384–392. ISSN: 1082-3409. ISBN: 978-0-7695-0909-9. DOI: 10.1109/TAI.2000.889898.
- (2001). Efficient Defeasible Reasoning Systems. In: *International Journal on Artificial Intelligence Tools* 10(4) (Dec. 2001), pp. 483–501. ISSN: 0218-2130. DOI: 10.1142/S0218213001000623.

- Tristan Miller (2003a). Essay Assessment with Latent Semantic Analysis. In: *Journal of Educational Computing Research* 29(4) (Dec. 2003), pp. 495–512. ISSN: 0735-6331. DOI: 10.2190/W5AR-DYPW-40KX-FL99.
- (2003b). *Generating Coherent Extracts of Single Documents Using Latent Semantic Analysis*. M.Sc. thesis. Department of Computer Science, University of Toronto, Mar. 2003.
- (2003c). Latent Semantic Analysis and the Construction of Coherent Extracts. In: *International Conference Recent Advances in Natural Language Processing: Proceedings [RANLP]*. Ed. by Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov. Sept. 2003, pp. 270–277. ISBN: 978-954-90906-6-6.
- (2004). Latent Semantic Analysis and the Construction of Coherent Extracts. In: *Recent Advances in Natural Language Processing III*. Ed. by Nicolas Nicolov, Kalina Botcheva, Galia Angelova, and Ruslan Mitkov. Vol. 260 of Current Issues in Linguistic Theory. Amsterdam/Philadelphia, PA: John Benjamins, pp. 277–286. ISSN: 0304-0763. ISBN: 978-1-58811-618-5. DOI: 10.1075/cilt.260.31mil.
- Bertin Klein, Tristan Miller, and Sandra Zilles (2005). Security Issues for Pervasive Personalized Communication Systems. In: *Security in Pervasive Computing: Second International Conference, SPC 2005*. Ed. by Dieter Hutter and Markus Ullmann. Vol. 3450 of Lecture Notes in Computer Science. Springer, Apr. 2005, pp. 56–62. ISSN: 0302-9743. ISBN: 978-3-540-25521-5. DOI: 10.1007/978-3-540-32004-3_7.
- Tristan Miller (2005a). Biblet: a Portable B_BT_EX Bibliography Style for Generating Highly Customizable XHTML. In: *TUGboat* 26(1), pp. 85–96. ISSN: 0896-3207.
- (2005b). Using the RPM Package Manager for L^AT_EX Packages. In: *TUGboat* 26(1), pp. 17–28. ISSN: 0896-3207.
- Tristan Miller and Stefan Agne (2005). Attention-based Information Retrieval Using Eye Tracker Data. In: *Proceedings of the Third International Conference on Knowledge Capture: K-CAP’05*. Ed. by Peter Clark and Guus Schreiber. New York, NY: ACM, Sept. 2005, pp. 209–210. ISBN: 978-1-59593-163-4. DOI: 10.1145/1088622.1088672.
- Tristan Miller and Elisabeth Wolf (2006). Word Completion with Latent Semantic Analysis. In: *The 18th International Conference on Pattern Recognition: ICPR 2006*. Ed. by Yuan Yan Tang, S. Patrick Wang, G. Lorette, Daniel So Yeung, and Hong Yan. Vol. 1. IEEE Press, Aug. 2006, pp. 1252–1255. ISSN: 1051-4651. ISBN: 978-0-7695-2521-1. DOI: 10.1109/ICPR.2006.1191.
- Elisabeth Wolf, Shankar Vembu, and Tristan Miller (2006). On the Use of Topic Models for Word Completion. In: *Advances in Natural Language Processing: 5th International Conference, FINTAL 2006*. Ed. by Tapio Salakoski, Filip Ginter, Sampo Pyysalo, and Tapio Pahikkala. Vol. 4139 of Lecture Notes in Computer Science. Berlin/Heidelberg: Springer, Aug. 2006, pp. 500–511. ISSN: 0302-9743. ISBN: 978-3-540-37334-6. DOI: 10.1007/11816508_50.

- Tristan Miller, Bertin Klein, and Elisabeth Wolf (2009). Exploiting Latent Semantic Relations in Highly Linked Hypertext for Information Retrieval in Wikis. In: *International Conference Recent Advanced in Natural Language Processing: Proceedings [RANLP]*. Ed. by Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov. Sept. 2009, pp. 241–245. ISSN: 1313-8502.
- Tristan Miller, Chris Biemann, Torsten Zesch, and Iryna Gurevych (2012). Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In: *Proceedings of COLING 2012: Technical Papers*. Ed. by Martin Kay and Christian Boitet. Dec. 2012, pp. 1781–1796.
- Tristan Miller, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych (2013). DKPro WSD: A Generalized UIMA-based Framework for Word Sense Disambiguation. In: *ACL 2013: 51st Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference: System Demonstrations*. Stroudsburg, PA: Association for Computational Linguistics, Aug. 2013, pp. 37–42.
- Michael Matuschek, Tristan Miller, and Iryna Gurevych (2014). A Language-independent Sense Clustering Approach for Enhanced WSD. In: *Proceedings of the 12th Edition of the KONVENS Conference*. Ed. by Josef Ruppenhofer and Gertrud Faaß. Universitätsverlag Hildesheim, Oct. 2014, pp. 11–21. ISBN: 978-3-934105-46-1.
- Tristan Miller and Iryna Gurevych (2014). WordNet–Wikipedia–Wiktionary: Construction of a Three-way Alignment. In: *LREC 2014, Ninth International Conference on Language Resources and Evaluation*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis. European Language Resources Association, May 2014, pp. 2094–2100. ISBN: 978-2-9517408-8-4.
- Tristan Miller, Darina Benikova, and Sallam Abualhaija (2015). GermEval 2015: LexSub – A Shared Task for German-language Lexical Substitution. In: *Proceedings of GermEval 2015: LexSub*. Sept. 2015, pp. 1–9.
- Tristan Miller and Iryna Gurevych (2015). Automatic Disambiguation of English Puns. In: *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: Proceedings of the Conference [ACL-IJCNLP]*. Vol. 1. Stroudsburg, PA: Association for Computational Linguistics, July 2015, pp. 719–729. ISBN: 978-1-941643-72-3.
- Christian F. Hempelmann and Tristan Miller (2016). Phonology and Taxonomies of Puns. In: *Handbook of Language and Humor*. Ed. by Salvatore Attardo. Routledge Handbooks in Linguistics. In preparation. New York, NY: Routledge.
- Tristan Miller, Mohamed Khemakhem, Richard Eckart de Castilho, and Iryna Gurevych (2016). Sense-annotating a Lexical Substitu-

tion Data Set with Ubyline. In: *LREC 2016, Tenth International Conference on Language Resources and Evaluation*. To appear. European Language Resources Association, May 2016.

Tristan Miller and Mladen Turković (2016). Towards the Automatic Detection and Identification of English Puns. In: *European Journal of Humour Research* 4(1) (Jan. 2016), pp. 59–75. ISSN: 2307-700X.

INDEX

- accuracy
 - clustering, *see* cluster purity
 - information retrieval, 96
 - word sense disambiguation,
see recall
- ACL Anthology, 2
- adjudication of sense annotations,
30, 86, 119–121, 144, 143–
146, 153–155
- advertising, 6, 76, 81, 132
- aggregate analysis engines, 102,
103
- AI-completeness, 2
- AIDA CoNLL-YAGO, 105
- alignment, *see* lexical-semantic
resources, alignment
- all-words disambiguation, 2, 23–
24, 30–32, 33, 46, 49, 51,
48–51, 52, 53, 104, 105
- alliteration, 80
- American National Corpus, 15
- analysis components, 99–106
- anaphora resolution, 40
- annotation time, 127, 133
- annotator bias, 26
- ant colony optimization, 55
- antonymy, 15, 40, 80
- Apache Software Foundation, 102
- automatic summarization, 57, 115
-
- BabelNet, 100, 110
- Bayesian networks, 21
- Biemann, Chris, 8
- bootstrapping, 16, 22
- British National Corpus, 15, 16,
66
- Brown Corpus, 86, 87
-
- C++, 102
- CAS, *see* common analysis struc-
tures
- CAS consumer, 102–105
- Chinese, 32
- chunking, 17, 18
- clustering
 - entropy, 67
 - purity, 67, 124–125
 - search results, 23, 32, 55
 - word senses, 6, 10, 31, 58, 61–
63, 67–71, 90, 108–109,
130
- cognitive science, 75
- Cohen’s κ , 26, 120–121
- CoInCo, 117, 123–126
- collection readers, 102–105
- collocation resources, 16, 21
- common analysis structures, 102,
104, 105
- compounds, 78, 79, 82, 119, 150
 - splitting, 17, 79, 119
- computational complexity, 2, 37,
38, 67
- computational humour, 75, 80,
132
- computational linguistics, 2, 8, 26,
75, 119, 129
- confusion matrices, 95, 95, 107
- conjoint alignments, 63–64, 66, 67,
73
- context, 13, 16–18, 20, 21, 38, 53,
55, 56, 87, 131
 - overlaps, *see* Lesk algorithms
- corpora, 4, 13, 105, *see also* data
sets
 - manually annotated, 4, 5, 7,
15, 18, 21–23, 29–30, 43,
51, 81, 99, 100, 111, 129
 - construction, 6, 21–22, 30,
35, 47, 84–86, 111, 118–
120, *see also* knowledge
acquisition bottleneck
 - puns, *see* puns, corpora
 - raw, 4, 15–16, 18, 19, 22, 35,
36, 40, 44, 49, 52, 129,
131
- coverage
 - data sets, 86, 121, 127, 130
 - lexical-semantic resources, 6,
7, 15, 16, 23, 57, 59, 66,
88, 122–123, 127
 - word sense disambiguation,
28, 47, 90
 - word sense induction, 22
- Crosbie’s *Dictionary of Puns*, 81
- cross-lingual wsd, 32
- crowdsourcing, 25, 117

- CSV, 107
- DANTE, 100, 110
- data sets, 24–28, 33, 99, 100, 105,
 see also corpora
- decision lists, tables, and trees, 21
- deep learning, 55
- definitions, *see* glosses
- degree centrality, 20, 49, 106
- deWaC, *see* wacky, deWaC
- DeWSD, 112, 113
- Dice coefficient, *see* interanno-
 tator agreement, raw,
 mean Dice
- dictionaries, 4, 13, 15, 16, 29, 58,
 111, 119, 135
 - bilingual, 1
 - machine-readable, 1, 3, 15,
 18, 35, 36, 43, 51, 61, 79,
 100, 129
 - pronunciation, 15, 79, 83
- digital humanities, 77, 132
- Dijkstra-wsa, 60–61, 64, 90
- distributional similarity, 5, 39–41,
 62, 129
- distributional thesauri, *see* the-
 sauri, distributional
- DKPro, 7, 103
 - DKPro Core, 103, 106
 - DKPro Lab, 103, 106
 - DKPro LSR, 60*n*, 103
 - DKPro Similarity, 103
 - DKPro Statistics, 103
 - DKPro TC, 132
 - DKPro WSD, 7, 10, 68, 92, 99–
 101, 104, 107, 108, 103–
 110, 130, 132
- domain disambiguation, 77, 80,
 98, 132
- domain-specific WSD, 2, 32, 33, 43
- Eckart de Castilho, Richard, 10
- eigenvector centrality, 20
- encyclopedias, 15, 58
- English, 4–7, 16, 30–32, 35, 43, 46,
 59, 60, 80, 105, 111, 112,
 115, 117, 119, 123, 124,
 131, 135, 141, 143
- English Internet Corpus, 116
- entity linking, *see* named entity
 linking
- Erbs, Nicolai, 10
- error propagation, 67, 73, 131
- Estonian, 104, 104, 105
- EuroWordNet, 105
- EVALITA 2009, 115, 115–116
- evaluation of word sense align-
 ments, 67–73
- evaluation of WSD, 33, 22–33
 - extrinsic (*in vivo*), 7, 22–23,
 32, 55, 111–112, 118, 126
 - for puns, 90–94, 130
 - intrinsic (*in vitro*), 7, 22–24,
 111–112, 118, 126
 - lower bounds, 29–30
 - metrics, 28–29, 47, 100
 - upper bounds, 25, 30
- exact match criterion, 28, 90
- example sentences, 15, 16, 38, 43,
 58–60, 138, 140, 145, 154
- F-score, 29, 47, 90, 96
- feghoots, 77
- FrameNet, 100, 106
- French, 30, 115, 116
- frWaC, *see* wacky, frWaC
- full alignments, 63–64, 67, 70, 73
- GAMBL, 70
- genetic algorithms, 55
- German, 4, 5, 7, 16, 111, 112, 115,
 116, 119, 122, 124, 126,
 127, 130, 147, 153
- GermaNet, 16, 106, 112, 113, 114,
 119–125, 127, 133, 153,
 154
- GermEval 2015, 10, 115, 115–117,
 118, 121
- GLASS, 7, 111–112, 113, 122, 123,
 125, 118–127, 130–132,
 157–161
- glosses, 6, 15, 16, 54, 58–60, 62,
 73, 131, 138, 145
 - overlaps, *see* Lesk algorithms
- glue code, 102
- graphs, 15, 19, 49, 60–61
 - connectivity, 20, 64, 65, 98,
 106
- Harrap's Easy English Dictionary*, 3
- Hector, 30, 33
- heuristics, 20–21, 39, 61, 62, 124
- holonymy, 16, 124
- homography, *see* puns, homogra-
 phy

- homonymy, *see* word senses, discreteness and granularity
- homophony, *see* puns, homophony
- HTML, 107, 108
- human-computer interaction, 76, 132
- humour, 6, 8, 76–78, 81
 - incongruity, 77, 80, 90, 132
- hyperlinks, 59, 61, 66
- hypernymy and hyponymy, 16, 39, 40, 119, 120, 122, 124, 125, 127, 131, 133, 149–150, 154
- I Can Sense It, 101
- IBM, 102
- IMS, 101, 106
- incongruity, *see* humour, incongruity
- information content, 19
- information extraction, 2, 115, 129
- information retrieval, 2, 28, 32, 40, 95, 125, 129
- informational gap, 4–6, 35–36, 38, 40, 41, 42, 43, 53–54, 71, 88, 93, 124, 129
- interannotator agreement, 24–28, 33, 62, 120–121, 143, 153
 - chance, 26
 - raw, 25–26, 30
 - average pairwise, 25
 - maximum, 25
 - mean Dice, 25
 - minimum, 25
- Italian, 30, 115, 116
- Iverson bracket, 11
- Japanese, 30, 80
- Java, 100, 102, 109
- JMWNL, 105
- jokes, *see* humour
- JWKTLL, 59, 64
- k-nearest neighbour classification, 21
- Kegel, Stan, 83
- Khemakhem, Mohamed, 10
- knowledge acquisition bottleneck, 3, 22, 35, 55, 57, 111, *see also* corpora, manually annotated, construction
- knowledge sources, 3–5, 13–18, 35, 129, *see also* lexical-semantic resources
- construction, 4
- contextual, *see* context
- external, 14–16
- knowledge-based wsd, 4–6, 18–22, 30–32, 35, 36, 40, 52, 54, 55, 71, 73, 89, 92, 97, 106, 129, 131, 132
- Koç University wsd system, 70
- Krippendorff's α , 27–28, 120
- language models, 13, 105
- latent Dirichlet allocation, 40, 49, 56, 131
- latent semantic analysis, 40, 41
- least-frequent first purification, 71, 70–71
- Leipzig Corpora Collection, 44
- lemmatization, 13, 17, 17–19, 37, 43, 79, 105, 106
- Lesk algorithms, 19, 36–39, 80
 - extended, 38–39, 106
 - lexically expanded, 48, 49, 51, 52, 53, 41–56, 72, 89, 92, 92–93, 106
 - original, 36–38, 106
 - simplified, 37–39, 41, 42, 43, 48, 49, 51, 52, 71, 72, 89, 92, 92–93, 106
 - simplified extended, 39, 43, 48, 49, 51, 53, 72, 89, 92, 94, 92–94
- lexical acquisition, 115
- lexical ambiguity, 1, 2, 4, 80, 81
 - intentional, *see* puns
- lexical chains, 20
- lexical expansions, 8, 36, 41–46, 48–56, 72, 129–131
- lexical gap, *see* informational gap
- lexical sample task, 2, 23–24, 30–32, 33, 36, 46, 51, 50–51, 105, 119, 130
- lexical substitution, 2, 7, 10, 23, 31, 41, 111, 115, 114–117, 127, 130, 133, *see also* word senses, relationship to lexical substitutes
- defined, 114–115

- evaluation, 115–117
- lexical-semantic resources, 4, 5, 18, 22, 29, 39, 51, 57–59, 79, 99, 111, 129
- alignment, 5–6, 8, 57–61, 90, 106, 129, 131
- coarsening, 6, 31, 62, 67, 130
- construction, 57
- lexicography, 3, 30, 40
- licensing of software and resources, 7, 100, 101, 113, 112–114, 116, 118
- log-likelihood test, 44
- lower bounds, *see* evaluation of wsd, lower bounds
- machine learning, 18, 21, 80, 98, *see also* supervised wsd
- machine translation, 1–2, 23, 30, 56, 57, 76–77, 129, 132
- market research, 76
- MASC, 100, 105
- MASI, 27, 86
- maximum entropy classification, 21
- meronymy, *see* holonymy
- Merriam-Webster's Collegiate Dictionary*, 3
- metonymy, 3
- Mihalcea, Rada, 47
- MII+ref, 49
- minimally supervised wsd, 22
- monosemous relatives, 22
- monosemy, 22, 61, 66, 105, 127
- most frequent sense baseline, 29–30, 43, 47, 49, 52, 54, 72, 92, 92–93, 106
 - adaptation to pun disambiguation, 91–92
- most-frequent first purification, 71, 70–71
- MuchMore, 112, 113, 120–121
- multiword expressions, 17, 58, 78, 79, 82, 122, 123
- named entity linking, 2, 10, 105, 107–109
- natural language generation, 75, 76, 81
- neural networks, 5, 21, 49, 55, 129
- NLTK, 103
- NP-completeness, 2
- OASIS, 102
- OmegaWiki, 90, 106
- one sense per collocation, 21, 39
- one sense per discourse, 20
- OntoNotes, 32, 33
- Open Mind Word Expert, 15
- Oxford Advanced Learner's Dictionary*, 38
- Oxford Dictionary of Current English*, 3
- Oxford Dictionary of English*, 62
- PageRank, 20, 106
- parameterization of wsd, 2, 46, 99, 102, 106–107
- paraphrasing, 41, 115
- parasets, 125, 124–126, 133
- paronomasia, *see* puns
- parsing, 13, 17, 18, 44, 79, 106
- part-of-speech tagging, 13, 14, 17, 19, 44, 79, 87, 89, 105, 106
- PHP, 84
- polysemy, 1–2, 62, 80, 87, 94, 105, 116, 120, 121, 122
 - systematic, 3*n*, 90
- precision, 21, 28–29, 47, 73, 90, 107, 115, 131
- preprocessing, 17, 19, 43, 79, 99, 101, 104, 106
- processing frameworks, 101–103
- Pun of the Day*, 82, 83
- Punnotator, 10, 84, 85, 84–85, 137, 139, 136–141
- puns, 4–6, 75–98, 135–141, 143
 - corpora, 81–88
 - definition and classification, 77–78
 - detection, 75–77, 79–82, 88, 97, 94–132
 - disambiguation, 6, 73, 75–81, 92, 94, 86–94, 97, 98, 130–132
 - generation, 75, 132
 - homography, 78–80, 83, 130, 132
 - homophony, 78
 - imperfect, 78–81, 132
 - location, 87–88, 94
 - phonology, 75, 77, 79, 81, 83, 132
 - syntax, 77, 80
 - translation, 76–78

- question answering, 57, 115
- random clustering baseline, 62, 63, 68–71, 73, 109
- random pun location baseline, 96
- random sense baseline, 29, 47*n*, 47, 48, 48, 49, 51, 52, 92, 92–94, 106
 - adaptation to pun disambiguation, 91
- readability assessment, 127, 133
- recall, 21, 29, 47, 90, 107, 115, 131
- replicability, 30
- Roget's Thesaurus*, 15
- SATANic, 84
- segmentation, 13, 17, 56, 106, 119
- selectional preferences, 18–19
- semantic concordances, *see* collocation resources
- semantic indexing, 56
- semantic networks, *see* wordnets
- semantic relatedness, 19–20, 40, 61, 62, 80
- semantic relations, 15, 16, 20, 39, 43, 49, 58, 59, 61, 123, 122–124, 131
- semantic roles, 54
 - labelling, 31
- semantic similarity, 40, 131
- SemCor, 15, 47, 72, 100, 105
- SemDis 2014, 115, 115–116
- SemEval, 15, 23, 31, 32, 82, 85, 86, 100, 101, 105
 - SemEval-2007, 31–32, 33, 46, 48, 49, 48–50, 51, 52, 53, 52–54, 108, 115, 115–116
 - SemEval-2010, 32, 33
 - SemEval-2013, 32, 55, 108, 109
 - SemEval-2015, 32
- semi-supervised WSD, 22, 35
- sense embeddings, 49
- sense inventories, 2, 13, 19, 22–24, 36, 40, 46, 55, 88, 99, 100, 105–106, 108, 119
- sense keys, 16, 31, 106
- SenseLearner, 70, 101
- SenseRelate::TargetWord, 101
- SENSEVAL, 15, 23, 82, 84–86, 100, 101, 105, 107, 111
 - SENSEVAL-1, 30, 33
 - SENSEVAL-2, 30–31, 33, 46–47, 51, 50–51, 104
 - SENSEVAL-3, 31, 33, 46–47, 51, 50–51, 62, 69, 70, 71, 71, 72, 105
- sentiment analysis, 76, 132
- Shakespeare, William, 77
- shared tasks, 33, 30–33, 86, 100, 115
- simulated annealing, 55
- spelling correction, 2
- Stamp, 84
- Stanford parser, 44
- Stanford POS tagger, 89
- stemming, 13, 17–18, 43
- stop lists, 37, 43
- supervised WSD, 4, 6, 15, 18, 20–24, 29, 31, 32, 35, 40, 43, 47, 51, 55, 58, 91, 98, 106, 129, 132
- support vector machines, 21
- synonymy, 15, 40, 60, 63, 66, 67, 116, 122, 127, 131
- synsets, 16, 59, 60, 61, 122, 124–125
- TüBa-D/Z, 113, 114, 120, 121
- TAC KBP, 105, 108–109
- targetted tagging, 24, 147–151
- text categorization, 115
- text similarity, 41, 56, 60, 73
- text simplification, 40, 115
- text watermarking, 115
- thesauri, 3, 13, 15–16, 40, 117
 - distributional, 5, 8, 36, 40–41, 42, 45, 43–46, 129, 131
- TKB-UO, 48
- tokenization, 13, 14, 17, 16–17, 19
- Tom Swifties, 77, 81
- topic signatures, 40
- TreeTagger, 105
- Turk Bootstrap Word Sense Inventory, 106, 117
- Turković, Mladen, 10
- TWSI, *see* Turk Bootstrap Word Sense Inventory
- UBY, 60*n*, 103, 106, 119
- Ubyline, 10, 119, 120, 127, 133, 147, 148, 149, 147–151
- UIMA, 7, 100, 102–106, 109, 130
- UKB, 101
- underspecification, 3*n*

- unsupervised wsd, *see* word sense induction
- upper bounds, *see* evaluation of wsd, upper bounds
- Usenet, 16
- VerbNet, 100, 106
- visualization, 100, 106, 107
- Wacky, 15, 16
 - deWaC, 112, 116
 - frWaC, 116
- Web 1T, 16
- WebAnno, 84
- WebCAGE, 100, 105, 112, 113
- WikiCAGE, 113, 112–114
- Wikipedia, 49, 57, 61, 59–61, 64, 66*n*, 66, 66, 68, 71, 73, 90, 100, 106, 112, 116, 118, 129, 131
- Wiktionary, 15, 57, 59–60, 61, 64, 66, 66, 68, 71, 73, 90, 100, 106, 119, 129, 131
- WN-Map, 70, 106
- word embeddings, 5, 16, 40, 41, 49, 55, 129
- word sense induction, 2, 10, 16, 22, 31, 32, 40, 55, 108–109
- word senses
 - alignment, *see* lexical-semantic resources, alignment
 - clustering, *see* clustering, word senses
 - discreteness and granularity, 3–6, 22, 24, 26, 46, 58, 79, 87, 88, 129, 140, 143
 - distribution, 23, 26, 55, 66, 66
 - frequency, 5, 21, 29, 36
 - relationship to lexical substitutes, 122–126, 133
- word sketches, 40
- WordNet, 15–16, 20, 29–31, 33, 39, 43, 46, 47, 49, 61, 57–64, 66, 66–68, 70, 71, 73, 82, 85–87, 89*n*, 89–91, 105, 106, 108, 117, 119, 123, 124, 129, 131, 138, 143, 145
- WN++-DC, 105
- wordnets, 4, 15, 35, 43, 58, 129
- World Wide Web, 2, 16, 81, 83, 84, 101, 112
- XML, 85, 85, 104, 107, 117, 118, 118–119, 149, 151
- Zesch, Torsten, 10
- Zipf's law of meaning, 1, 66
- Zorn, Hans-Peter, 10

ERKLÄRUNG ZUR DISSERTATION

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 22. März 2016

Tristan Miller

COLOPHON

This document was typeset in \LaTeX using modified versions of the ClassicThesis and ArsClassica packages. Titles are set in Iwona, math is set in Euler, monospaced text is set in Bera Mono, and the main text is set in Adobe Palatino. The design is inspired by *The Elements of Typographic Style* by Robert Bringhurst.

