Technische
Universität
Berlin

berlin

Department of Software Engineering and
Theoretical Computer Science

Master's Thesis

Learning semantic relations with distributional similarity

Priska Herger

September 23, 2014

Advisors:

Prof. Dr. Chris Biemann
Language Technology Group
Technische Universität Darmstadt
Germany

Prof. Dr. rer. nat. Volker Markl
Database Systems and Information Management Group
Technische Universität Berlin
Germany

Abstract

Learning to detect and distinguish semantic relations is an important building block in the formalization of natural language, in tandem with entity recognition. Distributional semantics has taken us a big step in that direction by providing a methodological and theoretical framework in which we can distill synsets, which are sets of words that share attributional or relational similarity, although under a variety of different semantic respects. In this thesis I examine an approach to further discriminate among different types of relational similarity found in such synsets. Therefore, pairs of distributionally similar nouns are represented by their shared contexts within sentences from a gigaword-sized corpus. That is, words which are commonly examined in a paradigmatic manner are now investigated in terms of their syntagmatic properties. From the contexts extracted when such words co-occur, three sets of features and their set-theoretic combinations are derived: context features, similarity features, and features from topic modeling. A logistic regression model and the Chinese Whispers clustering algorithm are trained to distinguish hypernyms, co-hyponyms, and meronyms. The resulting models are evaluated with the BLESS and SAT data sets. The feature sets are analyzed in an ablation test, which shows that subtree patterns along dependency parses containing both nouns in a pair are the single most predictive feature. Predictions are best for co-hypernyms with an F-score of 0.90, slightly less reliable but still fairly good for meronyms with 0.79, and most difficult for hypernyms with 0.56. The clustering returns largely pure clusters for hypernymy and co-hyponymy, while it fails to form meronymy clusters. Thorough analysis of the BLESS data set shows that de-lexicalization before stratification is essential to prevent overfitting. Overall, the approach achieves good results, which might be improved further in future work by integrating multiword expressions.

Zusammenfassung

Das Erkennen und Unterscheiden von semantischen Relationen ist ein elementarer Baustein der Formalisierung von natürlicher Sprache, in Kombination mit der Eigennamenerkennung. Die distributionale Semantik bringt uns diesem Ziel einen großen Schritt näher, indem sie eine methodische und theoretische Einbettung zur Verfügung stellt, um semantisch ähnliche Worte zu finden und zu gruppieren. In dieser Arbeit untersuche ich einen Ansatz diese Mengen von ähnlichen Worten genauer dem Typ ihrer semantischen Relation nach zu unterscheiden. Zu diesem Zwecke werden Paare von distributional ähnlichen Worten durch ihren gemeinsamen Kontext in Sätzen eines gigawort-großen Korpus repräsentiert. Es werden also Worte, die üblicherweise in einem paradigmatischen Verhältnis zueinander stehen, bezüglich ihrer syntagmatischen Eigenschaften untersucht. Aus den extrahierten Kontexten werden dreierlei Merkmalsmengen und deren mengentheoretische Kombinationen gewonnen: Kontextmerkmale, Ähnlichkeitsmerkmale und thematische Merkmale. Ein logarithmisches Regressionsmodell und der Chinese Whispers Clustering-Algorithmus werden darauf trainiert Hypernyme, Co-Hyponyme und Meronyme zu unterscheiden. Die trainierten Modelle werden anhand der BLESS- und SAT-Daten evaluiert. Eine Analyse der Merkmalsmengen zeigt, dass das Merkmal, welches auf Teilbäumen von dependenzgrammatikalischen Syntaxbäumen basiert, die Klasse des Nomenpaares am zuverlässigsten vorhersagt. Insgesamt sind die Vorhersagen am besten für Co-Hyponyme mit einem F-Maß-Wert von 0.90, etwas weniger zuverlässig aber immer noch gut für Meronyme mit 0.79 und am schwierigsten für Hypernyme mit 0.59. Der Clustering-Algorithmus findet reine Gruppen von Hypernymen und Co-Hyponymen, allerdings keine guten Meronymgruppen. Eine gründliche Analyse der BLESS-Daten zeigt, dass es essentiell ist die Trainings- und Testsdaten zu delexikalisieren, um Überanpassung zu vermeiden. Insgesamt ergibt der untersuchte Ansatz gute Ergebnisse, denen die Betrachtung von Mehrwortbegriffen noch zuträglich sein könnte.

Acknowledgements

I would like to thank Alan for being my advisor on behalf of Prof. Markl, Steffen for writing and sharing *simsets*[1], Eugen for the overlap-based similarity space, Johannes for answering many Hadoop-related questions, Jochen for sending rigorous scientific papers my way when I was about to lose faith in academia, Paula for many candid discussions, Dominik for encouraging and Joscha for sobering comments, PoC||GTFO[2] for introducing me to polyglots, and most of all Chris for being a cognitive scientist at heart and for being my advisor!

[1] HTTPS://GITHUB.COM/POLICECAR/SIMSETS

[2] HTTPS://ARCHIVE.ORG/DETAILS/POCORGTFO01

Contents

1 Introduction

1.1 Motivation

Bootstrapping semantics from text is as challenging as it would be scientifically rewarding. Early attempts at computer programs that would understand natural language were published within a decade of the first international conference on natural language processing in 1952; which was also the time when John McCarthy first coined the term *artificial intelligence*. These programs included STUDENT by Daniel Bobrow, which solved algebra word problems, and ELIZA by Joseph Weizenbaum, an interactive program simulating a Rogerian psychotherapist. They were followed shortly after by Terry Winograd's SHRDLU and many more. While these early programs were very encouraging and impressive at the time, they were also restricted in terms of possible input and domain of conversation.

Since the early days of natural language processing, huge advances have been made in formal theories, the quantities of data available, performance and disposability of computing resources, and by combining methods from diverse disciplines. Big successes have be registered from machine translation, which is on hand now in many smartphone applications, to natural language user interfaces that answer (some of) your questions and semantic web technologies and resources.

Machine reading, as Oren Etzioni termed the interdisciplinary take on autonomous understanding of text by machines recently (ETZIONI ET AL. [2006]), could soon bring about a natural language interface to all digitally stored textual knowledge and with it a whole series of immediate applications: automatic lexical inference and remedies to data sparseness being just two of the more obvious ones.

One of the prevalent semantic theories[1] in this context is *distributional semantics*, which combines linguistic methods with statistics. It aims at approximating semantics with the distributions of terms in text collections on the assumption that words with similar distributions have similar meanings. Distributional semantics is a good match for machine reading as they're both based on inherently unsupervised methodology. *Statistical semantics*, as it is also known, caught on because it provides both large-scale coverage and "quantitative predictions about degrees of similarity (or relatedness)" while requiring little manual supervision (BARONI [2013]). Statistical semantics has shown promise in various problem fields of linguistic, with its models outperforming formal lexical representations, like semantic networks, in a variety of applications from word sense discrimination to selectional preferences and synonym detection (SCHÜTZE [1998], ERK ET AL. [2010], PADÓ AND LAPATA [2007]).

Assertions about degrees of relatedness, however, do not suffice to make text palatable to inference and reasoning, which are integral building blocks of semantic machines (ETZIONI ET AL.

[1] Semantic theories are theories that deal with assigning meaning to linguistic expressions.

[2006]). Much rather our models need to be able to discriminate among different types of semantic relations that display similar distributional behavior but under a variety of different semantic respects; for instance, near-synonymy, co-hyponymy, meronymy, holonymy, hypernymy, and antonymy. Learning and predicting such relations reliably, would form an important building block on the path to machine reading, in that components like entity resolution and textual entailment would benefit greatly.

1.2 Hypothesis

The aim of this work is to examine whether information derived from distributional similarity is sufficient to learn specific semantic relations. This is no start from scratch – many approaches already provide mechanisms to obtain groups of distributionally similar words (e.g., Lin [1998b], Rapp [2004]). A distributional thesaurus is but one (prominent) example here (cf. Biemann and Riedl [2013]). I build on their work to examine whether it is possible to further separate the sets of distributionally similar words provided by such a mechanism into more refined groups that share a specific semantic relation using more distributional similarity. One could think of it as higher-level distributional similarity.

The basic idea is as follows: In distributional semantics, the meaning of a word is derived from its distributional properties in a corpus compared to those of other words in that corpus. This means that we look at what other words a word co-occurs with (its syntagmatic relations) and what words occur with the same other words but not usually with each other (its paradigmatic relations). Table 1.1 illustrates the setting. In the given examples *coffee* and *drink* or *they* and *gulp* are in a syntagmatic relation, whereas *coffee* and *cocoa* are in a paradigmatic relation.

I	drink	coffee
you	sip	tea
they	gulp	cocoa

I	drink	coffee
you	sip	tea
they	gulp	cocoa

Table 1.1: Syntagmatic (left) versus paradigmatic relations; from Sahlgren [2012].

Words that are in a paradigmatic relation, i.e. co-occur with the same *other* words but usually not with each other, are considered distributionally similar and hence similar in meaning. Their meaning, however, is often similar along a broad definition of similarity which includes various relations like near-synonymy, hypernymy, or even antonymy. Every so often though, such paradigmatic words, like *tea* and *coffee*, co-occur in the same sentences, for instance, in enumerations and definitions.

If we now represent such pairs of distributionally similar words only by information extracted from sentences in which they co-occur, and reapply methods of distributional semantics, will pairs with high similarity in the resulting model share the same semantic relation? Put differently: If we examine both paradigmatic and syntagmatic relations between pairs of words, might that suffice to turn up the information required to distinguish the semantic relations that exist between these words?

1.3 Terminology

The terminology used in this thesis is generally in line with the terminology outlined in Medelyan et al. [2013], which corresponds to common practice in linguistics. Since usage is less consistent across related fields like data mining, natural language processing (NLP), or knowledge-engineering, I will take a moment to review some basic terms.

The smallest unit of language that carries semantic meaning is called a *morpheme*. There are *free* and *bound* morphemes, and free morphemes are *words*. Words undergo morphological processes that may lead to variation in meaning, word class, tense, number, plurality etc. – s. Manning and Schütze [1999] for a detailed account. One such process is *inflection* which yields forms like *makes* or *made* from the root *make*. The inflectional forms of a word are often subsumed into a *lexeme*. Inversely, one could say, that a lexeme is the result of applying *stemming* or *lemmatization* to an inflected word. According to Manning et al. [2008], stemming and lemmatization differ in that stemming is more of "a crude heuristic process that chops off the ends of words" while lemmatization uses vocabulary and morphological analysis "aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma".

Another morphological process is *compounding*: Words can be joined into new words (e.g., *home* + *school* → *homeschooling*) or multiword expressions (e.g. *ham radio* but also idioms, like *to break the news*) and are then called *compound words*. Further terms to consider are *concept* and *term*. Medelyan et al. [2013] define concepts as representing "classes of things, entities, or ideas, whose individual members are called *instances*" and terms as "words or phrases that denote, or name, concepts". Further terminology, like *semantic relations*, will be explained when used in text.

1.4 Outline

With the hypothesis pinned down and some basic terminology untangled, the next chapter gives a review of the theoretical underpinnings of this work. There I trace distributional semantics back to its beginnings and motivate its relevance to present-day research; I delineate the many shades of semantic similarity, the second pillar of this work, and give an overview of related research. Chapter 3 contains detailed descriptions of the applied methods, while at the same time placing them in the bigger picture of the fields they are taken from. Evaluation sets and procedures are depicted in that chapter as well. The results reached are presented and discussed in Chapter 5. Finally, the last chapter gives conclusions drawn and an outlook on future work.

2 Background

This chapter contains the theoretical background of distributional similarity and provides an overview of research related to this study. I begin with the big picture, briefly considering the prevalent semantic theories of the last century up until now, to then focus on distributional semantics, the distributional hypothesis, and distributional semantic models including their applications and challenges. In the second part I inquire into semantic similarity and delineate different levels of analysis together with respective experiments found in the literature. I conclude with further related research that fits none of the above categories but is no less relevant to this work.

2.1 Distributional semantics

There are three long-standing schools of theorizing the science of linguistics: the Externalists, the Emergentists and the Essentialists, Scholz et al. [2014]. It would by far exceed the scope of this thesis to discuss them in any detail, but I want to quickly mention their main tendencies to highlight the backdrop distributional semantics arises from. The Externalists might as well have been called 'structural descriptivists' or 'empiricists' as they are especially concerned with building models able to predict the structure of natural language expressions in accord with empirical data collected from language use. The Emergentists, on the other hand, "aim to explain the capacity for language in terms of non-linguistic human capacities: thinking, communicating, and interacting", Scholz et al. [2014]. One field of study close to the essence of this view is the impact of social status on linguistic structure. Thirdly, the Essentialists, who are sometimes referred to as 'formalists', center on abstract universal principles of language from which to derive properties of specific languages. Noam Chomsky's *universal grammar* (Chomsky [1965]) is the most prominent example here[1]. Distributional semantics with its focus on predicting similarity in meaning from distributions of terms in corpora and its proximity to corpus linguistics mainly rests upon the Externalists' empirical outlook.

In order to understand and analyze the essence of language, the question about the nature of meaning and reference of linguistics expressions is as important as the one about the subject of linguistics touched upon above. Philosophy of language is the branch of philosophy traditionally concerned with the topics of meaning and reference. Similarly to the philosophy of linguistics, it can only be nodded to in passing here. There are two sorts of theories of meaning: *semantic theories*, or semantics for short[2], and *foundational theories of meaning*, Speaks [2014]. We will only deal with the former sort here, which is concerned with assigning semantic contents to the expressions of a language. In contrast, foundational theories of meaning describe the psychological and sociological givens that lead to linguistic expressions carrying the meaning they do for a particular person or population.

[1] Refer to Scholz et al. [2014] and the sources cited therein for a more comprehensive account.

[2] Remember, the etymological derivation of *semantics* includes the Greek *σημαντικός* :: significant, and *σημαινειν* :: to show by sign, to signify, to point out.

Following LENCI [2008], there are three prevalent semantic theories in the last century: the conceptualist view, formal model-theoretic semantics, and squeezed in between, more of a methodological approach, distributional semantics. The conceptualist or cognitive stance views the meaning of linguistic expressions as that which is evoked by cognitive principles or mental experiences. In LENCI [2008]'s words: "the emphasis of cognitive semantics is on an intrinsically embodied conceptual representation of aspects of the world, grounded in action and perception systems.". Formal model-theoretic semantics, on the other hand, takes a denotational stance which conceives of linguistic expressions as references to propositions and objects in the world which can be mapped to truth values. This approach is often formalized using the λ-calculus, cf. HEIM AND KRATZER [1998].

Lastly, distributional semantics argues for a usage-based account of meaning in agreement with the late Wittgenstein who said that "the meaning of a word is its use in the language" (WITTGENSTEIN [1953]), and "if we had to name anything which is the life of the sign [term], we should have to say that it was its use", WITTGENSTEIN [1958]. Or as the psychologists MILLER AND CHARLES [1991] would later formulate: "What people know when they know a word is [...] how to use it (when to produce it and how to understand it) in everyday discourse". Distributional semantics operationalizes this notion by approximating semantic similarity with the distributional properties of words and phrases in corpora.

2.1.1 Distributional hypothesis

Distributional semantics is based on the *Distributional Hypothesis* (DH) which is often stated in one of the following ways (see, e.g., TURNEY ET AL. [2010], LENCI [2008]):

> "Words that occur in similar contexts tend to have similar meanings.", attrib. Z. Harris
> "You shall know a word by the company it keeps.", attrib. J.R. Firth

BARONI AND LENCI [2010] are a bit more specific: "[T]he degree of semantic similarity between two words (or other linguistic units) can be modeled as a function of the degree of overlap among their linguistic contexts.". SAHLGREN [2008] turns the wording around saying that "differences of meaning correlate with differences of distribution". All of the above formulations share two crucial aspects: there are differences /similarities in meaning between terms and they can be captured by comparing the linguistic contexts in which these words occur. However, as SAHLGREN [2008] points out as well, none of these statements specify "what kind of distributional information we should look for, nor what kind of meaning differences it mediates.". I will come back to these issues in Sections 2.1.2 and 2.2 when taking a closer look at distributional semantic models and semantic relations in general. For now, may the following example further illustrate the general idea. Consider the sentences:

> "He filled the wampimuk with the substance, passed it around and we all drunk some."
> "We found a little, hairy wampimuk sleeping behind the trees."[3]

The linguistic structures, in which the fictional word *wampimuk* appears, give important clues about its meaning. In the first example, *wampimuk* seems to refer to a container that one can fill liquid into. In the second case, more likely a kind of animal. The point being that language

[3] Unfortunately, I could not trace down the original source of these sentences, but I first found them at HTTP://PARLES.UPF.EDU/LLOCS/GLIF/HTM/ACTIV/BARONI1.PDF.

itself captures important aspects of the meaning of terms which the distributional hypothesis and respective models take advantage of when modeling meaning.

Lenci [2008] distinguishes two versions of the distributional hypothesis. The weak DH as a quantitative method for semantic analysis and lexical resource induction useful for identifying semantically similar terms through inspection of their distributional contexts. And the strong DH which is a cognitive hypothesis about the form and origin of semantic representations and the constitutive role of word distributions therein. In this study I will deal only with the weak distributional hypothesis, leaving out aspects about possible representations of meaning in cognitive systems.

2.1.2 Distributional semantic models

According to Baroni et al. [2014], *Distributional Semantic Models* (DSMs) are the most straightforward implementation of the distributional hypothesis in computational linguistics. In DSMs each term – word, word pair, or phrase – is represented as a collection of context features. Vectors and matrices are frequently used as mathematical objects for handling such collections; not least because efficient computational tools exist to calculate similarities on them. DSMs using vectors are also known as *Vector Space Models* (VSMs), where each vector represents a term in the vector space.

The kind of matrix used varies with the type of similarity to discover: among the most common ones are document similarity, word similarity – which is also called attributional similarity, and relational similarity, cf. Turney et al. [2010]. The respective matrices are called term–document matrix, word–context matrix, and pair–pattern matrix. However, these three kinds of matrices do not exhaust the possibilities. Various researchers have used higher-order tensors[4]; for instance, term–document–language tensors to model multilingual document similarity (Chew et al. [2007]) and verb–subject–object tensors to induce selectional preferences of verbs (Van de Cruys [2010]).

Recent work in the machine learning community has brought forth a different mechanism, where word representations embedded in neural networks are used to capture relational similarities, which can then be recovered using vector arithmetic, e.g., Bengio et al. [2006], Mikolov et al. [2013]. Levy and Goldberg [2014] subsequently showed that these embedded word representations are functionally equivalent to the sparse representations of distributional semantic models.

DSMs differ by the terms examined (words, multiword expressions, pairs etc.), the contexts considered (context type, context window), the interpretation of co-occurrence (frequency weighing), the measure of dimensionality reduction, and the measure of similarity employed. They can be formalized as tuples $< T, C, R, W, M, d, S >$ of target words T represented with contexts C, a relation R between words and contexts, a weighing scheme for contexts W, a distributional matrix M: $T \times C$, a dimensionality reduction function $d : M \rightarrow M'$, and a distance measure S between the vectors in M' – see the tutorial Stefan Evert gave at the NAACL-HLT 2010[5]. Typical repre-

[4] “An nth-rank tensor in m-dimensional space is a mathematical object that has n indices and m^n components and obeys certain transformation rules” like the dot product, cross product, and linear maps; cf. HTTP://MATHWORLD.WOLFRAM.COM/TENSOR.HTML.

[5] HTTP://WORDSPACE.COLLOCATIONS.DE/DOKU.PHP/COURSE:ACL2010:START

sentatives of distributional semantic models include *Latent Semantic Analysis* (Landauer and Dumais [1997]) and *Hyperspace Analogue to Language* (Lund and Burgess [1996]), which are defined by a specific choice for the above mentioned parameters. In case of LSA and HAL it is the use of word regions (e.g., documents) and word windows of immediately adjacent words as contexts, respectively.

The imprecise formulation of the distributional hypothesis has led to great variation in the design choices for each of the model parameters mentioned above. Exemplarily, find the coarse categories for the parameters *context* and *weighing scheme* in the following, and refer to Sections 3.2.1 and 3.2.2 for more details. The more widely used contexts can be grouped in the following way:

- linear contexts, in particular word windows or windows of larger linguistic units; for example, in Rapp [2003]
- syntactic dependencies with various types of dependencies and lengths of paths; see, e.g., Lin [1998b], Padó and Lapata [2007]
- lexico-syntactic patterns; for instance, Hearst [1992]

Weighing schemes transform term representations from raw counts to log-frequency space such as to smoothen high frequency differences and /or give more weight to contexts that significantly associate with a target word. Popular measures include *mutual information*, *log-likelihood ratio*, and *term frequency-inverse document frequency* (tf-idf) (see, e.g., Church and Hanks [1990], Dunning [1993], Salton and Buckley [1988]). Even from these brief enumerations, it becomes obvious that variation abounds when talking about DSMs. We will see in the next paragraphs that work is underway towards a more unified model.

2.1.3 Applications and Challenges

A wide selection of linguistic as well as cognitive phenomena have successfully been modeled with distributional similarity and distributional semantic models. On the side of computational linguistics these include word sense discrimination (Schütze [1998]), single-word translation (Sahlgren and Karlgren [2005]), phrase similarity (Clark [2012]), synonym detection (Padó and Lapata [2007]), and selectional preferences (Erk et al. [2010]). Simulations of cognitive phenomena range from semantic priming (Lund and Burgess [1996]) and word association norms (Griffiths et al. [2007]) to vocabulary acquisition (Landauer and Dumais [1997]).

Despite these successes, "no single distributional semantic model meets all requirements posed by formal semantics or linguistic theory, nor do they cater for all the aspects of meaning that are important to philosophers or cognitive scientists." – as the motivational text of a recent seminar on "Computational Models of Language Meaning in Context" at the Leibniz-Zentrum für Informatik[6] ascertained. According to these researchers, which include Alessandro Lenci, these requirements encompass at least the following:

- an account for linguistic compositionality
- robust first-order models of inference, and
- integrating DSMs into a broader model theoretic framework

[6] cf. http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=13462

Further shortcomings include the *symbol grounding problem*, a general proposition by Harnad [1990] which stresses that purely symbolic models are disconnected from the referents of their symbols which reside in the world outside the model – a limitation one could, for instance, overcome by integrating perceptions. In other words, according to this objection, distributional semantics can only ever provide relative or model-internal assertions of similarity since it defines the meaning of a word entirely in terms of its relations to other words. Or as Lenci [2008] put it: "both cognitive approaches and model-theoretic ones agree on refusing distributional semantics because meaning can not be explained in terms of language-internal word distributions, but needs to be anchored onto extra-linguistic entities, being them either conceptual representations in the speakers' mind or objects in the world." See Bruni et al. [2014] for a comprehensive account of this objection.

Despite these seemingly fierce limitations, it would be too early to dismiss distributional semantics as a semantic theory altogether. Recent research is tackling most of the submitted contentions with promising results. Turney [2008] started working towards a unified model in which a range of semantic phenomena like synonymy, antonymy, associations, and analogies are consolidated in a semantic model of relational similarity. Baroni et al. [2009] proposed a general semantic model, called *semantic memory*, from which task-specific semantic spaces can be extracted on demand. Baroni et al. [2014] use function application from formal semantics to capture compositionality in terms of a syntax-driven calculus. Mitchell and Lapata [2010], in turn, suggest pairwise additive and multiplicative vector mixtures as compositional operations. Bruni et al. [2012] and Bruni et al. [2014] combine standard DSMs with image analysis to form multimodal distributional semantic models in which semantic word representations are enriched with low-level visual features. Last but not least, Turney and Mohammad [2013] showed that lexical entailment can be conceived of as a relation modeled in terms of similarity differences over word-context matrices.

This short overview gave an impression of the open challenges in the field of distributional semantics and pointed to some of the manifold approaches to resolving them currently on the anvil. In the following section I will take a closer look at the semantic relations being modeled in DSMs and will show how the ones used in this study fit in.

2.2 Semantic similarity

As explained in Section 2.1, distributional semantic models interpret distributional similarity as semantic similarity. If two vector representations $\vec{a}$ and $\vec{b}$ of words a and b are closer to each other than the vectors $\vec{a}$ and $\vec{c}$, then, it is presumed, word a is semantically more similar to b than to c. The types of semantic similarity, however, that exist between distributionally similar words vary widely and so do the attempts at categorizing them. Semantic similarity and semantic similarity measurement is a broad field researched in disciplines as diverse as linguistics, computational linguistics, artificial intelligence, computer science, psychology, philosophy, cognitive neuroscience, psycholinguistics, and mathematics. Measures of semantic similarity reach from geometric conceptual spaces (Gärdenfors [2004]) to set-theoretic feature matching models (Tversky [1977]), from distance measurement on graphs (Rada et al. [1989]) to information content-based similarity (Resnik [1995]), and from similarity as analogy (Gentner and Markman [1997]) to transformational similarity (Hahn et al. [2003]); just to name a few. It is impossible to cover all of these approaches here and I will not attempt to. Please refer to the papers mentioned and for a more general discussion from a linguistic viewpoint to Murphy [2003] as starting points for further investigation. Instead I present two coarse classifications of semantic similarity: the one put forward by Turney [2006] and another which goes back to de Saussure [1916], as found in a survey by Khoo and Na [2006]. Subsequently, I present the semantic relations most relevant to this study and discuss various related research with a focus on relation discovery.

Turney [2006] distinguishes two types of semantic similarity:

- attributional similarity
- relational similarity

He uses the terms *attributional* and *relational* as defined in Medin et al. [1990]: "Attributes are predicates taking one argument (e.g., *X is red*, *X is large*), whereas relations are predicates taking two or more arguments (e.g., *X collides with Y*, *X is larger than Y*). Attributes are used to state properties of objects; relations express relations between objects or propositions.". Therefore, the degree of *attributional similarity* between two words depends on the correspondence of attributes between the words and the degree of *relational similarity* between two pairs of words depends on the correspondence of relations between the word pairs. Mapping this to DSMs would mean that models representing single words are useful for measuring attributional similarity and those representing pairs of words quantify relational similarity.

But what are the concrete relations that fall into one category or the other? Attributional similarity, sometimes also called *semantic relatedness* (cf. Budanitsky and Hirst [2001]) includes relations such as synonymy (*eye doctor*, *oculist*), antonymy (*odd*, *even*) as well as functional relationships (*candle*, *match*) or frequent associations (*quokka*, *cuteness*). Relational similarity, on the other hand, comprises relations like *meronymy* (the *part–of* relation), *hypernymy* (the *is–a* relation), and *co-hyponymy* (the *shared–hypernym* relation).

In contrast, Khoo and Na [2006] – referring to de Saussure [1916] – group semantic relations into syntagmatic and paradigmatic relations. Syntagmatic relations are relations between words that co-occur in the same sentence whereas paradigmatic relations are relations between words that can occur in the same position in a sentence. Referring back to Table 1.1, *coffee* and *cocoa* are in paradigmatic relation while *sip* and *tea* are in a syntagmatic relation with each other. Examples of paradigmatic relations include hypernymy and hyponymy (its inverse), meronymy and holonymy (the *whole-of* relation), synonymy, antonymy, and troponymy (the relation of *manner* between verbs, e.g., *drink – gulp*). Examples of syntagmatic relations are case relations (*agent* vs. *patient*, e.g., in *Aino lost face*, *Aino* is the agent, *face* the patient), and associations (*kick the bucket*), Khoo and Na [2006].

Distributional semantic models often span several of these kinds of semantic relations or are continuous models which can be applied to tasks involving specific relations using different similarity measures. In the following paragraphs I describe related work dealing with semantic relations on various levels including

- automatic pattern extraction
- pattern-based and cluster-based relation discovery
- DSMs using bless for evaluation.

Automatic discovery of semantic relations The automatic discovery of semantic relations was pioneered by Marti A. Hearst, who used hand-crafted patterns to automatically extract hyponym relations from text, Hearst [1992]. Although more recent research additionally automates the pattern generation process, these now-called *Hearst patterns* are still frequently used as a litmus test to such generators; cf., e.g., Snow et al. [2004]. See Table 2.1 for examples of Hearst patterns including text excerpts to which they apply. Much research has been done on relation harvesting in the wake of Hearst [1992]. The following overview groups them by the kinds of relations considered with a particular focus on publications dealing with relations between nominals as these are the most relevant to this thesis.

Relation	Example
NP$_0$ such as {NP$_1$, NP$_2$ ..., (and | or)} NP$_n$	"The bow lute, such as the Bambara ndang, ..."
such NP as {NP ,} {(or | and)} NP*	"... works by such authors as Herrick, Goldsmith, and Shakespeare."
NP {, NP} {,} or other NP*	"Bruises, wounds, broken bones or other injuries ..."
NP {, NP} {,} and other NP*	"... temples, treasuries, and other important civic buildings."
NP {,} including {NP ,} {or | and} NP*	"All common-law countries, including Canada and England ..."
NP {,} especially {NP ,} {or | and} NP*	"... most European countries, especially France, England, and Spain."

Table 2.1: Hearst patterns with text snippets; from Hearst [1992].

There are the studies that extract a single broad semantic relation at a time – most frequently hypernymy. Snow et al. [2004], for instance, classify unseen noun pairs with a pre-computed list of dependency patterns that they tested to be good predictors of hypernymy and reach an F-score of 0.36. Ritter et al. [2009] built a hypernym finder for arbitrary proper nouns based on the frequency with which noun pairs were seen in one of a range of Hearst patterns. They approximate recall by the percentage of nouns they can predict one or more good hypernyms for, and produce a precision value of about 0.63 for 0.65 percentage coverage (numbers read out from figure).

Another branch of investigation deals with extracting the seven relations specified in Task 4 of SemEval-2007, a workshop on semantic evaluations. Table 2.2 enumerates these relations including example instances. Girju et al. [2007] describes the task setting and gives a summary of the competition results. F-scores between 0.68 and 0.82 were measured depending on the relation, with *content–container* scoring the highest precision at 0.93. Unfortunately, original papers of the different groups are not cited, making it hard to capture the details of their respective systems. According to Girju et al. [2007], the team that generally reached the highest scores extracted lexico-syntactic patterns from semantic parses and used WordNet[7] sense labels in the data sets.

[7] HTTP://WORDNET.PRINCETON.EDU/

Relation	Example
cause - effect	laugh - wrinkles
instrument - agency	laser - printer
product - producer	honey - bee
origin - entity	message - from outer space
theme - tool	news - conference
part - whole	the door - of the car
content - container	the apples - in the basket

Table 2.2: Relations from Task 4, SemEval 2007; examples from GIRJU ET AL. [2007].

Further work considers narrow relations like *acquirer–acquiree*, *person–birthplace*, *company–headquarters*. BOLLEGALA ET AL. [2009], for instance, automatically extract shallow lexical patterns for word pairs, cluster these patterns into groups considered representative of semantic relations, and measure the similarity between these clusters with an information theoretic metric. Their system is evaluated with a classification task and using the SAT analogy data set[8]. They report an average precision of 0.74 and accuracy of 0.93 averaged over all five relations used. Thereby values vary between 0.37 for *person–birthplace* and 0.96 for *CEO–company.* Their SAT score peaks at 51.1 percent which compares favorably to the human score of 57 percent but is still topped by TURNEY [2006] with 56.1 percent. The core idea behind TURNEY [2006]'s system, which he calls *latent relational analysis*, is to extract patterns between word pairs and near-synonym variants of these word pairs collected from a thesaurus of synonyms. The observed patterns are enhanced by producing variants in which some words are replaced with wild cards. The resulting pair–pattern matrix is condensed using *singular value decomposition.* To solve the analogy questions, the relational similarity between any two word pairs can be calculated as the average Cosine between the word pairs in question and their near-synonym alternate pairs. Subsequently, the pair most similar to the question pair is chosen for the answer. Note that TURNEY [2006] does not group word pairs into semantic relations but works on the level of semantic similarity without discretization.

PANTEL AND PENNACCHIOTTI [2006] present a generic system capable of extracting various (binary) relations, including hypernymy and meronymy but also more specific relations such as *reaction* and *succession.* They use seed terms to extract patterns from the World Wide Web. For instance, given the seed pair *wheat* and *crop* for the relation hypernymy, they search for sentences containing these seed terms and extract as pattern the tokens between the two terms. The retrieved patterns are generalized, ranked, and pruned with a custom measure of association based on point-wise mutual information (PMI). The top k ranked patterns are retained and used to find new pattern instances[9] which are in turn ranked with that same PMI-based measure, keeping only the top ranked instances. The system was evaluated on the TREC and CHEM data sets and achieved fairly good precision values, however general performance, in terms of F-scores, is hard to compare since recall was given relative to one of the variants of their system.

Similar to Pantel's ESPRESSO system described above, BOLLEGALA ET AL. [2010] presents a system that is capable of extracting both patterns and pattern instances. In contrast to ESPRESSO,

[8] cf. HTTP://WWW.ACLWEB.ORG/ACLWIKI/INDEX.PHP?TITLE=SAT_ANALOGY_QUESTIONS

[9] A *pattern instance* is a pair of words that has been observed with the respective pattern in text.

Bollegala et al. proceed in a single-pass fashion using even less supervision. They simultaneously extract noun phrases from a corpus, using shallow linguistics processing tools like a chunk annotator, and patterns as the tokens between noun pairs with skips allowed. Both surface forms and part-of-speech tag sequences are retained as patterns. They then apply co-clustering to simultaneously find the relations between nouns and the best patterns representing those relations, in what Bollegala et al. [2010] call *relational duality.* The system scores well in a relational similarity task using the ENT data set – which is similar to the SAT used in this thesis but specializes in named entities rather than common nouns – and gains promising F-scores in a relation classification task, cf. Figure 2.1.

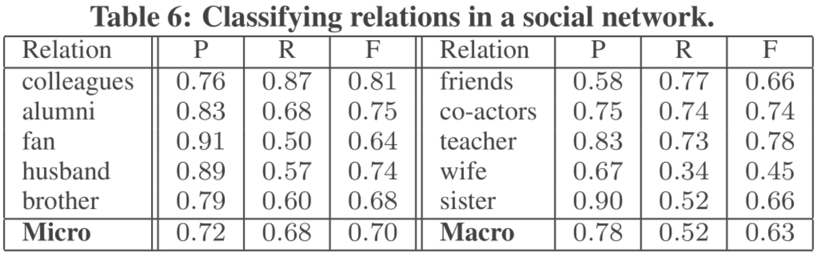


Table 6: Classifying relations in a social network.

Relation	P	R	F	Relation	P	R	F
colleagues	0.76	0.87	0.81	friends	0.58	0.77	0.66
alumni	0.83	0.68	0.75	co-actors	0.75	0.74	0.74
fan	0.91	0.50	0.64	teacher	0.83	0.73	0.78
husband	0.89	0.57	0.74	wife	0.67	0.34	0.45
brother	0.79	0.60	0.68	sister	0.90	0.52	0.66
Micro	0.72	0.68	0.70	**Macro**	0.78	0.52	0.63

Figure 2.1: F-scores in Bollegala et al. [2010]

Moldovan et al. [2004] define 35 different semantic relations between noun phrases at various semantic levels, manually annotate sentences and short syntactic patterns with these relations, and use the data to train a classification algorithm they call *semantic scattering.* They distinguish F-scores for different classes of nominals and attain scores between 0.33 for adjective phrases and 0.75 for genitives with *of.* Even though their approach lacks scalability due to the extensive manual labor involved, it provides interesting results since such a wide variety of semantic relations between nominals is predicted. Unfortunately, their annotated data seems unavailable for comparative evaluation.

Finally, Snow et al. [2006] propose a probabilistic framework for taxonomy induction over word senses and demonstrate their method for a taxonomy comprising hypernymy and co-hyponymy. They train separate classifiers for each semantic relation using the shortest path connecting two words along a dependency parse as feature and logistic regression as learning algorithm. The predictions from the classifiers are used to generate candidate relation instances. Subsequently, Snow et al. [2006] jointly infer which pairs and relations to insert into the taxonomy such that the likelihood of the taxonomy given the evidence is maximized. Basically, from all possible taxonomies, they select the one that maximizes the conditional probability of the evidence. To avoid an explosion of the search space they define an ADD-RELATION operator which allows them to incrementally build up the optimal taxonomy. The reported (averaged?) F-score for a randomly sampled, hand-labeled test set[10] is 0.31, with a precision of 0.58 and a recall of 0.21.

At long last, there is a very recent publication by Weeds et al. [2014], where the authors train a linear support vector machine (SVM) to distinguish lexical entailment /hypernymy and co-hyponymy. In the process they show that different combinations of noun vector representations

[10] see Snow et al. [2004]

lend themselves to predicting different semantic relations between nominals. In particular they find that taking the vector difference between two noun representations is conducive to predicting entailment while co-hyponymy becomes more salient when summing up the noun vectors. Extracted patterns were based on grammatical dependencies involving open class parts-of-speech (POS). These patterns were weighted with the *positive point-wise mutual information.* Weeds et al. [2014] report their results in terms of the average accuracy for a variety of similarity measures. Scores for hypernymy oscillate between 0.37 and 0.75 and peak when using vector difference. Co-hyponymy behaves similarly with accuracy values between 0.37 and 0.79 but it seems that best results are reached when using the Cosine similarity or Lin's similarity measure (Lin [1998a]).

2.3 Further related research

In this last section of the chapter on related work I examine research that used the bless evaluation data to assess the quality of distributional semantic models and semantic relation predictors. Scientific articles published to date that cite Baroni and Lenci [2011] and actually use the evaluation set, fall into three categories: those that evaluate semantic similarity measures, those that identify semantic relations, and the ones that use multimodal distributional semantics to improve semantic representations of word meaning. I distinguish the latter two because their focus is slightly different, with multimodal distributional semantics, being a rather recent development, still working out how to best combine textual and image representations of word meanings.

Both Panchenko and Morozova [2012] and Panchenko et al. [2012] are of the first group and therefore do not compare to the work at hand. Bruni et al. [2012] and Bruni et al. [2014] work on multimodal distributional semantics and report their results as distributions of *z-normalized cosines* as suggested by Baroni and Lenci [2011] (see Figure 2.2 for an example). While these boxplots are useful when exploring a word space, the measure is too lenient and incomparable to other state-of-the-art work when identifying semantic relations and scoring the output of a particular predictor.

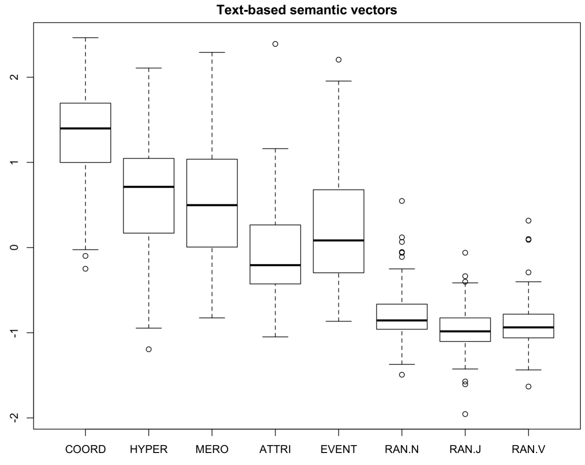
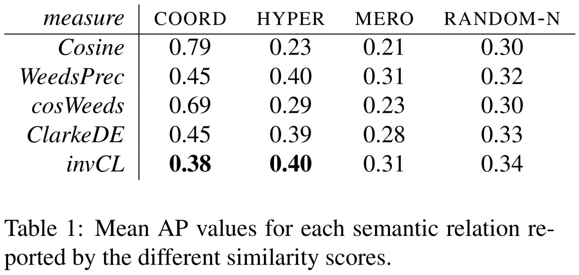


Figure 2.2: Distribution of z-normalized cosines; from Bruni et al. [2014]

This leaves Santus et al. [2014] and Lenci and Benotto [2012] who actually identify semantic relations and provide average precision in addition to distributions of cosine distances. Lenci and Benotto [2012] evaluate several directional, asymmetric similarity measures with hypernym prediction. These measures are based on including features of a term a, which is semantically narrower than term b, in the feature vector of term b; a procedure based on the *distributional inclusion hypothesis.* Lenci and Benotto's own suggested measure additionally excludes features of hypernym b from the vector of hyponym a. As features they use "direct and inverse links formed by (partially lexicalized) syntactic dependencies and patterns" weighted with *local mutual information* (LMI)[11]. Figure 2.3 shows their average precision values in comparison to several other measures from the field.

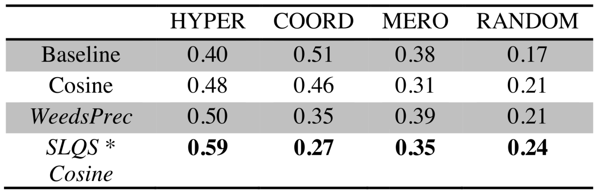


measure	COORD	HYPER	MERO	RANDOM-N
Cosine	0.79	0.23	0.21	0.30
WeedsPrec	0.45	0.40	0.31	0.32
cosWeeds	0.69	0.29	0.23	0.30
ClarkeDE	0.45	0.39	0.28	0.33
invCL	**0.38**	**0.40**	0.31	0.34

Table 1: Mean AP values for each semantic relation reported by the different similarity scores.

Figure 2.3: Average precision values computed over the bless data for several similarity measures; from Lenci and Benotto [2012].

Santus et al. [2014] define an entropy-based measure for estimating the informativeness of contexts and use differences in informativeness of contexts to distinguish relations.
If distributional similarity of two vector representations (measured in Cosine similarity) comes paired with considerable difference in informativeness, Santus' measure will predict hypernymy as the given relation rather than co-hyponymy which tends to be symmetric. Contexts are computed using a window-based pattern scheme between a target word and its two nearest content words neighboring on the left and right side. Again, observations are weighed using the LMI. Like Lenci and Benotto [2012], they evaluate their measure in terms of average precision, borrowing from methodology developed by Kotlerman et al. [2010]. See Figure 2.4 for how their entropy-based measure compares to other prevalent approaches.



	HYPER	COORD	MERO	RANDOM
Baseline	0.40	0.51	0.38	0.17
Cosine	0.48	0.46	0.31	0.21
WeedsPrec	0.50	0.35	0.39	0.21
*SLQS * Cosine*	**0.59**	**0.27**	**0.35**	**0.24**

Figure 2.4: Average precision values computed over the bless data for several similarity measures; from Santus et al. [2014].

[11] also known as Lexicographer's mutual information

3 Methods

3.1 Overview

As detailed in Section 1.2, the basic idea underlying this work is to build noun pair representations from sentences containing pairs of distributionally similar nouns, and use these representations to learn the semantic relation that holds between the nouns in such a pair. This chapter describes the specifics of how these two aspects, representation and learning, were implemented and unrolls the design choices made. There are two parts to each topic: The first two parts deal with representing the input: initially in terms of a pair-pattern matrix in feature space (Section 3.2.1), which is then further transformed into a pair-pair matrix in similarity space (Section 3.2.3). The ensuing two sections are all about learning to predict relations between nouns in a pair – in a supervised manner (Section 3.3.1) and without supervision (Section 3.3.2). The final section describes the evaluation procedures applied. Figure 3.1 gives a high-level overview of the whole system[1].

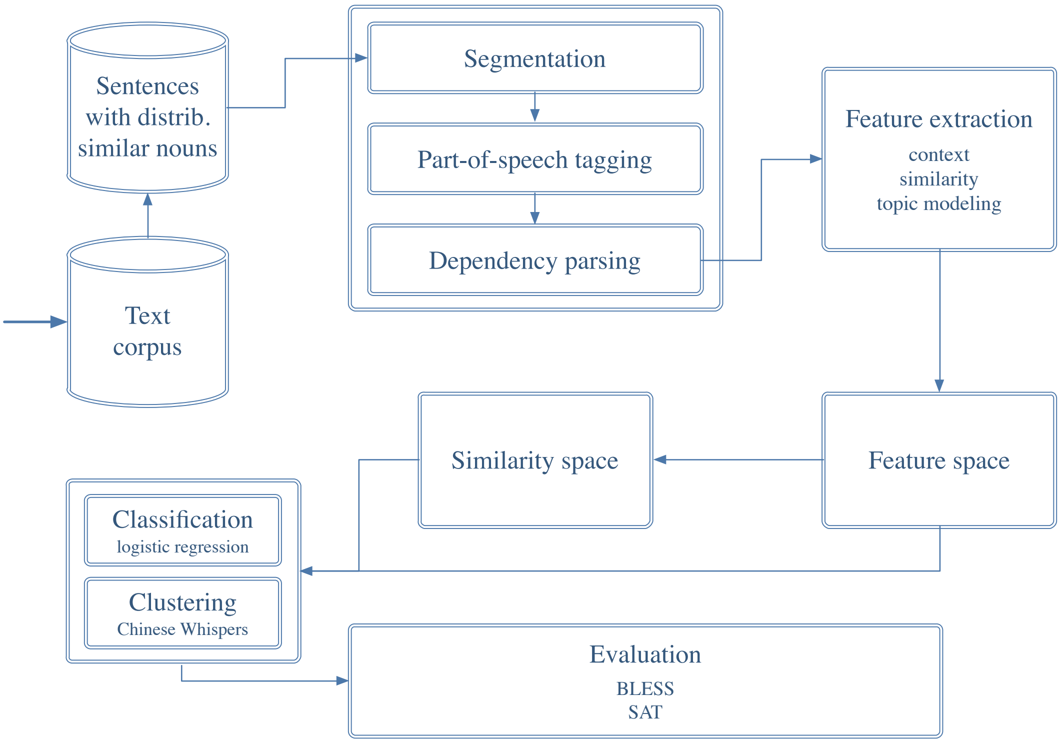


Figure 3.1: Data flow diagram of the system used in this work.

[1] See HTTPS://GITHUB.COM/POLICECAR/SENSIM and HTTPS://GITHUB.COM/POLICECAR/SIMSETS for the code.

3.2 Representation

3.2.1 Feature extraction

Data acquisition and preprocessing Two corpora served as interchangeable input: a corpus of English news text from the years 2005 to 2010, henceforth called *News120M*, and the *PukWaC* corpus from the collection of *WaCky* corpora (Baroni et al. [2009]). Both are gigaword-sized corpora in the English language built from web crawls and contain more than 80 million sentences. While News120M comes in plain text, the PukWaC corpus is annotated with part-of-speech tags and lemmata using the TreeTagger (see Schmid [1994] and Baroni et al. [2009]) and with dependency parses generated by the MaltParser (Nivre et al. [2006]). Accordingly, PukWaC was used as is while News120M was annotated using software for segmentation, part-of-speech tagging, lemmatizing, and parsing provided by Stanford's natural language processing group (cf. Toutanova et al. [2003], Klein and Manning [2003], and De Marneffe et al. [2006]). These particular corpora were chosen because they are sizeable, single-language, and freely available. In case of PukWaC, the recommendation by the authors of the evaluation set bless (see Section 3.4.1) weighed in additionally. bless having been constructed from PukWaC gave reason to expect maximal data overlap, thereby substantiating classification results.

Subcorpus selection For this work only sentences containing pairs of semantically similar nouns are of interest. Therefore a selection mechanism was required to produce the relevant subcorpus. Established techniques from lexical semantics used to measure semantic similarity can be grouped into two categories:

- knowledge-based and
- knowledge-free

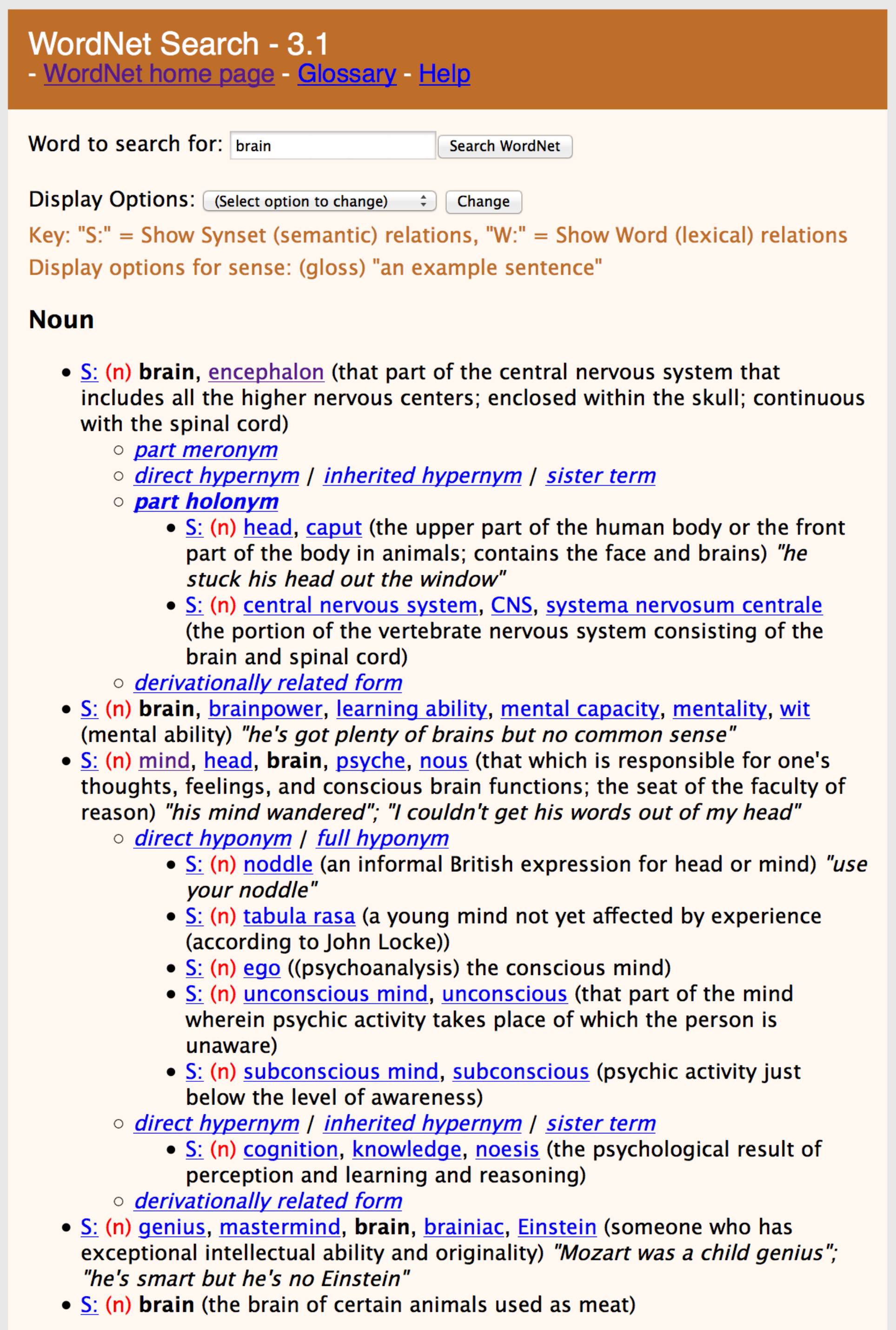
Knowledge-based approaches include thesauri, semantic networks, taxonomies and encyclopedias manually built by human experts. Knowledge-free strategies try to induce sets of near-synonyms (synsets) from distributional properties of words in a corpus in an unsupervised manner.

WordNet[2] is one of the better known and more widely used knowledge-based resources. It is a large English lexical database of synsets "interlinked by means of conceptual-semantic and lexical relations"[2]. Figure 3.2 shows some of the semantically related terms for the word *brain*, taken from their online demo. One of the problems with WordNet in the context of automatic use is that it is difficult to determine a useful number of hops to follow along a specific relation; for example to retrieve indirect hypernyms or hyponyms of a word. The hypernym *animal*, e.g., can be found six hops away from the word *alligator* but only one hop away from the word *pet*. This is to be attributed to the divergent degrees of specificity in the database across different topics.

An alternative, knowledge-free approach to building distributional thesauri (DT) is described in Biemann and Riedl [2013] and implemented in the freely available *JoBimText* software[3]. It facilitates automatic thesaurus generation for a given corpus thereby supporting flexibility regarding the input corpus and language used. With the default settings the distributional thesaurus comprises the 200 most distributionally similar words (*expansions*) for each of the

[2] WordNet. Princeton University. 2010. HTTP://WORDNET.PRINCETON.EDU

[3] cf. HTTP://MAGGIE.LT.INFORMATIK.TU-DARMSTADT.DE/JOBIMTEXT/



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: brain [Search WordNet]

Display Options: (Select option to change) [Change]

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **brain**, encephalon (that part of the central nervous system that includes all the higher nervous centers; enclosed within the skull; continuous with the spinal cord)
  - *part meronym*
  - *direct hypernym* | *inherited hypernym* | *sister term*
  - ***part holonym***
    - S: (n) head, caput (the upper part of the human body or the front part of the body in animals; contains the face and brains) *"he stuck his head out the window"*
    - S: (n) central nervous system, CNS, systema nervosum centrale (the portion of the vertebrate nervous system consisting of the brain and spinal cord)
  - *derivationally related form*
- S: (n) **brain**, brainpower, learning ability, mental capacity, mentality, wit (mental ability) *"he's got plenty of brains but no common sense"*
- S: (n) mind, head, **brain**, psyche, nous (that which is responsible for one's thoughts, feelings, and conscious brain functions; the seat of the faculty of reason) *"his mind wandered"; "I couldn't get his words out of my head"*
  - *direct hyponym* | *full hyponym*
    - S: (n) noddle (an informal British expression for head or mind) *"use your noddle"*
    - S: (n) tabula rasa (a young mind not yet affected by experience (according to John Locke))
    - S: (n) ego ((psychoanalysis) the conscious mind)
    - S: (n) unconscious mind, unconscious (that part of the mind wherein psychic activity takes place of which the person is unaware)
    - S: (n) subconscious mind, subconscious (psychic activity just below the level of awareness)
  - *direct hypernym* | *inherited hypernym* | *sister term*
    - S: (n) cognition, knowledge, noesis (the psychological result of perception and learning and reasoning)
  - *derivationally related form*
- S: (n) genius, mastermind, **brain**, brainiac, Einstein (someone who has exceptional intellectual ability and originality) *"Mozart was a child genius"; "he's smart but he's no Einstein"*
- S: (n) **brain** (the brain of certain animals used as meat)

Figure 3.2: Excerpt from WordNet's synset for the term *brain*.

100 000 most frequently observed words (*targets*). Every *target-expansion pair* also receives a within-target ranking. Table 3.1 shows examples of target-expansion pairs including rank, with part-of-speech tags removed, which are part of the original DT. For News120M a DT can be downloaded from the JoBimText website[4], for PukWaC a fresh one was computed. These were filtered for the specific needs of this study: common nouns (NN and NNS) were selected and some words including email addresses and words consisting only of digits were filtered out. The exact Pig Latin[5] regular expression for this filter reads:

```
filtered = filter target_expansions by
    (regex_extract_all(target, ’.*[0-9\\.\\+@].*’) is NULL) and
    (regex_extract_all(expansion, ’.*[0-9\\.\\+@].*’) is NULL);
```

These target-expansion pairs were then pruned to match the available computational resources, keeping only the top 50 most similar expansions per target word. The resulting pairs were used to filter the corpora. For this purpose, all common nouns in the annotated sentences were marked. A word was considered a common noun, if it was tagged with one of the POS tags NN or NNS. Subsequently, only those sentences were selected that contained at least two distributionally similar common nouns.

	News120M		PukWaC	
Target	Expansion	Similarity	Expansion	Similarity
brain	lung	200.0	liver	128.0
brain	liver	158.0	kidney	120.0
brain	pancreas	131.0	lung	106.0
brain	kidney	131.0	heart	103.0
brain	stomach	117.0	mind	99.0
brain	retina	115.0	tissue	97.0
brain	intestine	111.0	bowel	95.0
brain	tissue	110.0	organ	90.0
brain	nerve	104.0	skin	90.0
brain	heart	101.0	gland	89.0
brain	ovary	101.0	pancreas	88.0
brain	abdomen	99.0	marrow	84.0
brain	organ	99.0	bladder	83.0
brain	spleen	98.0	prostate	83.0
brain	gland	95.0	intestine	81.0
brain	bladder	95.0	cortex	80.0
brain	marrow	95.0	muscle	74.0
brain	cortex	94.0	spleen	74.0
brain	spine	92.0	tract	73.0
brain	skin	92.0	bone	71.0
brain	muscle	90.0	stomach	69.0
brain	uterus	88.0	nerve	67.0

Table 3.1: Expansions for the target word *brain* for PukWaC and News120M.

[4] Download similarities-news120M_stanford_lemma_np.tar.gz_1-3 from HTTP://SOURCEFORGE.NET/PROJECTS/JOBIMTEXT/FILES/DATA/MODELS

[5] HTTPS://PIG.APACHE.ORG

Word representations As briefly introduced in Section 2.1.2, when talking about different kinds of contexts used in the literature, there are ample possibilities to represent words or word pairs in distributional semantic models. The most common options are: bag-of-words, n-gram models, skip-gram models, and dependency paths. **Bag-of-words** models collect words adjacent to the target word within a chosen word window and arrange them in no particular order, for instance, alphabetically. ***N*-gram models** take the n words preceding the target word and collect them preserving the original word order. **Skip-gram models** are analogous to n-gram models except that they permit skipping some of the collected words in the final representation. **Dependency paths** do not rely on the linear or surface word order as the previous models but rather consider words that are connected to the target word with grammatical dependencies and thereof a particular number of hops. Figure 3.3 displays a dependency parse for an example sentence[6]. Dependency paths can be combined with skip-gram models in that dependency paths are used for pattern extraction but also some of the retrieved tokens can be omitted from the pattern.

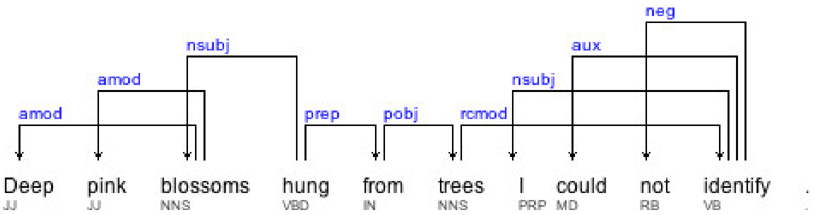


Figure 3.3: Visualization of a dependency parse; generated with `DependenSee`.

To yield more general patterns and to capture the linguistic structure of sentences better, dependency parses were used as the basis of pattern extraction in this study. As Figure 3.4 shows, dependency parses can capture long-range relations with a smaller window than a linear view of the sentence would permit[7]. For example, the words *electronencephalography* and *medical tool* are connected with 4 dependency arcs, which would take a word window of at least 10 hops. A larger word or arc window is not necessarily a disadvantage in itself, but it introduces more noise into the patterns.

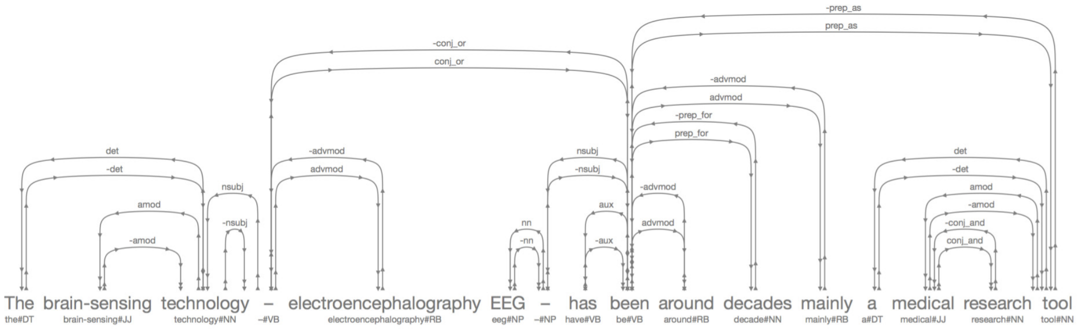


Figure 3.4: Visualization of a dependency parse with collapsed dependencies; generated with `JoBimText`

[6] The `DependenSee` code can be found at HTTPS://GITHUB.COM/AWAISATHAR/DEPENDENSEE.

[7] Figure 3.4 was generated with HTTP://MAGGIE.LT.INFORMATIK.TU-DARMSTADT.DE:10080/JOBIM/.

Dependency paths From each annotated and selected sentence, **all subtrees** along the dependency parse containing at least one target-expansion pair were extracted, yielding patterns as depicted in Table 3.2. The length of these subtrees was restricted to 6 or less dependency arcs in a trade-off between computational feasibility and semantic requirements. One of the reasons to choose all subtrees over the shortest path was to include patterns like ⟨*include X and Y*⟩ – where this specific one might be indicative of co-hyponymy. While the *all subtrees* approach at first produces much more patterns, the infrequent ones thin out during frequency pruning later on.

Additionally, nouns were lemmatized for further processing – a useful generalization at the expense of dismissing some information. The advantage gained in terms of reduced data sparseness, achieved by merging singular and plural versions of many regular nouns, outweighed the loss in semantic distinction in cases like *glass* and *glasses*. Target and expansion words were replaced with *X* and *Y*, respectively, to facilitate generalization of patterns across pairs while marking the position of nouns in the pattern. Finally, all tokens were converted to lower case. Due to the graph library used for pattern extraction[8], only sentences of a maximum number of 30 tokens could be considered.

Noun	Noun	Pattern
alligator	crocodile	include X and Y
alligator	crocodile	be an X or a Y
alligator	crocodile	not just X – Y ,
alligator	crocodile	X Y forms such as
snake	crocodile	X and Y
snake	crocodile	animals include X , tortoises Y
snake	crocodile	showed us a X a Y
snake	crocodile	of creatures including X and Y
animal	alligator	X such as Y are
animal	alligator	X including Y and
animal	alligator	X as Y are raised and
animal	alligator	other X such as Y ,
alligator	mouth	head inside an X Y
alligator	mouth	head inside X 's Y
alligator	mouth	head inside X Y

Table 3.2: Raw features as extracted from parsed and annotated text.

Pair-pattern matrix For each noun pair the extracted patterns were collected, co-occurrences counted, and the result stored in a table as well as in vector representation for further processing. The combination of all these row vectors equals a pair–pattern matrix as they are commonly used in distributional semantic models.

[8] HTTP://JGRAPHT.ORG

3.2.2 Measures of association

In the previous section, co-occurrences of patterns and noun pairs were extracted from a corpus and used to span a word space into which to place these noun pairs. Thereby their frequency counts were taken to be the value or weight of the observation. Such raw co-occurrence data, however, have serious shortcomings. The first is that the observed raw counts only provide information about the particular excerpt of text they were derived from, i.e. the specific corpus. They, therefore, directly mirror the contingency of the evidence. A second shortcoming is that the plain frequencies do not necessarily capture statistically significant associations of pairs and patterns. If a pattern and a pair occur sufficiently frequent in the corpus, their co-occurrence might be merely coincidental.

According to Evert [2005], the most common method for distinguishing random co-occurrences from true statistical associations are *association measures.* Association measures compute a score for each raw co-occurrence datum, here pair and pattern, which can be used to rank all associations of a pair or select the best associations based on a threshold. Among the most widely used association measures are mutual information (Church and Hanks [1990]), the t-score measure (Church et al. [1991]), the log-likelihood ratio (Dunning [1993]), and the χ^2 statistic (Agresti [1990])[9]. Which measure to use, varies with the assumptions that can be made about the data or one is willing to make. The t-test, for example, presupposes a normal distribution of the data – an assumption that oftentimes does not hold for word frequencies, which tend to follow a Zipfian distribution. The χ^2 test, on the other hand, can be problematic for sparse data and small sample sizes. While the log-likelihood measure works well with sparse data and is more easily interpretable than the χ^2 test, we decided to use derivatives of mutual information as they have been shown to work fairly well in the context of relation extraction, see, e.g., Pantel and Pennacchiotti [2006], Weeds et al. [2014].

Mutual information Mutual information (MI) is originally a measure from information theory which describes the information overlap between two events or distributions. More formally speaking, MI measures the average reduction in uncertainty in one variable that results from learning the value of the other. It is based on the concepts of *(Shannon) entropy*, *marginal entropy*, and *conditional entropy*[10]. Thereby, the entropy of a random variable X with distribution p is a measure of its uncertainty and is denoted by $H(X)$. The mutual information between two random variables X and Y is then defined as

$$I(X;Y) \equiv H(X) - H(X|Y) \tag{3.1}$$

and satisfies

$$\begin{aligned} I(X;Y) &= I(Y;X) && (3.2)\\ I(X;Y) &\geq 0 && (3.3)\end{aligned}$$

– see MacKay [2003]. For discrete random variables, the entropy is defined as

$$H(X) \equiv -\sum_{x} p(x) \cdot log_2\, p(x) \tag{3.4}$$

[9] Citations refer to publications that introduced or prominently applied the respective measure to research in computational linguistics.

[10] For an introduction to information theory read Shannon [1948] and MacKay [2003] and for the foundations of probability theory, refer to Papoulis [1965], Jaynes [2003] or Murphy [2012].

and the conditional entropy of two discrete random variables is

$$H(X|Y) = -\sum_{x,y} p(x|y) \cdot log_2\, p(x|y) \tag{3.5}$$

Substituting H in Equation 3.1 with Equations 3.4 and 3.5 and considering that the joint probability distribution is $p(x, y) = p(y|x) \cdot p(x) = p(x|y) \cdot p(y)$ gives us

$$I(X;Y) = \sum_{x,y} p(x,y) \cdot log_2 \frac{p(x,y)}{p(x) \cdot p(y)} \tag{3.6}$$

The formulation of the mutual information above is the expected value over all possible outcomes or values of two random variables. In order to compute the measure for two specific values of the random variables, that is for a particular pair and pattern, we need to compute the **point-wise mutual information** (PMI) which is just

$$PMI(x;y) = log_2 \frac{p(x,y)}{p(x) \cdot p(y)} \tag{3.7}$$

It computes the logarithmic ratio of the actual joint probability of a pair and a pattern to their expected joint probability assuming they are independent events. Since both the marginal probabilities and the joint probability are usually unknown, it is common practice to approximate them with the normalized frequency counts from the observed data.

However, the PMI has a preference for low-frequent words and lacks a fixed upper bound, which is the reason several variants of it are used in computational linguistics and natural language processing. Most popular are the **normalized point-wise mutual information**, Bouma [2009], and the **Lexicographer's or local mutual information**, Bordag [2008]. The normalized point-wise mutual information (NPMI) gives the PMI a fixed upper (and lower) bound by normalizing the measure with the logarithm of the joint probability $log_2 \frac{1}{p(x,y)}$ such that the maximum value is 1.0 (highest association) and the minimum value is 0.0 (no association). Meanwhile, the Lexicographer's mutual information (LMI) compensates for the high significance scores assigned to low-frequent pair-pattern combinations by weighing the PMI score with the pair-pattern frequency. Equation 3.9 shows the resulting formula.

$$NPMI(x;y) = \frac{PMI(x;y)}{-log_2\, p(x,y)} \tag{3.8}$$

$$LMI(x;y) = p(x,y) \cdot PMI(x;y) \tag{3.9}$$

In this work, the LMI was used as a measure of association since it performs as good as the log-likelihood on the task while being cheaper to compute (Biemann and Riedl [2013]), and yields better results than the PMI (Riedl and Biemann [2013]). The LMI was preferred over the NPMI because scoring pair-pattern associations proportionate to their joint frequency carried more weight than normalization and the used algorithms did not depend on the latter. Table 3.3 shows the top ranked patterns for the pair *plum::@::cherry* produced by the raw frequency counts and the LMI, respectively.

PATTERN	FREQUENCY	PATTERN	LMI SCORE
X and Y	405	spicy X Y	5823.70
of X Y	390	of X Y	3562.79
spicy X Y	378	X and Y	3089.95
X , Y	126	of spicy X Y	1417.72
X Y and	109	spicy X and Y	1412.13
of X and Y	102	intense X Y	1340.86
with X Y	95	of X and Y	900.61
intense X Y	93	with X Y	765.89
spicy X and Y	92	X , Y	760.23
of spicy X Y	92	X Y and	731.37
X , Y and	52	intense X Y and	570.75
with X Y and	37	with intense X Y	533.75
with X , Y	37	intense X , Y	533.75
with intense X Y	37	dark with X Y	533.03
intense X Y and	37	with X Y and	313.50
intense X , Y	37	X , Y and	299.83

Table 3.3: Top ranked patterns for *plum::@::cherry* – frequency counts versus LMI.

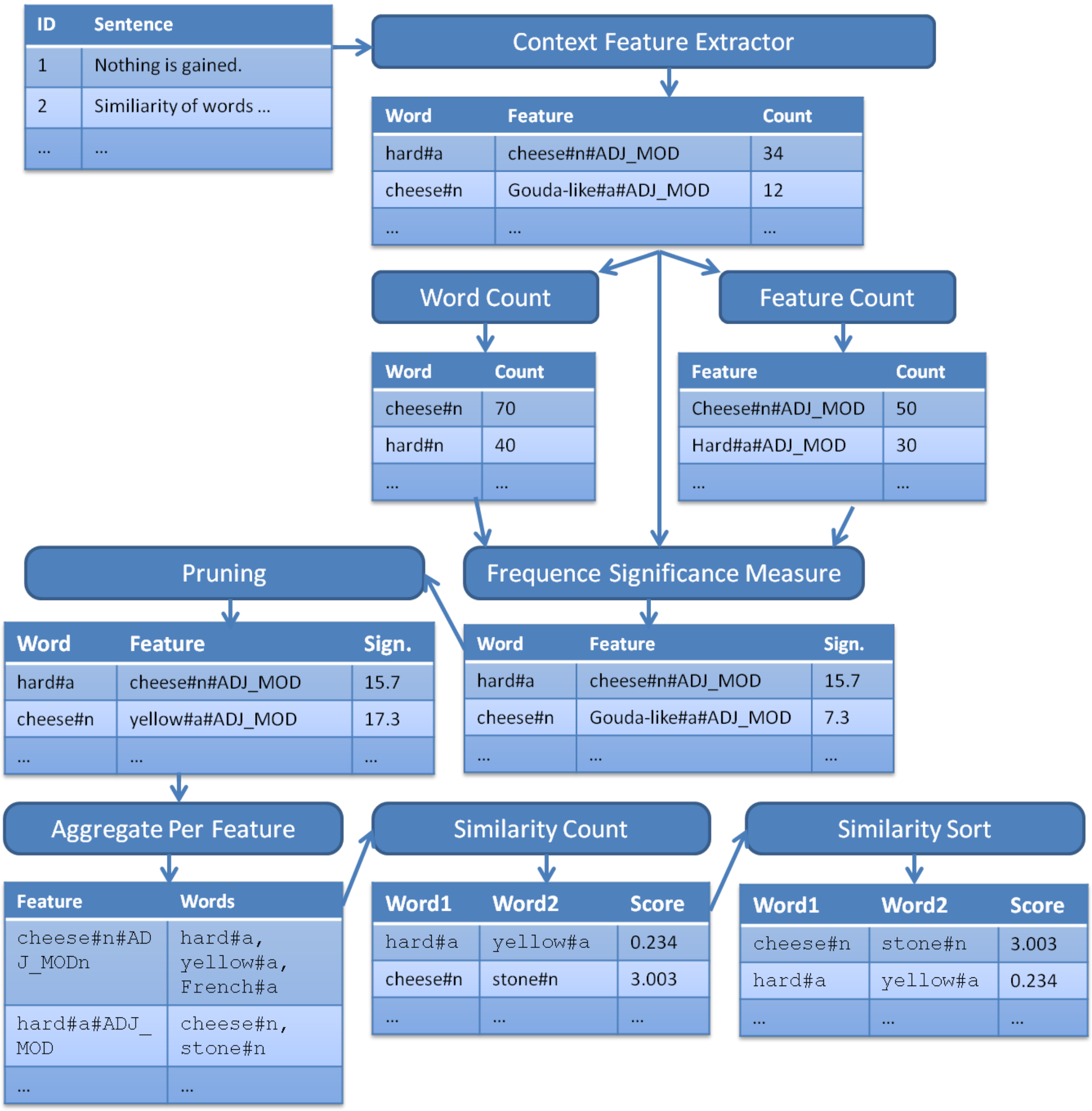
3.2.3 Similarity measures

In order to make similar pairs available as an extra feature in classification and for later clustering of the distance matrix, a similarity space was constructed in addition to the feature space described in Sections 3.2.1 and 3.2.2. This was done in a twofold manner, both times using the JoBimText software and workflow as depicted in Figure 3.5 – see Biemann and Riedl [2013] for a more detailed description of the component parts. For one, the existing feature space was transformed into similarity space. On that account the number of matching non-zero entries between any two pair vectors was added up and used as similarity count. From the result the 200 most similar noun pairs per pair were retained.

Moreover, a second similarity space was constructed based on *subject-verb-object* features. Again, the 200 most similar pairs per pair were computed, this time based on their 1000 most significant features in terms of the Lexicographer's mutual information. Similarity was calculated using **pattern overlap count** as above. The resulting similarity space constitutes a second, separate representation space of the noun pairs, which was used as an additional feature group during classification.

The similarity measure used here is one of the most basic ones, often used as a baseline, cf., e.g., Bordag [2008]. It was chosen at long last after the available computational resources could not compute the Cosine similarity, not even on ten percent of the data.

The Cosine similarity is a measure commonly used in studies of distributional semantics because it spans a space that produces good results in many tasks. But the variety of similarity measures out there is vast and "it is difficult to reach a conclusion from the literature regarding which similarity measure is best; again this appears to depend on the application and which relations one hopes to extract.", Clark [2012].



ID	Sentence
1	Nothing is gained.
2	Similiarity of words ...
...	...

Word	Feature	Count
hard#a	cheese#n#ADJ_MOD	34
cheese#n	Gouda-like#a#ADJ_MOD	12
...	...	...

Word	Count
cheese#n	70
hard#n	40
...	...

Feature	Count
Cheese#n#ADJ_MOD	50
Hard#a#ADJ_MOD	30
...	...

Word	Feature	Sign.
hard#a	cheese#n#ADJ_MOD	15.7
cheese#n	yellow#a#ADJ_MOD	17.3
...	...	...

Word	Feature	Sign.
hard#a	cheese#n#ADJ_MOD	15.7
cheese#n	Gouda-like#a#ADJ_MOD	7.3
...	...	...

Feature	Words
cheese#n#ADJ_MODn	hard#a, yellow#a, French#a
hard#a#ADJ_MOD	cheese#n, stone#n
...	...

Word1	Word2	Score
hard#a	yellow#a	0.234
cheese#n	stone#n	3.003
...	...	...

Word1	Word2	Score
cheese#n	stone#n	3.003
hard#a	yellow#a	0.234
...	...	...

Figure 3.5: JoBimText workflow as implemented in MapReduce; image from Biemann and Riedl [2013].

No less, two coarse groups of (dis)similarity measures can be distinguished in the context of DSMs: those that are specific to distributional semantics and aim to incorporate linguistic knowledge in the measure and, secondly, generic measures that are deployed in a wide variety of fields. The former include WeedsPrec (Weeds et al. [2004]), ClarkeDE (Clarke [2009]), and invCL (Lenci and Benotto [2012]). These are directional, asymmetric similarity measures that take into account differences in the generality or informativeness of features. The underlying assumption is that hypernyms (e.g., *vehicle*) tend to occur in more contexts than their hyponyms (e.g., *motor cycle*) – a piece of information that can be exploited for predicting semantic relations with distributional similarity.

Among the more widely used (dis)similarity measures that consider general mathematical properties are:

- Euclidean distance
- Cosine similarity
- Jaccard index

Depending on the measure used the given feature representations are conceptualized differently. The **Euclidean distance** measures distance in terms of the length of the path connecting two points in space. Accordingly, feature representations are construed as points in a Euclidean space with as many dimensions as overall features observed. Meanwhile the **Cosine similarity** calculates similarity as the cosine of the angle between two vectors[11]; i.e. features are represented as vectors in a space spanned by all features. Conversely, the **Jaccard index** is an index or coefficient of the similarity of sets. Correspondingly, a term is represented as the set of features it was observed with – rather than the highly sparse objects used in Euclidean distance or Cosine similarity.

While some measures compute the *distance* between objects and others the *similarity*, this is primarily a different way of looking at the same thing and conversion between values of similarity and distance is typically straight forward. Given a similarity measure $s(x, y)$ and a distance measure $d(x, y)$, their conversion could take the following form:

$$d(x,y) = 1 - s(x,y) \tag{3.10}$$

$$s(x,y) = \frac{1}{d(x,y)} \tag{3.11}$$

Both similarity and distance measures are considered metrics if they satisfy the following conditions (given for the case of a distance measure):

- Non-negativity: $d(x, y) \geq 0$
- Reflexivity: $d(x, y) = 0$ if and only if $x = y$
- Symmetry: $d(x, y) = d(y, x)$
- Triangle inequality: $d(x, y) + d(y, z) \geq d(x, z)$

The equivalent of non-negativity for similarity measures is the so-called *limited range* with $s(x, y) \leq s_0$ for some arbitrarily large number s_0 and the triangle inequality becomes $s(x, y) \cdot s(y, z) \leq [s(x, y) + s(y, z)] \cdot s(x, z)$, Goshtasby and Le Moign [2012].

After this brief digression on similarity measures, let us now see how the representations acquired can be utilized to learn semantic relations using algorithms from machine learning.

[11] I use the terms *distance measure* and *similarity measure* almost interchangeable here – bear with me for a moment, I will explain why in the next paragraph.

3.3 Learning

In the preceding sections I explained and described how the raw textual data were transformed into numeric noun pair representations amenable to applying machine learning algorithms to. I will now give a brief introduction to different learning paradigms, and in particular to classification and clustering, each time followed by a detailed account of the specific algorithm used in this work.

Three learning paradigms dominate the field of machine learning:

- supervised learning
- unsupervised learning
- reinforcement learning

These paradigms differ primarily in the amount and kind of feedback or error correction they provide during learning. In supervised learning the model receives labeled training data where the label is the correct class of a sample or data point. Unsupervised learning omits labels altogether and instead aims at deducing statistical regularities inherent in the data. Lastly, reinforcement learning provides delayed reward signals instead of direct feedback on each data point.

There are other paradigms in the wild and in the literature – for instance, active learning or semi-supervised learning which is a mixture of supervised and unsupervised learning, as well as the recently emerging field of probabilistic programming – but the three listed above are the mostly widely spread so far.

In the following supervised and unsupervised learning will be considered further: unsupervised learning because it is knowledge-free thereby matching the constraints imposed upon this work. And supervised learning to assess whether the setting at hand does in principle represent a learnable problem and to facilitate comparison with related work which is much harder with cluster evaluation because measures of evaluation vary widely. The following sections provide a closer look at these paradigms and their concrete implementation in this thesis.

3.3.1 Classification

Algorithms for supervised learning fall into one of two categories: they are regressors or classifiers. Both rely on labeled data for training but regressors predict continuous target values, while classifiers make predictions about discrete classes. Among the classifiers are algorithms which can handle multiclass and /or multi-label classification[12] and others that specialize in binary classification. For either of these options different strategies are available which include cost-sensitive one-against-all or weighted-all-pairs for multiclass and one-against-all or error correcting tournament for binary classification. Additional design choices to keep in mind are the type of algorithm used, regularization methods, loss functions, optimization strategies, sampling procedures, etc.

Learning algorithms consist of and differ along three dimensions, Domingos [2012]:

[12] Multiclass means that the classification task has more than two classes; e.g., distinguishing cat, squirrel, skunk, and quokka images. Multi-label means that any sample can have more than a single label; as might be the case if the training data contains images of quokkas and squirrels together.

- the representation used, also known as the *hypothesis space*
- their evaluation function, also known as *objective function* or scoring function, and
- the optimization technique used to find the optimal classifier

The representation or hypothesis space determines the set of classifiers that can be learned and is basically a restriction upon the space of all functions. Table 3.4 shows some kinds of hypothesis spaces together with examples of models that implement them.

Type of representation	Exemplary model(s)
Instances	k-nearest neighbors
Hyperplanes	Naive Bayes, Logistic regression
Sets of rules	Inductive logic programming
Decision trees	Decision trees, random forests
Artificial neural networks	Perceptron
Graphical models	Conditional random fields

Table 3.4: Common hypothesis spaces for learning algorithms.

Learning algorithms can further be distinguished with regard to the objective function they use to separate good from bad classifiers. The objective function is also known as loss or cost function since it maps an event, e.g., a prediction made, to the loss or cost associated with that event. Table 3.5 presents a range of common objective functions:

Objective functions
Precision, recall, F-score
Squared error, zero-one loss
Maximum Likelihood
Posterior probability
Kullback-Leibler divergence
Backpropagation

Table 3.5: Objective functions used in learning algorithms; see also Domingos [2012].

Finally, there are a number of optimization techniques to search for the highest-scoring classifier among all classifiers in the hypothesis space. They differ in terms of the properties of the optimization function (linear, quadratic, non-linear, non-smooth) and the constraints imposed on this function (unconstrained, bound, linear, smooth, discrete). Table 3.6 gives several examples of different algorithms for optimization[13].

[13] Note: convex optimization is merely a special case of optimization with continuous domain. Also, *combinatorial* is synonymous with *discrete* here.

Kind of optimization	Example algorithms
Convex optimization	Least-squares Linear programming
Continuous optimization	Gradient descent Quadratic programming
Combinatorial optimization	Greedy search Beam search

Table 3.6: Optimization techniques for finding the optimal classifier in the space of possible classifiers.

3.3.1.1 Logistic regression

In this thesis logistic regression, as implemented in scikit-learn.org, Pedregosa et al. [2011], was used for classification. Reasons for choosing it were:

- It is a standard linear classifier to begin with, before moving on to more complex models that might not be necessary to explain and separate the data at hand.
- It can be trained comparatively quickly and does not take long for classification even in the face of a large feature space as is the case here.
- It comes with free feature ranking as a treat.

Despite its name, logistic regression is a statistical model that can be used for classification. It is part of a group of models called *generalized linear models* in which the target value, also known as *class*, is expected to be a linear combination of the input values. Mathematically they generally solve a problem of the form:

$$\hat{y}(\beta, x) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \tag{3.12}$$

The dependent variable, i.e. the prediction $\hat{y}$, is usually dichotomous or binary, meaning it can take one of two values, 0 or 1; though multinomial versions exist. This type of variable is called a *Bernoulli variable*, which has the property that if the target variable takes the value 1 with a probability p, the probability of value 0 is $1 - p$. The independent variables, on the other hand, can take any form – no assumptions about the distribution of the input variables are made. In the case of logistic regression, the function solving for the relationship between input and target variables is not a linear function as in Equation 3.12 but the logistic regression function[14]:

$$\begin{aligned} log\Big(\frac{p(x)}{1-p(x)}\Big) &= \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \qquad (3.13)\\ &= \beta_0 + \boldsymbol{x} \cdot \boldsymbol{\beta} \end{aligned}$$

Solving this for p, the predicted probability that the input values belong to class 1, gives:

$$p(x; \beta_0, \beta) = \frac{e^{\beta_0 + x \cdot \beta}}{1 + e^{\beta_0 + x \cdot \beta}} \tag{3.14}$$

[14] Matrix notation included in bold face. I will omit the bold face in the following and presume that variables without indices are vectors or matrices where appropriated by context.

which is equivalent to

$$p(x; \beta_0, \beta) = \frac{1}{1 + e^{-(\beta_0 + x \cdot \beta)}} \tag{3.15}$$

This means, logistic regression can be used as a linear classifier with the decision boundary $\beta_0 + x \cdot \beta = 0$ separating the two predicted classes. In terms of the categorization set forth in the previous section this means logistic regression operates in a hyperplane-based hypothesis space. Its objective function is maximum likelihood as it tries to maximize the likelihood of the data for a predicted class. It requires solving a convex optimization problem, for which efficient algorithms exist, see for instance Lee et al. [2006].

3.3.2 Clustering

Clustering is probably the most widely used technique in unsupervised learning, but it is not the only one: self-organizing maps, hidden Markov models, and blind source separation are among other methods that populate the paradigm. Alas, the most popular approach is the one I shall stick to here as it lends itself formidably to separating a heap of word pair representations into smaller heaps of more similar pairs.

The goal of clustering is to find *intrinsic* categories in data. This is achieved by dividing a data set into subsets (*clusters*) such that objects in the same subset are more similar to each other than to objects in other subsets with respect to a given similarity measure, Kriegel et al. [2009]. Conversely, one could say that clustering aims at grouping objects such that *intra-cluster* distance deceeds *inter-cluster* distance – for some definition of distance.

Figure 3.6, taken from Kerdels and Peters [2014], exemplifies the situation. Depicted are several data sets, some of which typically serve as benchmarks for testing the suitability and generality of clustering algorithms. For the majority of them the underlying intrinsic structure of the data is apparent to the human eye and the algorithm used concurs. Its clusterings are shown in color[15]. Moreover, subplot (c), a 2-D projection of the high-dimensional Swiss Roll data set, illustrates the crucial role of the distance measure used to the quality of a clustering: “points far apart on the underlying manifold, as measured by their geodesic, or shortest path, distances, may appear deceptively close in the high-dimensional input space, as measured by their straight-line Euclidean distance.” Tenenbaum et al. [2000].

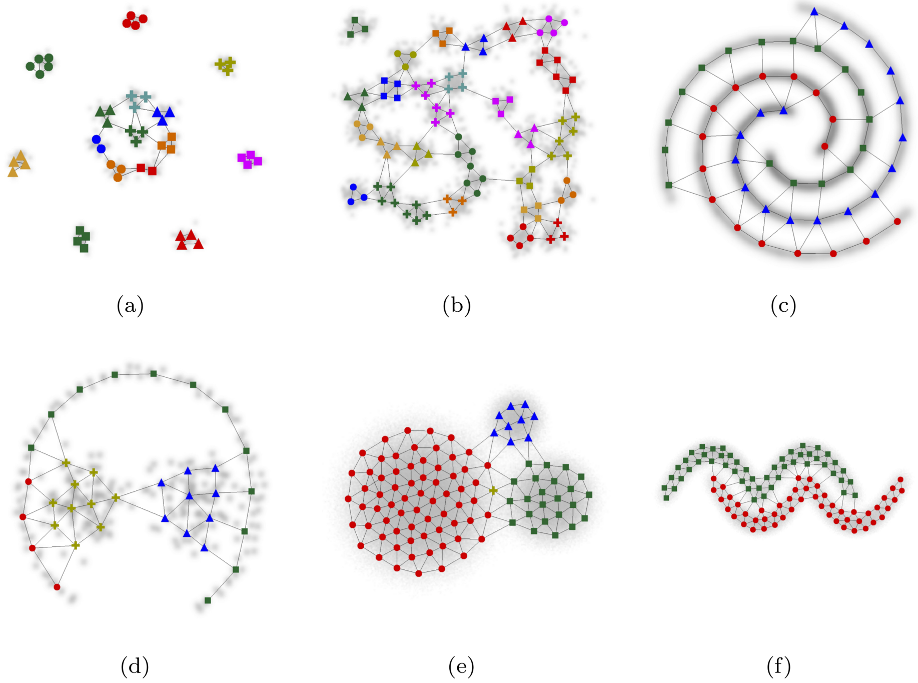
One way to categorize clustering algorithms is by the type of input and output they take, see, e.g., Murphy [2012]. In *feature-based clustering* the input is a $N \times D$ feature matrix and in *(dis)similarity-based clustering* the input is a $N \times N$ distance or similarity matrix – where N is the number of samples and D the number of features[16]. Apart from these different inputs, there are also two possible types of output:

- flat clustering, also called partitional clustering
- hierarchical clustering

A **flat clustering** is a partition of the input space into disjoint sets. A **hierarchical clustering**, on the other hands, is a nested tree of partitions. See Figure 3.7 for a visualization of the

[15] The algorithm used was a growing neural gas with local input space histograms.

[16] See sections 3.2.1 and 3.2.3 for how these matrices were constructed.


(a) (b) (c)

(d) (e) (f)

Figure 3.6: Various data sets with different intrinsic structures; figure from Kerdels and Peters [2014].

difference. Hierarchical clustering basically allows the user to pick the level of resolution from the completed clustering whereas in partitional clustering the number of clusters usually has to be decided upon ahead of computation time. Respectively, flat clusterings are often computationally cheaper to compute ($\mathcal{O}(ND)$) than hierarchical clusterings ($\mathcal{O}(N^2 \log N)$), Murphy [2012].

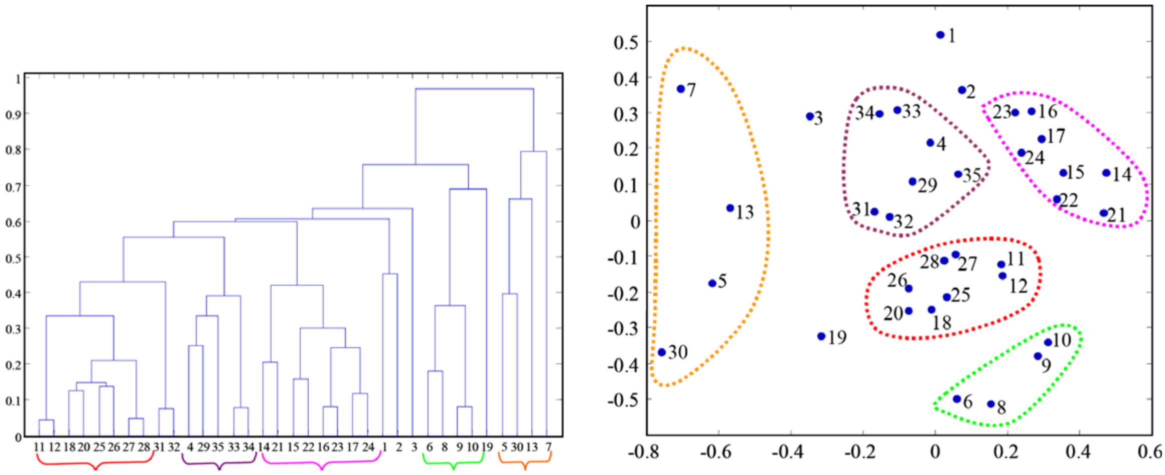


Figure 3.7: Hierachical (left) versus flat clustering of a data set; figure from Jain [2010].

Clusterings can furthermore be distinguished in terms of how cluster membership is enforced. A *complete* clustering assigns every object to a cluster, whereas a *partial* clustering may leave objects unassigned. Moreover, cluster membership can be assigned in *exclusive*, *overlapping*, or *fuzzy* manner. This means that every object is assigned either to a single cluster (*exclusive*), to one or more clusters (*overlapping*), or to every cluster with a membership weight between 0.0 (does not belong) and 1.0 (fully belongs)(*fuzzy*), Pang-Ning et al. [2006].

Since there is no clear-cut definition of the notion of a cluster (see Estivill-Castro [2002]), there exist a variety of cluster models. I will give a brief overview of the most prominent ones in Table 3.7 before describing the model used in this work in more detail.

Cluster model	Example algorithm
Connectivity-based clustering	agglomerative clustering
Prototype-based clustering	k-means
Density-based clustering	DBSCAN
Distribution-based clustering	Latent Dirichlet Allocation (LDA)
Graph-based clustering	Chinese Whispers

Table 3.7: Cluster models and implementations thereof.

K-means, agglomerative clustering and DBSCAN are probably the most commonly used and implemented algorithms. ***K*-means** finds linearly separated clusters and prefers compact and isolated clusters because the similarity measure is based on the Euclidean distance. Furthermore the number of clusters has to be specified as a parameter to the algorithm. This lead to dismissing it as a feasible option in this thesis because the Euclidean distance is not very useful in high-dimensional and highly sparse spaces and compactness can not be guaranteed or even assumed when dealing with natural language. **Agglomerative clustering**, on the other hand, while highly useful in NLP, has a complexity of $\mathcal{O}(N^3)$ in the general case, making it too slow for large data sets. **DBSCAN** is a density-based algorithm with a runtime “slightly higher than linear in the number of points”, Ester et al. [1996]. Even though this makes it a promising candidate for a large data set like the one at hand, I decided in favor of **Chinese Whispers** for three reasons:

- DBSCAN requires a global density parameter which is used as a threshold to determine the reachability distance between data points. However, different regions in the data may have different densities and as a result a single density parameter might make it difficult to effectively find the clusters, cf. Aggarwal and Reddy [2013], page 459.
- Density-based methods are naturally defined on data points in a continuous Euclidean space, which means they often cannot be used meaningfully in a discrete or non-Euclidean space, unless the data are embedded first, cf. Aggarwal and Reddy [2013], page 7.
- Chinese Whispers has been shown to yield good results on various problems from natural language processing like word sense disambiguation, language separation, and acquisition of syntactic word classes, cf. Biemann [2006].

All of the above make Chinese Whispers appear like the more suitable algorithm for the discrete data with unknown density behavior that I am dealing with. But let me present the algorithm in more detail.

3.3.2.1 Chinese Whispers

Chinese Whispers (CW) is a randomized graph-clustering algorithm which can be considered a special case of Markov-Chain Clustering, Biemann [2006]. Its runtime complexity is linear in the number of entries in the similarity matrix. Data points are represented as vertices in a graph and relations, here: similarity, as edges between data points. The degree of similarity is applied as weight to the edges; think a numeric label. Such a graph can be formalized as an ordered pair $G = (V, E)$ comprising a set V of vertices together with a set E of edges. If the graph is *undirected*, as is the case in Version 1.0.1 of CW used[17], then a weighted edge $(v_i, v_j, w_{ij}) \in E$ between vertices v_i and v_j implies that there is also a symmetric edge $(v_j, v_i, w_{ij}) \in E$.

The basic idea behind Chinese Whispers is to assign each vertex its own cluster ID at first, and then, iteratively, make them adopt the cluster ID that is prevalent in the local neighborhood. Prevalence is computed as the sum of weights of the edges connecting to a vertex, optionally downgraded by the degree of these edges. This leads to a stabilization of regions which expand over iterations until they encounter other regions, Biemann [2006]. Note that while convergence is not guaranteed and hence the number of iterations has to be preset, clusters stop changing after a few tens of iterations, unless there is a tie. A disadvantage of CW is that its output is non-deterministic which is the result of the randomized and continuous manner in which cluster membership is determined per iteration. However, its favorable runtime complexity and hence ability to handle large amounts of data as well the produced clusterings make it a good choice for clustering natural language data. For further details, please refer to Biemann [2006].

3.3.2.2 Cluster evaluation

Two main approaches to measuring the quality of a clustering exist: If *ground truth* is available, extrinsic methods can be used, which compare a clustering against the ground truth and assign a score to the clustering. Ground truth, also known as *gold standard*, is an ideal clustering, usually designed by human experts. If ground truth is unavailable, intrinsic methods can be used, which evaluate the quality of a clustering by how well the clusters are separated, cf. Han and Kamber [2006], Chapter 10.

Extrinsic methods can be distinguished by which and how many of the following criteria they satisfy, Amigó et al. [2009]:

- cluster homogeneity
- cluster completeness
- rag bag
- small cluster preservation

Cluster homogeneity deals with cluster purity: the purer the clusters in a clustering, the better the clustering. Cluster completeness assigns a better score to a clustering, if objects that belong to the same category according to ground truth are assigned to the same cluster. The rag bag criterion penalizes putting an object of a different category into an otherwise pure cluster and requires that it rather be added to a cluster of miscellaneous, left over objects. Small cluster preservation promotes the preservation of small clusters at the expense of splitting large clusters, where the choice arises. Examples of extrinsic measures for cluster evaluation are the *v-measure*,

[17] see HTTP://WORTSCHATZ.INFORMATIK.UNI-LEIPZIG.DE/~CBIEMANN/SOFTWARE/CW.HTML

which is the harmonic mean of homogeneity and completeness, the *adjusted rand index* and the *Jaccard coefficient*, which also satisfy the first two criteria, and *BCubed*, which satisfies all four criteria.

More precisely, **homogeneity** is a score between 0.0 and 1.0, where a value of 1.0 means a perfectly homogeneous labeling and 0.0 indicates complete lack of homogeneity. It is a measure of entropy that computes the mutual information of true and predicted labels scaled by the entropy of the true labels.

The **BCubed** measure is based on the following definition of *correctness*:

$$Correctness(e, e') = \begin{cases} 1 & \text{iff } L_e = L_{e'} \iff C_e = C_{e'} \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

That is, two data points e and e' that share the same label L are correctly related if and only if they appear in the same cluster C. Given complete, exclusive clustering and single labels, *precision* and *recall* for a single data point are:

$$Precision_e = \frac{|Class_{L_e} \cap Cluster_{C_e}|}{|Cluster_{C_e}|} \tag{3.17}$$

$$Recall_e = \frac{|Class_{L_e} \cap Cluster_{C_e}|}{|Class_{L_e}|} \tag{3.18}$$

Over all data points this can be formulated as follows, Amigó et al. [2009]:

$$Precision_{BC} = Avg_e\,[\,Avg_{e'.C_e=C_{e'}}\,[\,Correctness(e, e')\,]] \tag{3.19}$$

$$Recall_{BC} = Avg_e\,[\,Avg_{e'.L_e=L_{e'}}\,[\,Correctness(e, e')\,]] \tag{3.20}$$

Plugging $precision_{BC}$ and $recall_{BC}$ in the harmonic mean F-score, we get

$$F_1 = 2 * Precision_{BC} * Recall_{BC} \,/\, (Precision_{BC} + Recall_{BC}) \tag{3.21}$$

Clusterings in this work will be evaluated with BCubed and homogeneity using BLESS labels as ground truth; see Section 4.2.2 for reasoning and results.

Intrinsic methods If, on the other hand, reference to ground truth is unavailable, intrinsic methods can be used to evaluate a clustering. These evaluate a clustering by determining how well separated and how compact its clusters are. In particular, the inter-cluster and intra-cluster distances play a crucial role here. The probably most well-known intrinsic measure is the *silhouette coefficient*, but there are others, e.g., the *Dunn index*. The silhouette coefficient is defined as

$$s(o) \equiv \frac{b(o) - a(o)}{max\,\{\,a(o), b(o)\,\}} \tag{3.22}$$

where $a(o)$ is the average distance between an object o and all other objects in the same cluster, which reflects the compactness of the cluster. And $b(o)$ is the minimum average distance from

o to all clusters o does not belong to, which reflects the degree of separation of o from other clusters. Formally that amounts to

$$a(o) \quad = \quad \frac{\sum_{o' \in C_i, o' \neq o} dist(o, o')}{|C_i| - 1} \tag{3.23}$$

$$b(o) \quad = \quad min_{\;C_j : 1 \leq j \leq k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|} \right\} \tag{3.24}$$

where C_i is the cluster that object o belongs to, C_j any of the other clusters, and k the total number of clusters. To score a whole clustering with the silhouette coefficient, we take the average over the silhouette coefficient as computed for every clustered object.

After having taken a look at the various options of cluster evaluation, in the next section I take a step back and suggest several ways to evaluate the distributional semantic model, that has been presented in this chapter, as a whole.

3.4 Evaluation

It is good practice in computational linguistics to validate a system both in terms of *intrinsic* and *extrinsic* evaluation (Baroni and Lenci [2011])[18]. Intrinsic evaluation refers to the process of testing a system in itself, often with respect to some gold standard; for instance, when evaluating the word space spanned by a distributional model or computing the overall loss of a predictor. Extrinsic evaluation is about measuring a system's performance in a specific task or embedded application; e.g. how well a distributional predictor performs on the multiple-choice questions of the Scholastic Aptitude Test (sat), a standardized test commonly used for college admission in the United States of America. In the following I describe the data sets used for intrinsic and extrinsic evaluation in this work, namely the bless data by Baroni and Lenci [2011] and the sat data set as provided by Turney [2011].

3.4.1 Intrinsic evaluation

To counteract the advancing fragmentation of data sets and evaluation metrics used for intrinsic evaluation in distributional semantics, Baroni and Lenci [2011] devised a data set specifically designed for evaluating distributional models, including compositional distributional models. Their intention was to promote comparability of studies and advance scientific progress[19]. The bless data set consists of 200 nouns, both animate and inanimate, from different categories including tools, vehicles, animals, etc. For each noun there is a set of other words that are associated in one of the following relations:

- hypernymy
- co-hyponymy
- meronymy
- typical attribute
- typical related event

[18] The terms *intrinsic* and *extrinsic* have slightly different meanings in this section than the previous one. Yet, both refer to the original senses of *internally, in itself* and *externally, from the outside.*

[19] see also HTTPS://SITES.GOOGLE.COM/SITE/GEOMETRICALMODELS/SHARED-EVALUATION

· random

Those other words can be either nouns, verbs or adjectives. For instance, the noun *alligator* could be paired with the adjective *aquatic* which is considered a typical attribute or with the noun *carnivore* which is a hypernym of it, and so forth. A random pairing of *alligator* could be with *teenager* or *propel.* Table 3.8 lists a few examples in the original data formatting, though exclusively paired with nouns since these are the only considered kind of words here.

NOUN	CATEGORY	RELATION	OTHER NOUN
lizard-n	amphibian_reptile	coord	chameleon-n
		hyper	animal-n
		mero	blood-n
		random-n	majesty-n
dishwasher-n	appliance	coord	freezer-n
		hyper	artifact-n
		mero	button-n
		random-n	dentist-n
sword-n	weapon	coord	missile-n
		hyper	device-n
		mero	pommel-n
		random-n	annihilation-n

Table 3.8: Data samples from the BLESS evaluation set.

The BLESS data set consists of a total of 26 554 labeled pairings, thereof 14 547 noun-noun pairings. Of these 1 337 are in a hypernym relation, 3 565 are co-hyponyms, 2 943 meronyms, and 6 702 random pairings. There are 200 unique primary nouns, that would be the nouns in the first column in Table 3.8, and 5 676 unique secondary nouns (the fourth column). Primary nouns occur between 37 and 147 times as can be seen in the top left histogram in Figure 4.3 which displays the distribution of overall occurrences of primary nouns. In total, 6 414 of 14 547 noun pairs from BLESS were found in the data extracted from PukWaC.

The BLESS data was preferred over other candidates like the TOEFL data (Landauer and Dumais [1997])[20], SemEval 2012 Task 2 (Jurgens et al. [2012]) or SemEval 2014 Task 1 (Marelli et al. [2014]) data sets for various reasons: For one, they are a reasonable size, whereas, for instance, the TOEFL data consist only of 80 multiple-choice synonym questions. In contrast to SemEval 2012 Task 2, BLESS uses the type of semantic relations we were interested in rather than deploying a whole taxonomy of semantic relations ranging from *space–time* to *agent–object* and *cause–effect.* Lastly, SemEval 2014 Task 1 operates on a different level of analysis, looking at the sentence level rather than single nouns or multi-word expressions.

3.4.2 Task-oriented evaluation

On the side of extrinsic evaluation, there have long been a variety of tasks and respective data sets in use for testing distributional models. These include:

[20] Note that the TOEFL data can be used both as a (mini) gold standard for synonyms as well as an extrinsic evaluation task in which performance is measured in comparison to humans.

Task	Example(s)
Vocabulary tests	ESL, SAT
Thesaurus comparison	correlation with graph distance in WordNet, overlap of word space and thesaurus
Behavioral tests	association norms, semantic priming

Table 3.9: Tasks for extrinsic evaluation in distributional models.

Among the more widely used ones is the SAT data set, Turney [2011], which has been used in more than 20 studies[21]. It consists of 374 multiple-choice analogy questions with 5 choices per question. Sample data are shown in Table 3.10. The data was originally collected by Michael L. Littman and is available on request from Peter Turney[22] who started using them for measuring relational similarity in 2003.

Question	Answers
ostrich : bird	lion : cat goose : flock ewe : sheep cub : bear primate : monkey
tunnel : mine	conduit : fluid corner : intersection sign : detour aisle : seat corridor : building

Table 3.10: Data samples from the SAT evaluation set.

The SAT data is the evaluation set I used in this study as well, as it turned out that alternative tasks were not as well suited. Thesaurus comparisons, in particular with WordNet, are complicated in the given setting because the degree of detail and hence variance in graph distance diverge considerably for different domains. Behavioral experiments were forbone due to the organizational and financial efforts involved in inviting a significant number of human subjects to participate.

[21] cf. HTTP://ACLWEB.ORG/ACLWIKI/INDEX.PHP?TITLE=SAT_ANALOGY_QUESTIONS

[22] ibidem.

4 Results and Discussion

In the previous chapter I described the methods used to learn semantic relations with distributional similarity. In this chapter I present the results, illustrate how well semantic relations could actually be learned, discuss issues encountered and analyze the errors. As before, the topical order is: representation, then learning, and finally error analysis.

4.1 Features

Pipeline Table 4.1 shows the NLP pipeline – as depicted in Section 3.1, Figure 3.1 – viewed as a combination of filters over incoming data points. The number of incoming sentences is reduced by roughly 9 percent when removing duplicate sentences. Filtering these unique sentences with a distributional thesaurus yielded 200 069 unique pairs; using BLESS instead yielded 6 414 and 6 718 unique pairs for PukWaC and News120M, respectively. Note that numbers for the DT track are given for roughly ten percent of the total data, since the complete corpus exhausted the available computational resources. This is also the reason the DT was applied to only one corpus, namely PukWaC. The JoBimText records contain more unique pairs than the feature records because inverted pairs were added[1] with the respective pattern marked as $\langle \textit{pattern} \rangle^{-1}$. For instance, if the pair *(capitalism,war)* was observed with the pattern ⟨*X requires Y*⟩, we added the inverted pair *(war,capitalism)* with the pattern $\langle \text{X requires Y} \rangle^{-1}$.

	PukWac		News120M
	DT	BLESS	BLESS
Sentences in input corpus	88 214 600		88 942 335
Unique sentences in corpus	76 020 095		81 026 917
Pairs in DT resp. BLESS	36 070 044	14 547	
Pairs in feature records	200 069	6 414	6 718
Pairs in JoBimText records	341 339	11 216	11 570
Patterns in feature records	19 389 359	3 275 482	4 383 134
Patterns in JoBimText records	38 778 750	6 550 964	8 766 266
Frequent patterns ($> 5\times$)	470 798	65 148	41 572

Table 4.1: The pipeline viewed as a set of filters over incoming data points.

As the stark shrinkage of patterns with frequency shows, the majority of patterns occurs rarely and might therefore not contribute to finding similarities between pairs for lack of overlap. This will be explored further when looking at distributions of patterns and pairs, their intertwining, and at pruning. It is worth noting that, although the BLESS data was generated from PukWaC, it is not the case that more BLESS pairs were found in that source corpus than in News120M.

[1] JoBimText records do not necessarily double counts because sometimes inverted pairs already existed.

Distributional thesaurus Table 4.2 shows how the final distributional thesaurus is arrived at over several filtering steps. Thirty percent of the original entries involve pairs of common nouns and most of these are not the same noun, i.e. not pairs like *(year,year)*. A considerable number of nouns contain numbers and shtrudels which might be an indicator of boiler plate still present in the data. After filtering these the DT still consisted of about 90 million pairs. For further data reduction, three pruning factors were experimented with. A cap of 50 expansions per target word was finally chosen to maximize distributional similarity of pairs while avoiding unnecessary data loss. Restricting expansions to the top 25 per target word reduced the overlap with BLESS to a mere 395 training and test pairs while using the top 50 yielded 6 414 pairs. Unfortunately, using more expansions per target word would have exhausted the available computational resources.

	PukWaC
All target-expansion pairs	449 688 161
only NN and NNS	137 859 295
target $\neq$ expansion	136 130 697
skip words with [0-9.+@]	91 796 055
Top 100 expansions	61 703 722
Top 50 expansions	36 070 044
Top 25 expansions	19 934 687

Table 4.2: Target-expansion pairs in distributional thesaurus.

Patterns Extracted patterns followed a typical power law distribution with many patterns occurring very rarely and few patterns occurring very frequently, see top histogram in Figure 4.1. Extremes were 17 572 379 patterns observed a single time and the pattern $\langle X$ *and* $Y\rangle$ observed 100 955 times. In complement to Figure 4.1, Table 4.3 lists the 50 most frequent patterns across all pairs. The selection of patterns shows that there is ample repetition in patterns, particularly involving commas. For instance, one can find all of $\langle X\ ,\ Y\rangle$, $\langle X\ Y\ ,\rangle$, $\langle X\ ,\ Y\ ,\rangle$ and $\langle X\ ,\ ,\ Y\rangle$ in these top patterns. This is mainly because all subtrees along the dependency parse of length 6 or less were extracted as patterns thereby fostering repetition with slight variance. This study did not empirically test how the inclusion of punctuation marks, and in particular commas, effected the results, which might be worth examining in future work. However, some patterns including commas ended up having high coefficients for predicting cohypernymy, cf. Table 4.25. It is interesting, though not surprising, that the most frequent patterns contain only determiners, prepositions, and conjunctions besides X and Y, and no verbs, apart from the occasional inflected form of *to be*.

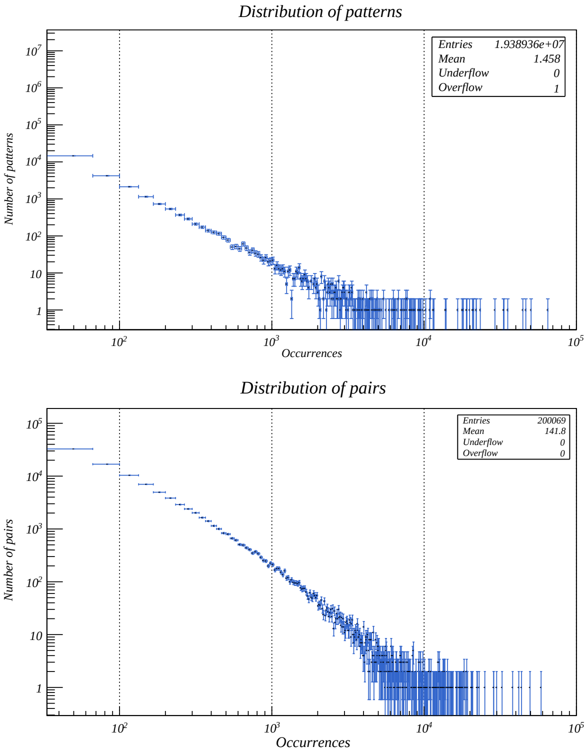


Figure 4.1: Histograms of pattern and pair frequencies in feature records.

OCCURRENCE	PATTERN	OCCURRENCE	PATTERN
100 955	X and Y	11 463	X of the Y
64 929	X , Y	10 984	X and Y of
50 429	the X Y	10 969	X Y , ,
46 539	X Y and	10 884	on X Y
44 528	of X Y	10 256	X Y is
35 418	X Y ,	9 808	X , , Y and
33 485	X , Y and	9 558	with X and Y
33 202	X of Y	9 436	the X Y of
29 176	a X Y	9 428	X Y in
23 371	X or Y	9 248	X the Y
22 002	X , Y ,	9 140	X in Y
21 999	X , and Y	8 755	for X and Y
21 154	the X and Y	8 653	X for Y
21 107	of X and Y	8 503	of the X Y
20 775	X , , Y	8 349	of X Y and
19 890	with X Y	8 297	as X Y
19 126	X Y of	8 162	X Y for
18 939	for X Y	8 069	is X Y
18 080	in X Y	8 005	X a Y
18 064	X Y , and	7 753	X and Y ,
17 772	the X of Y	7 735	X , Y , ,
16 616	to X Y	7 708	a X of Y
13 934	X , Y , and	7 578	the X of the Y
13 791	X Y are	7 576	X , , and Y
11 609	of X , Y	7 558	by X Y

Table 4.3: The 50 most frequent patterns in the feature records.

Pairs Pair occurrences behaved similar to those of patterns on a large scale, although the mean was higher by a factor of 100, see the bottom histogram in Figure 4.1. The most frequently observed pair was *(web,site)* with 58 645 counts and as many as 16 714 pairs were seen only a single time. The 20 most frequent noun pairs in the feature records are listed in Table 4.4.

OCCURRENCE	PAIR	OCCURRENCE	PAIR
58 645	(web, site)	20 543	(web, page)
43 646	(man, woman)	19 046	(click, link)
32 517	(term, condition)	19 043	(search, engine)
28 666	(application, form)	18 620	(anwer, question)
28 611	(day, week)	18 061	(click, button)
25 041	(name, address)	17 777	(research, project)
22 991	(email, address)	17 412	(credit, card)
22 378	(product, service)	17 356	(family, friend)
22 255	(hour, day)	17 219	(group, people)
20 845	(time, day)	17 046	(member, staff)

Table 4.4: The 20 most frequently observed pairs in the feature records.

Pairs-per-pattern, patterns-per-pair Before applying machine learning algorithms to the data, pairs-per-pattern and patterns-per-pair distributions were inspected to analyze whether the features used produced enough overlap between data points for similarity measures to make sense. The histograms in in Figure 4.2 show the bigger picture. Zooming in revealed that as many as 10^7 patterns occurred only with a single noun pair but just 9 651 pairs occurred only with a single pattern.

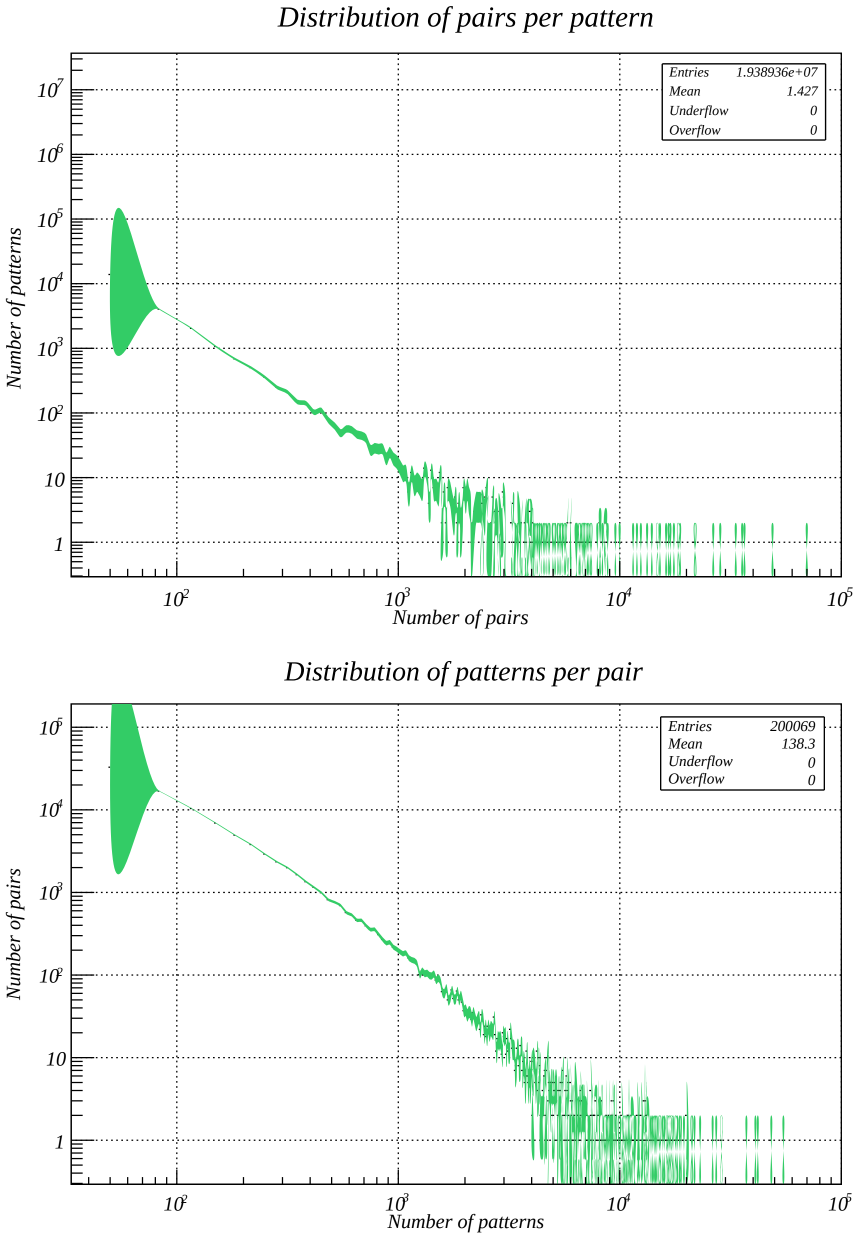


Figure 4.2: Histograms of pairs per pattern and patterns per pair in feature records.

4.2 Learning

4.2.1 Classification

A variety of parameters were tuned for classification, among them pattern pruning, different ways of splitting training and test data, binary versus continuous values for the feature matrix, varying subsets of BLESS for evaluation, and various combinations of feature sets. In the following paragraphs I present the outcomes and discuss them one by one, the upshot though is the following:

Predictions were best for co-hyponyms with an F-score of 0.90, slightly less reliable but still fairly good for meronyms with 0.79, and most difficult for hypernyms with 0.56. The feature that proved most contributive was subtree patterns containing both nouns, extracted from dependency parses.

Pruning Strong pruning did not produce positive effects on the classification results as measured by the F-score. Mild pruning, however, improved the results by some percentage points for predictions regarding both hypernymy and meronymy. Thereby only context features were pruned, choosing the n most informative contexts per noun pair measured with the Lexicographer's mutual information as described in Section 3.2.2. Similarity and topic modeling features were kept uncut. Table 4.5 shows the development of the F-score and one can observe how F-scores decline once pruning reduces the data to 500 patterns per pair and below. For one of the conditions a comparison was run with the same amount of patterns but randomly chosen from all context patterns and it seems to be the case that selecting patterns by their LMI ranking returns better results than just any patterns. To get an impression of the magnitudes, Table 4.6 shows the counts of (non-unique) patterns for different degrees of pruning.

	COORD	HYPER	MERO
No pruning	0.89	0.53	0.75
Top 1500 context patterns per pair	0.89	0.51	0.76
Top 1000 context patterns per pair	0.89	0.54	0.76
Top 500 context patterns per pair	0.87	0.53	0.76
Top 200 context patterns per pair	0.87	0.47	0.75
Some 200 context patterns per pair	0.80	0.31	0.70

Table 4.5: Comparison of F-scores for different amounts of pruning.

	Non-unique patterns
All context patterns for BLESS	25 758 378
Top 1500 context patterns per pair	21 293 362
Top 1000 context patterns per pair	20 602 498
Top 500 context patterns per pair	19 554 526
Top 200 context patterns per pair	8 637 714

Table 4.6: Counts of context patterns with different amounts of pruning.

De-lexicalizing As Table 4.7 distinctly marks, there is a considerable gap in classification results between lexicalized and de-lexicalized data. The difference between the two conditions is that in the lexicalized case the data are split into training and test sets using a standard procedure like stratified folds, whereas in the de-lexicalized case additionally care is taken to avoid vocabulary overlap in test and training sets. Overlap in vocabulary can lead to learning that a particular word, for instance, *animal*, is a typical hypernym and subsequently make predictions based on this information, which does not generalize very well.

		lexicalized	de-lexicalized
COORD	Precision	0.96	0.95
	Recall	0.96	0.83
	F-score	0.96	0.89
	Support	518	633
	Avg. precision	0.97	0.92
	Accuracy	0.97	0.92
HYPER	Precision	0.99	0.85
	Recall	0.88	0.40
	F-score	0.93	0.54
	Support	153	160
	Avg. precision	0.94	0.66
	Accuracy	0.98	0.93
MERO	Precision	0.93	0.82
	Recall	0.90	0.70
	F-score	0.91	0.76
	Support	440	593
	Avg. precision	0.93	0.82
	Accuracy	0.94	0.84

Table 4.7: Comparing classification reports for lexicalized and de-lexicalized training and test sets.

This seems to be a particular problem with the BLESS evaluation data, where the left nouns in a pair occur in all four relations but most of the right nouns occur only in one of the four relations, see the lower two histograms in Figure 4.3. 5 266 out of 5 676 unique right-hand nouns occur only in one relation, 410 in more than one relation. The word *artifact*, for instance, occurs in a hypernym relation 91 times. All in all, 1746 nouns are seen in the same relation more than once, 163 more than ten times, 37 more than twenty times, and 6 more than 50 times. The number of occurrences per left-hand and right-hand noun are shown in the upper histograms in Figure 4.3.

Binary features Comparing binary feature matrices to continuous ones showed that the crucial piece of information is whether or not a feature was observed significantly with a particular noun pair rather than the exact LMI value. Table 4.8 illustrates this conclusively.

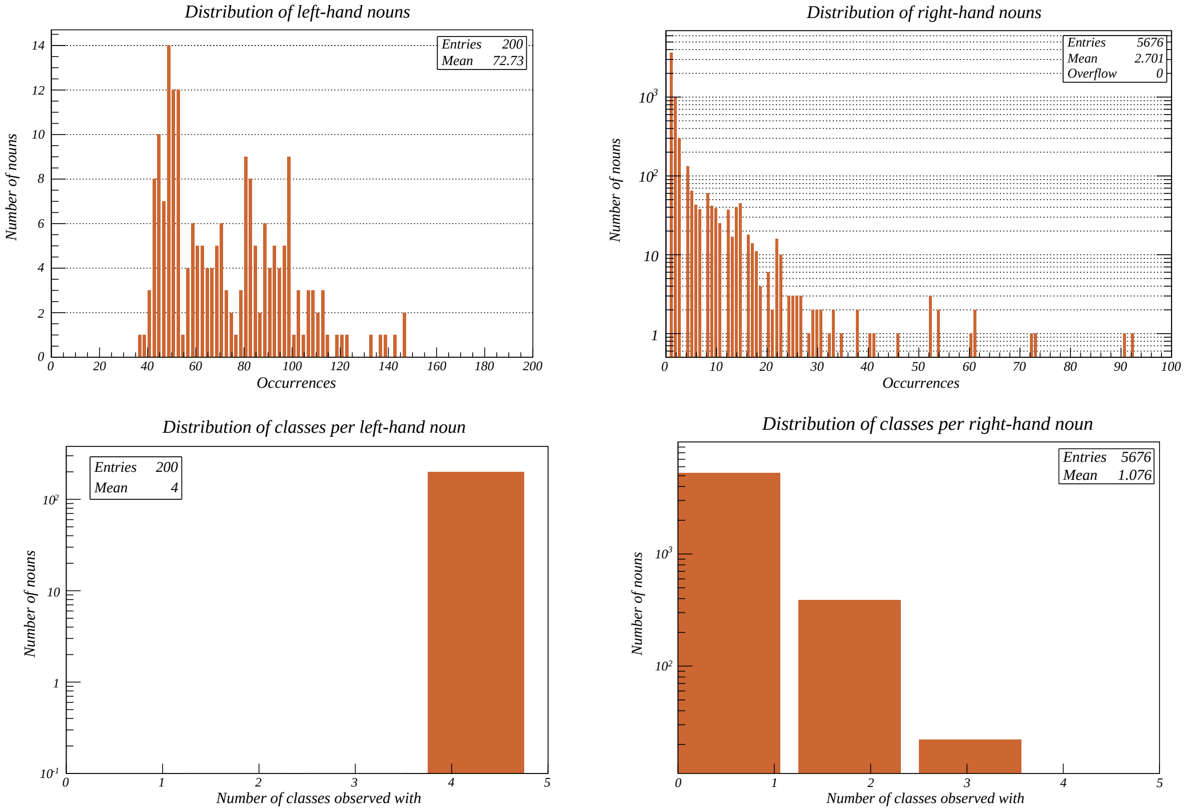


Figure 4.3: Histograms of the distributions of nouns in the BLESS data.

	COORD			HYPER			MERO		
	P	R	F	P	R	F	P	R	F
binary	0.95	0.83	0.89	0.85	0.40	0.54	0.82	0.70	0.76
continuous	0.94	0.63	0.75	0.77	0.39	0.52	0.83	0.45	0.58

Table 4.8: Comparison of F-scores for binary versus continuous feature matrix.

PukWaC versus News120M To compare the two data sources PukWaC and News120M, F-scores were computed for both in a basic setting with a pruning of 1000 patterns per pair, de-lexicalized training and test sets, and binary feature matrices. Since similarity and topic modeling features were not available for News120M, the comparison was run with context features only. The BLESS data without random pairs was used for training and testing to make the comparison as fair as possible – see the next paragraph for details on the different evaluation sets. Support was the same for both corpora with 1 004 co-hyponyms, 454 hypernyms, and 764 meronyms. The results for co-hyponymy and hypernymy are in the same ball park, but for meronymy the F-score is noticeably better with representations derived from News120M. This is probably because many more patterns were found for meronyms in the News120M corpus than in PukWaC, making predictions more reliable: for 2943 meronym pairs, on average 269 more patterns per pair were found in News120M; i.e. an average of 882 patterns per pair instead of 614.

	COORD			HYPER			MERO		
	P	R	F	P	R	F	P	R	F
PukWaC	0.93	0.65	0.77	0.95	0.24	0.38	0.73	0.56	**0.63**
News120M	0.86	0.66	0.75	0.91	0.28	0.42	0.60	0.82	**0.70**

Table 4.9: Comparing F-scores for PukWaC and News120M; context features only.

Evaluation sets When training and testing the logistic regressor with different subsets of the BLESS data, F-scores varied considerably with differences of up to 27 percentage points. This effect occurs because various features are combined for learning and, if for a particular data point some features have been observed but not others, a prediction will still be made unless no feature at all has been seen. In particular, features involving only single nouns are much more abundant than features requiring the presence of both nouns in a pair. At the same time the subtree feature, which is a pair feature, was the best tested predictor of the relation between two nouns, see the ablation test in Table 4.13. The significant decrease in recall and slight increase in precision, which were observed for larger evaluation sets and smaller overlap with PuKWaC pairs, can therefore be explained in terms of the contribution of the subtree feature to the model: In its presence, the number of correctly predicted from the relevant class, $\frac{TP}{TP+FN}$, is expected to grow, while in its absence but with generally higher support the number of correctly predicted from retrieved items, $\frac{TP}{TP+FP}$, can still be expected to remain stable[2].

The three different evaluation sets tested were: "all BLESS pairs" comprising all noun pairs from the BLESS data, "BLESS exc. random" skipping random pairs from that original set, following the logic that non-distributionally similar pairs rarely occur in the data set of this study. Lastly, "BLESS ∩ PukWaC" explicitly filtered out any noun pairs that had not been observed in the data. As Table 4.10 shows, the decrease in F-scores is mostly due to a decrease in recall.

[2] Slightly higher support in some conditions with generally less data are artifacts of de-lexicalization.

		ALL BLESS PAIRS	BLESS EXC. RANDOM	BLESS ∩ PUKWAC
ALL CTX FEATURES				
COORD	Precision	0.97	0.93	0.93
	Recall	0.50	0.65	0.76
	F-score	0.66	0.77	0.84
	Support	1021	1004	663
HYPER	Precision	0.94	0.94	0.83
	Recall	0.12	0.24	0.34
	F-score	0.22	0.38	0.49
	Support	394	454	160
MERO	Precision	0.89	0.73	0.82
	Recall	0.28	0.56	0.56
	F-score	0.43	0.63	0.67
	Support	769	764	593
SUBTREE FEATURE ONLY				
COORD	Precision	0.94	0.90	0.93
	Recall	0.74	0.84	0.78
	F-score	0.83	0.87	0.85
	Support	702	802	633
HYPER	Precision	0.80	0.89	0.78
	Recall	0.43	0.45	0.40
	F-score	0.56	0.60	0.53
	Support	199	280	160
MERO	Precision	0.83	0.69	0.82
	Recall	0.37	0.89	0.59
	F-score	0.51	0.78	0.68
	Support	577	548	593

Table 4.10: Comparing precision, recall, and F-score for different evaluation sets.

Class priors The prior distributions of classes in the evaluation data are given in Table 4.12, where hypernyms were clearly the least represented, while co-hyponyms were on a par with meronyms for all evaluation sets. It is surprising that 1 034 random pairings were seen in the noun pairs from the distributional thesaurus and should be examined further in future work. Table 4.11 gives some examples of such random pairings.

RANDOM PAIRS
(jet, jam)
(mug, ward)
(fridge, test)
(trout, hour)
(shirt, legend)
(cod, bed)
(fighter, piece)
(truck, visitor)
(truck, counterpart)
(gorilla, cash)
(cannon, platform)
(beetle, being)
(hat, meal)
(washer, tab)
(toaster, show)

Table 4.11: Selection of random pairs seen in DT.

	COORD	HYPER	MERO	RANDOM
all BLESS pairs	3 565	1 337	2 943	6 702
BLESS exc. random	3 565	1 337	2 943	–
BLESS ∩ PukWaC	2 589	762	2 199	1 034

Table 4.12: Prior distributions of the different classes for the different evaluation sets.

Combinations of features Three diverse sets of features were used: context features, similarity features, and features from topic modeling. Furthermore, for each set various set-algebraic combinations were computed. To measure the contribution of each feature set to the final result of the learner, an ablation test was conducted. Table 4.13 shows the results. It becomes clear that the context features contribute most to the total F-score, and notably the subtree feature is most indicative of the existing relation. The subtree features comprises all subtree patterns up to length 6 along the dependency parse that contain both nouns.

	COORD			HYPER			MERO		
	P	R	F	P	R	F	P	R	F
all	0.95	0.83	0.89	0.85	0.40	0.54	0.82	0.70	0.76
excl. sim	0.92	0.80	0.86	0.86	0.34	0.49	0.80	0.70	0.75
excl. ctx	0.88	0.60	0.71	0.46	0.17	0.25	0.74	0.55	0.63
excl. lda	0.96	0.79	0.87	0.85	0.40	0.54	0.82	0.61	0.70
ctx all	0.93	0.76	0.84	0.83	0.34	0.49	0.82	0.56	0.67
excl. X	0.93	0.76	0.84	0.83	0.34	0.49	0.82	0.58	0.68
excl. Y	0.93	0.76	0.84	0.85	0.34	0.49	0.82	0.58	0.68
excl. X ⟨...⟩ Y	0.78	0.14	0.24	0.00	0.00	0.00	0.65	0.16	0.26
excl. X ∨ Y	0.94	0.76	0.84	0.83	0.34	0.49	0.83	0.57	0.68
excl. X ∧ Y	0.93	0.76	0.84	0.86	0.34	0.49	0.82	0.58	0.68
excl. diff X Y	0.93	0.76	0.84	0.83	0.34	0.49	0.83	0.58	0.68
excl. X ¬ Y	0.93	0.76	0.84	0.86	0.34	0.49	0.83	0.58	0.68
excl. Y ¬ X	0.93	0.76	0.84	0.86	0.34	0.49	0.82	0.58	0.68

Table 4.13: Ablation test of the various sets of features used. X and Y are the left and right noun in a pair and ⟨...⟩ represents the subtree feature.

Multiclass Finally, after all the results were in, I computed a multiclass variant of logistic regression and it turned out that the scores improved when training all classes simultaneously. Table 4.14 shows the results attained with the "BLESS ∩ PukWaC" evaluation set.

	COORD			HYPER			MERO		
	P	R	F	P	R	F	P	R	F
binary – ctx feats	.93	.76	.84	.83	.34	.49	.82	.56	.67
multiclass – ctx feats	.89	.84	.86	.83	.39	.53	.77	.71	.74
binary – all feats	.95	.83	.89	.85	.40	.54	.82	.70	.76
multiclass – all feats	.91	.89	**.90**	.78	.44	**.56**	.78	.79	**.79**

Table 4.14: Precision, recall, and F-scores using multiclass logistic regression.

SAT multiple-choice analogy questions In order to evaluate the quality of the generated distributional space in a practical application, I used it to answer the SAT multiple-choice analogy questions, cf. e.g. TURNEY [2011]. For every question, I counted the feature overlap between question and possible answer, and picked the answer with the highest score. Making a prediction for every question, for which at least one of the possible answers was observed in the data, gave 44 percent correct replies, which is significantly better than random choice but worse than the performance of an average U.S. college applicant (57 percent[3]) and than the state of the art (56 percent, TURNEY [2006]).

Hereby, the greatest issue was data sparseness: only 2 out of 192 questions involving nouns were observed in the data with all five possible answers. 4 answers were observed for 7 questions, 3 for 9, 2 for 18, 1 for 18, and for 138 questions none of the answers was found. For two of the questions found with answers, no feature overlap was present, reducing the total number of questions answered to 34. One way to reduce such data sparseness, was used by TURNEY [2006], likewise in the context of answering SAT questions: For each noun pair *A:B* he constructed similar pairs by retrieving alternates for both *A* and *B* from LIN [1998B]'s thesaurus, as *A':B* and *A:B'*, and filtered these alternate pairs by co-occurrence frequency in phrases in a corpus, keeping only the most frequent ones. When answering a question, he then chose the answer pair with the highest average cosine similarity for all combinations of {question pair and alternate pairs} and {answer pair and alternate pairs}. With this approach, which I will consider for future work, Turney could answer 370 out of the 374 total SAT questions (nouns and other).

	CORRECTNESS	0-1 LOSS	SAMPLES PREDICTED
RANDOM	0.21	0.79	185
RESULTS	**0.44**	0.56	**34**
AVERAGE HUMAN	0.57	0.43	185

Table 4.15: Predicting SAT mutliple-choice analogy questions.

4.2.2 Clustering

Clustering was performed using the Chinese Whispers algorithm with a minimum edge weight threshold of 10, constant mutation, continuous update mode, and 30 iterations. All available strategies for updating a node's cluster ID were evaluated, whereby *top*, *dist log*, and *dist nolog* returned quite similar results. I finally settled for *dist log*, which performed just slightly better than the other two. *Dist log* computes a node's new cluster ID by downgrading the edge weight of neighboring nodes with the logarithm of their degree, i.e. the number of edges they have, and picking the cluster ID with the highest computed value.

For evaluation, all clustered noun pairs that also occurred in the BLESS data were selected and various cluster evaluation scores computed. At first, I considered using *B-Cubed* as primary score, but baseline scores showed that it values recall over precision with a single cluster of all data points producing an F-score of 0.7. For use cases of this work, however, the purity of clusters is by far more important than whether these clusters are also complete, i.e. contain all instances of a relation. Therefore, *homogeneity*, a measure of cluster purity, was chosen as

[3] cf. HTTP://WWW.ACLWEB.ORG/ACLWIKI/INDEX.PHP?TITLE=SAT_ANALOGY_QUESTIONS

primary score for evaluating different clusterings.

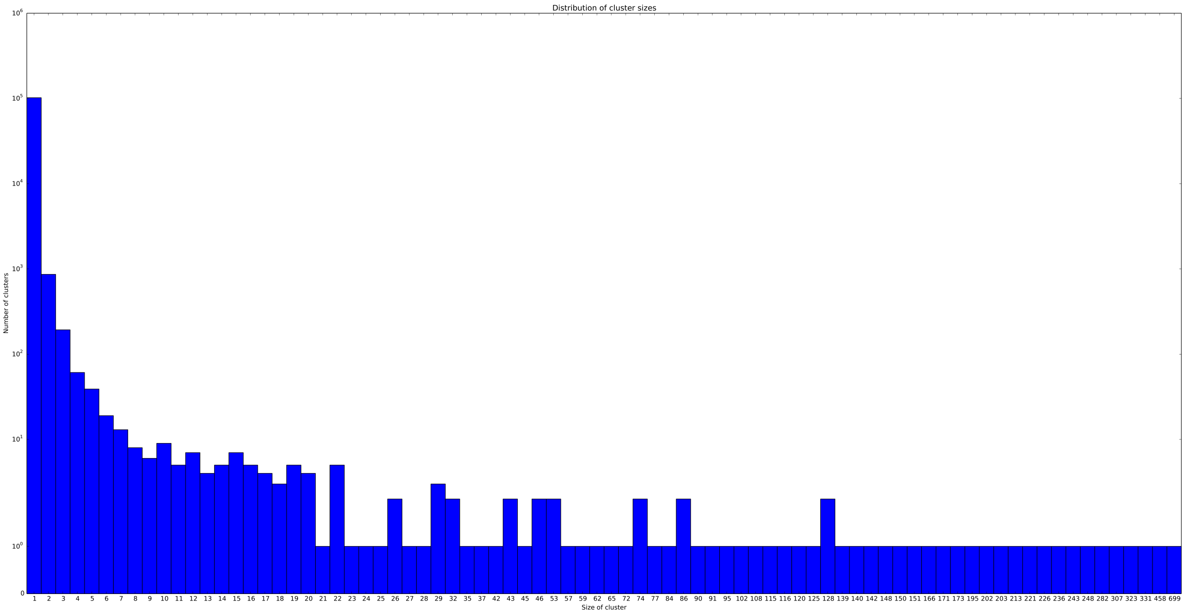


Figure 4.4: Distribution of cluster sizes

Cluster sizes of the best clustering are shown in Figure 4.4. A total of 103 363 clusters were found for 114 615 data points, the bulk of which (102 028) were single-item clusters. Of the remaining 1335 clusters, 178 were of cardinality greater than 5, 74 greater than 20, 32 greater than 100, and 14 contained more than 200 items. Of these 114 615 data points, 1 156 were evaluated – 832 co-hyponyms, 184 hypernyms, 137 meronyms, and 3 random pairs. Unfortunately, the distribution of classes was not very balanced in the evaluation data, even though I tried to keep a balance in the original cluster data by restricting pairs to the 50 most similar ones per pair. While meronyms were spread all over the clusters, there was at least one pure hypernym cluster and several good co-hyponym clusters. Table 4.17 shows various evaluation clusters: one hypernym cluster and three of the co-hyponym clusters; respective evaluation scores are given in Table 4.16. Finally, Table 4.18 shows that same hypernym cluster as it was found in the clustering, that is the complete cluster rather than just the evaluated part of it.

		Homogen.	Precision	Recall	B-Cubed
BASELINE	a single cluster	0.00	53.44	100.00	69.95
	each in own cluster	1.00	100.00	0.26	0.53
	4 random clusters	0.01	53.51	20.11	29.24
RESULTS	sim > 10	0.86	98.93	0.62	1.23
	sim > 20	0.90	99.36	1.02	2.01

Table 4.16: Homogeneity scores for different clusterings.

Cluster 134	Cluster 573
robin::@::sparrow	birch::@::tree
bomber::@::fighter	broccoli::@::vegetable
carrot::@::parsley	carrot::@::vegetable
squirrel::@::rabbit	fox::@::carnivore
deer::@::fox	fox::@::mammal
screwdriver::@::plier	fridge::@::appliance
donkey::@::goat	glove::@::accessory
mackerel::@::tuna	hat::@::apparel
grenade::@::missile	mackerel::@::fish
television::@::fridge	oven::@::appliance
hotel::@::restaurant	rabbit::@::pet
spinach::@::broccoli	salmon::@::fish
knife::@::scissors	tiger::@::creature
tiger::@::leopard	tuna::@::fish
(coord)	(*hyper*)
Cluster 249	Cluster 172
beetle::@::butterfly	carrot::@::bean
birch::@::pine	cow::@::deer
carrot::@::broccoli	fox::@::deer
cauliflower::@::broccoli	fridge::@::television
fox::@::pig	goat::@::donkey
giraffe::@::elephant	herring::@::mackerel
goat::@::dog	knife::@::scalpel
horse::@::dog	lemon::@::orange
sheep::@::dog	missile::@::grenade
toaster::@::refrigerator	pistol::@::grenade
wasp::@::butterfly	rabbit::@::squirrel
willow::@::pine	tiger::@::lion
(coord)	(*coord*)

Table 4.17: Examples of evaluation clusters.

Cluster 573	
salmon::@::fish	heart::@::organ
bread::@::food	mackerel::@::fish
carrot::@::vegetable	rabbit::@::pet
milk::@::product	monoxide::@::vapour
belief::@::concept	vegetable::@::wholefood
hawthorn::@::scrub	adjective::@::modifier
sausage::@::meat	worm::@::nasties
scientist::@::professional	observation::@::source
bacterium::@::organism	fox::@::mammal
investor::@::stakeholder	pea::@::legume
wedding::@::occasion	management::@::issue
otter::@::specie	skin::@::tissue
petrol::@::fuel	vegetable::@::food
religion::@::history	drum::@::instrument
meaning::@::abbreviation	polythene::@::plastic
language::@::subject	glove::@::accessory
oven::@::appliance	fatigue::@::symptom
windsurfing::@::sport	manager::@::stakeholder
furniture::@::object	acupuncture::@::therapy
tiredness::@::symptom	dizziness::@::symptom
problem::@::issue	doctor::@::professional
mouth::@::mucosa	math::@::subject
acupuncture::@::treatment	caffeine::@::stimulant
technician::@::worker	meat::@::dip
supply::@::consumable	daffodil::@::bulb
apple::@::vegetable	simulation::@::computation
gold::@::commodity	video::@::extra
invoice::@::form	pasture::@::habitat
aerospace::@::sector	presentation::@::coursework
diet::@::remedy	flood::@::hazard
asthma::@::disorder	sociology::@::subject
carbohydrate::@::nutrition	ammonia::@::poison
analyst::@::expert	constipation::@::discomfort
anxiety::@::difficulty	anxiety::@::seizure
hat::@::apparel	mite::@::irritant
cyanide::@::poison	racism::@::crime
copper::@::mineral	nurse::@::profession
painkiller::@::medication	cinnamon::@::spice
wool::@::fibre	skill::@::topic
violence::@::behaviour	bag::@::essential
resident::@::objector	cocaine::@::drug
tuna::@::fish	orange::@::citrus
birch::@::tree	zinc::@::metal
insect::@::organism	frustration::@::emotion
crisp::@::snack	hypertension::@::complication
sprout::@::brassica	rice::@::cereal
broccoli::@::vegetable	vanilla::@::flavour
brochure::@::collateral	liberalism::@::ideology
kite::@::raptor	dirt::@::contaminant
catalyst::@::solid	switch::@::modulator
ceramic::@::antiquity	nurse::@::ahps
clown::@::entertainer	oppression::@::evil
food::@::essential	microprocessor::@::ics
corticosteroid::@::immunosuppressants	louse::@::arthropod
steak::@::speciality	fox::@::carnivore
therapist::@::facilitator	professional::@::other
paste::@::bait	duck::@::waterfowl
pepper::@::vegetable	plate::@::tableware
fridge::@::appliance	golf::@::hobby
theatre::@::nightlife	rock::@::debris
helicopter::@::artillery	sand::@::aggregate
zinc::@::micronutrient	inflammation::@::lupus
ivy::@::evergreen	lawyer::@::intermediary
metal::@::recyclables	sponsor::@::guest
pharmaceuticals::@::product	physiotherapist::@::profession
picture::@::message	tiger::@::creature
taxi::@::transport	substance::@::hazard

Table 4.18: A hypernym cluster.

4.3 Error analysis

The previous two sections, Sections 4.1 and 4.2, present and analyze the results obtained from classification and clustering of noun pair representations. The best scores attained were:

	PRECISION	RECALL	F-SCORE
LOGISTIC REGRESSION	0.79	0.78	0.78
CHINESE WHISPERS	0.86		
	HOMOGENEITY		

Table 4.19: Best average scores for classification and clustering

While the results are fairly good, there is room for improvement and in the following I will analyze the errors made by the classifier in order to be able to optimize feature representations in future work.

Confusion matrix Figure 4.5 gives two confusion matrices: one for the evaluation set "BLESS ∩ PukWaC" with four classes and another for the evaluation set "BLESS except random" with three classes. The left matrix shows that random pairs are often mistaken for meronyms and vice versa. Hypernyms are confused with all other classes, and co-hyponyms are not confused much at all, but if they are, it is with meronyms and random pairs more than with hypernyms. As the right confusion matrix depicts, in the second case co-hyponyms and meronyms are sometimes mistaken for one another but rarely are they misclassified as hypernyms. Hypernyms, in turn, are mistaken for all the other classes. Table 4.20 shows some examples of misclassified noun pairs with true and predicted labels – taken from the four-class condition.

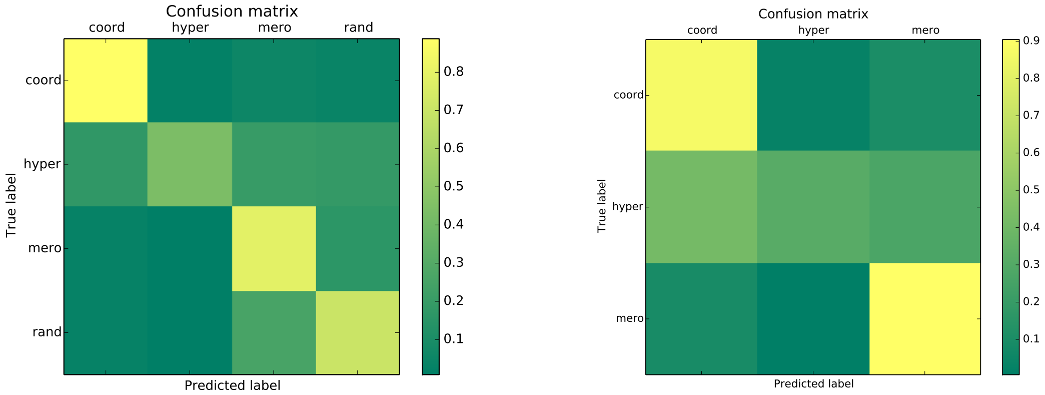


Figure 4.5: Confusion matrices – all four classes (left) and excluding random.

A closer look at the four-class condition reveals that misclassified random pairs were frequently represented by patterns with high coefficients for one of the other classes, cf. Table 4.25. For instance, by patterns like ⟨X or Y⟩, ⟨X and Y⟩, ⟨of X Y⟩, ⟨with X and Y⟩, ⟨X and its Y⟩, and ⟨X , Y⟩ which are indicators of co-hyponymy, hypernymy, and meronymy – two patterns per relation in that order. One way to try to minimize confusions with random pairs would be to include truly random pairs during training rather than only random pairs that already display significant distributional similarity which is why they became part of the DT and therefore evaluation set in the first place.

MISCLASSIFIED PAIRS		
PAIR	LABEL	PREDICTION
villa::@::house	hyper	coord
radio::@::music	mero	hyper
cockroach::@::animal	hyper	mero
scarf::@::garment	hyper	random
guitar::@::plastic	mero	random
jet::@::artefact	hyper	mero
bed::@::climate	random	mero
phone::@::dial	mero	coord
pig::@::counterpart	random	mero
car::@::reality	random	hyper
jar::@::jug	coord	mero
box::@::artefact	hyper	mero
guitar::@::top	mero	random
deer::@::composition	random	mero
alligator::@::eye	mero	random
lizard::@::supplement	random	mero
robe::@::stocking	coord	mero
bottle::@::gbp	random	mero
hotel::@::revenge	random	mero
guitar::@::arrangement	random	mero
missile::@::device	hyper	coord
oven::@::device	hyper	mero
horse::@::squirrel	coord	mero
truck::@::music	random	mero
bed::@::security	random	mero

Table 4.20: Selection of misclassified pairs with true and predicted labels.

Hypernym predictions might be improved by increasing the support for hypernymy which was significantly lower than for co-hyponymy and meronymy with about 150 pairs compared to 600 for the other conditions.

Hypernyms Inspecting subtree patterns with high model coefficients for hypernymy (see Table 4.25) reveals that these comprise Hearst patterns like ⟨X and other Y⟩, ⟨X other Y⟩, and ⟨include X Y and⟩ but not ⟨X such as Y⟩ and ⟨X especially Y⟩. However, a range of other highly predictive patterns are found, among them ⟨is X Y ?⟩, ⟨X is a Y⟩, and ⟨X , which is Y⟩.

Table 4.21 lists sentences that the pattern ⟨X is a Y⟩ occurs in and shows that mostly hypernyms are matched. Similarly, with ⟨X, which is Y⟩, the nouns extracted are mostly in a hypernym relation. Exceptions include sentences like

> *Police are currently trying to trace the man, and his car, which is a black or dark blue left hand drive VW Golf bearing Polish number plates.*

for the pair *car::@::plate*. This particular error can probably be attributed to incorrect parsing. Figure 4.6 displays a correct dependency parse of the sentence, which would not yield the pattern ⟨CAR is a PLATE⟩ but rather a pattern like ⟨CAR golf bearing PLATE⟩.

PAIR	SENTENCE
parsley::@::herb	With a mild and agreeable , tangy sweet and rich flavour , parsley is a popular kitchen herb .
rifle::@::weapon	The M4 rifle is a great weapon but it is better at close-quarter combat .
rat::@::beast	A well-fed adult rat is a fearsome beast , almost the size of a small cat .

Table 4.21: Examples sentences yielding the pattern ⟨X is a Y⟩.

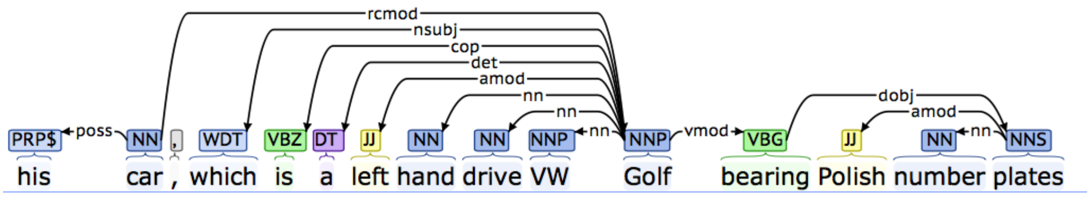

rcmod
nsubj
cop
det
amod
nn
nn
poss
nn
vmod
dobj
amod
nn
PRP$ NN . WDT VBZ DT JJ NN NN NNP NNP VBG JJ NN NNS
his car , which is a left hand drive VW Golf bearing Polish number plates

Figure 4.6: Correct partial dependency parse of *Police are currently [...].*

The pattern ⟨is X Y ?⟩, which ranks 4th and at first intuitive look might seem a good predictor of hypernymy, returns rather mixed results. Table 4.22 shows example sentences that this pattern occurs in. From these examples, two sources of error catch one's eye: 1. that the conjunction *or* does not appear in the extracted pattern, which is due to the way it is annotated in the dependency parse (cf. Figure 4.7). It therefore does not show up in the subtrees which walk the shortest path. And 2. that multiword expressions are either split into several separate words or only used partially, since multiword expressions are not considered in this work. Both problems should be alleviated and examined in future work.

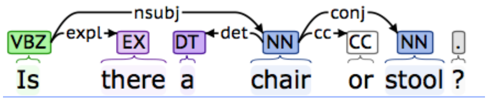


Figure 4.7: Dependency parse of the sentence *Is there a chair or stool ?*.

Co-hyponyms Similar problems occur with co-hypernym patterns. ⟨of X or Y⟩, the 4th most significant pattern for co-hyponymy (cf. Table 4.25), extracts many different co-hypernyms but some of the pairs it captures are hypernyms. These are included due to missed multiword expressions or quantifiers and general adjectives that were omitted from the pattern. Table 4.23 gives examples where either the pair or the pattern should be different – with adaptations suggested.

PAIR	SENTENCE	RELATION
cat::@::animal	Is your cat a party animal?	hyper
robin::@::bird	Is a robin a bird of prey?	hyper
coconut::@::fruit	Is a Coconut a fruit, vegetable or a nut?	hyper
cat::@::pet	But is a cat the right pet for you?	hyper
banana::@::fruit	What is a banana if not a fruit?	hyper
chair::@::stool	Is there a chair **or** stool?	coord
gun::@::bomb	Is it a gun , a tank , **or** a bomb?	coord
pub::@::restaurant	Is there a pub **or** restaurant nearby?	coord
cat::@::panther	Is it a black cat **or** a dangerous panther?	coord?
eagle::@::feather	Where is the eagle's feather you have?	mero
phone::@::number	What is my **phone number**?	random /mwe
elm::@::tree	How far away from you is the **elm tree**?	hyper /mwe
cat::@::tiger	Is a **Bengal cat** a tiger?	coord /hypo /mwe

Table 4.22: Examples sentences yielding the pattern ⟨is X Y ?⟩.

PAIR	SENTENCE	SUGGESTION
cat::@::animal	It is this devotion that makes a pet portrait of your cat , dog , horse or **any animal** a perfect gift for any occasion.	of X or any Y
glove::@::material	This can usually be overcome by the dentist using a low-allergy brand of gloves or **alternative materials**.	of X or alternative Y
knife::@::weapon	I dread to think what the consequences could have been if the person had been in possession of a knife or **other lethal weapon**.	of X or other Y
apple::@juice	Drink only pure water , or if you really cannot do this , allow yourself a small quantity of apple or **grape juice**.	apple juice and grape juice
sheep::@::animal	Finally , there is the sacrifice of a sheep or **other animal** as the climax of the pilgrim 's task.	of X or other Y
pistol::@::weapon	I told them to stop firing any kind of pistol, gun, rifle or **other weapon**, or not get dinner, and went back into the house.	of X or other Y
car::@::vehicle	Leasing options for any make or model of car or **commercial vehicle**.	car and commercial vehicle

Table 4.23: Example pairs erroneously extracted with ⟨of X or Y⟩.

Meronyms Patterns like ⟨X have Y⟩, ⟨X with Y⟩, and ⟨X contains Y⟩ very reliably retrieve foremost meronyms. The pattern ⟨X orange Y⟩, which ranks 2nd, is very rare with 9 occurrences but it returns almost only meronyms – if mostly for the wrong reasons. Table 4.24 shows these pairs and their context and makes clear that most pairs, though meronyms in themselves, are not represented as such in the text. Rather these sentences contain lists of co-hyponyms with the exception of lion::@::mane, which is the only true meronym found. The reason for this confusion seems to be missed multiword expressions and as a result, for instance, plum::@::peel is retrieved instead of plum::@::orangePeel.

PAIR	SENTENCE
apple::@::peel	Box of 75g 4.65 Fruit tea – pure natural tea, rich in apple, orange peel, hibiscus.
lemon::@::juice	Other common drinks include lemon, apple, and orange juice.
apple::@::apricot	Vitamin B17 is found in all fruit seeds such as the apple, peach, cherry, orange, nectarine and apricot.
plum::@::juice	June 4th, Monday Breakfast this morning, stewed plums, orange juice, and scrambled eggs.
lion::@::mane	Do you know a lion's orange bushy mane?
plum::@::peel	Medium bodied and fragrant, showing notes of smoky plum, orange peel and herbs, the refreshing aftertaste is in the style of wines from the northern Rhine.
grapefruit::@::juice	UGLY Pour equal amounts of grapefruit and orange juice over plenty of ice and serve with straws.
apple::@::juice	Drink was on the go too, but there was apple or orange juice for the drivers (promise, that's what is photographed!
apricot::@::juice	Place the apricots and orange juice in a bowl and set aside to marinate overnight.

Table 4.24: Sentences extracted with ⟨X orange Y⟩.

Negative coefficients Notably, subtree patterns with negative coefficients were indicative of the other classes. For instance, the patterns ⟨X and Y⟩ and ⟨X, Y⟩ predict co-hyponymy and have a negative coefficient for hypernymy and meronymy, the patterns ⟨X have Y⟩ and ⟨X on Y⟩ suggest meronymy and reduce the probability for predicting hypernymy and co-hyponymy, and the patterns ⟨X are Y⟩ and ⟨X and other Y⟩ point towards hypernymy and downgrade co-hyponymy and meronymy as prediction candidates, see Table 4.25.

In summary, it appears that multiword expressions are a crucial element when retrieving semantic relations from sentences, and they should be taken into consideration in future refinements. Additionally, it would be interesting to include *conj* dependencies together with their string representation in patterns, even if they are one hop away from the shortest path, and see how that improves retrieval.

COORD		HYPER		MERO	
+					
4.6	X and Y^{-1}	3.0	X and other Y	2.8	X have Y
4.2	X , Y^{-1}	2.9	X other Y	2.7	X orange Y
1.9	X Y and^{-1}	2.4	X become Y	2.3	X for Y and
1.5	of X or Y	2.3	is X Y ?	1.9	X with Y
1.4	X Y :	2.2	X used in Y	1.9	her X Y
1.4	X or Y^{-1}	2.2	X , etc. Y	1.7	put X Y
1.3	the X and the Y	2.1	X , and Y , and	1.7	X contains Y
1.3	of X Y made	1.9	X were Y	1.6	a X with a Y
1.2	X Y built	1.9	X are Y	1.5	their X , , Y
1.2	a X and a Y	1.8	X is a Y	1.3	that X Y
1.2	a X Y^{-1}	1.6	X , Y may	1.2	is a X Y
1.1	X , Y , ,	1.5	fitted X and Y	1.2	X has Y
1.1	a X or Y	1.5	X and soft Y	1.2	X use Y
1.1	X , Y and	1.5	, X Y would	1.2	X of Y
1.0	X , , , Y	1.4	some of the X Y	1.1	X had Y ,
1.0	about X Y	1.4	X - Y	1.1	the X from Y
1.0	X and Y	1.4	, the X Y	1.1	wears X with Y
1.0	includes X and Y	1.3	X another Y	1.1	X their Y
1.0	X and two Y	1.2	X waterproof Y	1.0	X including Y
0.9	X Y had	1.2	X over Y	1.0	two X Y ,
0.9	wearing a X Y	1.2	complete with X Y	1.0	than X , and Y
0.9	the X Y^{-1}	1.2	of his X Y	0.9	the X 's Y
0.9	for a X Y	1.1	X for Y	0.9	a X Y
0.8	of X Y^{-1}	1.1	of X and Y	0.9	for X Y and
0.8	especially X and Y	1.1	X and are Y	0.8	X or , Y
−					
3.4	X and other Y	2.1	X and Y^{-1}	2.8	X and Y^{-1}
2.5	X with Y	2.1	X , Y^{-1}	2.6	X , Y^{-1}
2.2	X are Y	1.3	X of Y	1.9	X is a Y
2.0	X is Y	1.3	X with Y	1.7	X Y and^{-1}
1.8	X 's Y	1.3	X has Y	1.6	X other Y
1.6	X has Y	1.1	X have Y	1.5	X and other Y
1.5	X , and Y and	1.1	X Y can	1.4	X for Y
1.3	X , a Y ,	1.0	X on Y	1.2	X was a Y
1.3	X orange Y	0.9	X is Y from	1.1	X Y in
1.3	X on Y	0.9	X , Y ,	1.1	X are Y of

Table 4.25: Subtree patterns with highest coefficients in the model.

5 Conclusions

"Come on, say something conclusive!"
(Homer Simpson in "Sleeping with the Enemy")[1]

To sum up, I will briefly revisit the hypothesis of this work, discuss to what degree it can be dismissed or confirmed, and share the main insights derived in the process.

The hypothesis of this work was to test whether syntagmatic representations of paradigmatically related pairs of nouns are sufficient and adequate to predict the semantic relation that holds between them. Where *paradigmatically related* also means that these nouns are distributionally similar, and *syntagmatic representations* implies that they are represented by patterns extracted from sentences containing both of these nouns.

As the results evince, the semantic relations learned could be predicted fairly well, albeit to varying degrees. Co-hyponymy was predicted most reliably with an F-score of 0.90. Thereby, patterns like $\langle$X and Y$\rangle^{-1}$ and $\langle$X , Y$\rangle^{-1}$ sported the highest coefficients. Meronymy scored with 0.79, and most difficult was hypernymy with 0.56. Although various Hearst patterns ranked highly in the patterns predictable of hypernymy, the lower support and confusion with multiword expressions contributed to the lower score.

Analysis of the contributions of different features to prediction results highlighted the subtree feature, which comprised all subtrees of a certain length from the dependency parse, as the single most predictive feature. Applying set operations to pair representations, on the other hand, did not contribute much to successful prediction. However, it might be worth combining pair and noun representations in future work, and applying *SimDiff* only to the latter, as this has already been shown to work well, see Turney and Mohammad [2013].

The bless evaluation set, though carefully devised, showed to introduce bias unless the training and test data were de-lexicalized before stratification, ensuring that the classifier did not merely learn single words as good representatives of a relation. This is due to the distribution of left and right nouns in the relations in bless, see Figure 4.3 in Section 4.2. Weeds et al. [2014], who were very careful about the properties of the evaluation sets they use, noticed similar effects with the bless data and addressed them in equal manner.

In task-oriented evaluation with the sat multiple-choice analogy questions, the data sparseness added by representing pairs rather than single nouns became very noticeable: only 2 out of 192 questions involving nouns could be observed in the data including all answer choices. This pertained despite mitigation attempts by searching both for questions as incoming and their mirrored form with each pair inverted. Strategies analogous to Turney [2006] might be more promising and could be applied in future evaluation.

[1] h/t Sahlgren [2008]

While first experiments with clustering gave good results, this should be explored further and in more detail in future work.

All things considered, the examined approach produced good results which are worthwhile building upon and refining in future work.

6 Affirmation

Hereby I confirm that I wrote this thesis independently and that I have not made use of any other resources or means than those indicated.

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, September 23, 2014

List of Figures

List of Tables

Bibliography

Unless noted otherwise, websites referenced below were last accessed on August 20, 2013. In case of unavailability at a later time, we recommend visiting the Internet Archive.

Charu C Aggarwal and Chandan K Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, 2013.

Alan Agresti. *Categorical data analysis*. John Wiley & Sons, 1990.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

Marco Baroni. Dr. strangestats or: How i learned to stop worrying and love distributional semantics, 2013. URL HTTP://WWW.DAGSTUHL.DE/MAT/FILES/13/13462/13462.BARONIMARCO.OTHER.PDF.

Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

Marco Baroni and Alessandro Lenci. How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics, 2011.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226, 2009.

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology*, 9, 2014.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

Chris Biemann. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the first workshop on graph based methods for natural language processing*, pages 73–80. Association for Computational Linguistics, 2006.

Chris Biemann and Martin Riedl. Text: Now in 2d! a framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95, 2013.

Danushka T Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring the similarity between implicit semantic relations from the web. In *Proceedings of the 18th international conference on World wide web*, pages 651–660. ACM, 2009.

Danushka Tarupathi Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *Proceedings of the 19th international conference on World wide web*, pages 151–160. ACM, 2010.

Stefan Bordag. A comparison of co-occurrence and similarity measures as simulations of context. In *Computational Linguistics and Intelligent Text Processing*, pages 52–63. Springer, 2008.

Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of the Biennial GSCL Conference*, pages 31–40, 2009.

Elia Bruni, Jasper Uijlings, Marco Baroni, and Nicu Sebe. Distributional semantics with eyes: Using image analysis to improve computational representations of word meaning. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1219–1228. ACM, 2012.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources*, volume 2. North American Chapter of the Association for Computational Linguistics, 2001.

Peter A Chew, Brett W Bader, Tamara G Kolda, and Ahmed Abdelali. Cross-language information retrieval using parafac2. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 143–152. ACM, 2007.

Noam Chomsky. Aspects of the theory of syntax cambridge. *Multilingual Matters: MIT Press*, 1965.

Kenneth Church, William Gale, Patrick Hanks, and Donald Kindle. Using statistics in lexical analysis. *Lexical acquisition: exploiting on-line resources to build a lexicon*, page 115, 1991.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

Stephen Clark. Vector space models of lexical meaning. *Handbook of Contemporary Semantics–second edition. Wiley-Blackwell*, 2012.

Daoud Clarke. Context-theoretic semantics for natural language: an overview. In *Proceedings of the workshop on geometrical models of natural language semantics*, pages 112–119. Association for Computational Linguistics, 2009.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, volume 6, pages 449–454, 2006.

Ferdinand de Saussure. 1959. course in general linguistics. *New York: Philosophical Library*, 1916.

Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

Katrin Erk, Sebastian Padó, and Ulrike Padó. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763, 2010.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231. AAAI Press, 1996.

Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.

Oren Etzioni, Michele Banko, and Michael J Cafarella. Machine reading. In *Proceedings of the 21st national conference on Artificial Intelligence*, volume 6, pages 1517–1519. AAAI Press, 2006.

Stefan Evert. *The statistics of word cooccurrences.* PhD thesis, Dissertation, Stuttgart University, 2005.

Peter Gärdenfors. *Conceptual spaces: The geometry of thought.* MIT Press, 2004.

Dedre Gentner and Arthur B Markman. Structure mapping in analogy and similarity. *American Psychologist*, 52(1):45, 1997.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. Semeval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 13–18. Association for Computational Linguistics, 2007.

Ardeshir Goshtasby and J Le Moign. *Image registration.* Springer, 2012.

Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211, 2007.

Ulrike Hahn, Nick Chater, and Lucy B Richardson. Similarity as transformation. *Cognition*, 87 (1):1–32, 2003.

Jiawei Han and Micheline Kamber. *Data Mining, Southeast Asia Edition: Concepts and Techniques.* Morgan kaufmann, 2006.

Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1): 335–346, 1990.

Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

Irene Heim and Angelika Kratzer. *Semantics in generative grammar*, volume 13. Blackwell Oxford, 1998.

Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8): 651–666, 2010.

Edwin T Jaynes. *Probability theory: the logic of science.* Cambridge University Press, 2003.

David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364. Association for Computational Linguistics, 2012.

Jochen Kerdels and Gabriele Peters. Supporting gng-based clustering with local input space histograms. In *Proceedings of the 22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2014)*, pages 559–564, 2014.

Christopher Soo Guan Khoo and Jin-Cheon Na. Semantic relations in information science. *Annual Review of Information Science and Technology*, 40:157–228, 2006.

Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(04):359–389, 2010.

Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211, 1997.

Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 401. AAAI, 2006.

Alessandro Lenci. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31, 2008.

Alessandro Lenci and Giulia Benotto. Identifying hypernyms in distributional semantic spaces. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pages 75–79. Association for Computational Linguistics, 2012.

Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2014.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, volume 98, pages 296–304, 1998a.

Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational linguistics*, volume 2, pages 768–774. Association for Computational Linguistics, 1998b.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996.

David JC MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of SemEval 2014: International Workshop on Semantic Evaluation*, 2014.

Olena Medelyan, Ian H Witten, Anna Divoli, and Jeen Broekstra. Automatic construction of lexicons, taxonomies, ontologies, and other knowledge structures. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(4):257–279, 2013.

Douglas L Medin, Robert L Goldstone, and Dedre Gentner. Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, 1(1): 64–69, 1990.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

Dan Moldovan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju. Models for the semantic classification of noun phrases. In *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics*, pages 60–67. Association for Computational Linguistics, 2004.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

M Lynne Murphy. *Semantic relations and the lexicon*. Cambridge University Press, 2003.

Joakim Nivre, Johan Hall, and Jens Nilsson. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the International Conference on Language Resources and Evaluation*, volume 6, pages 2216–2219, 2006.

Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.

Alexander Panchenko and Olga Morozova. A study of hybrid similarity measures for semantic relation extraction. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 10–18. Association for Computational Linguistics, 2012.

Alexander Panchenko, Olga Morozova, and Hubert Naets. A semantic similarity measure based on lexico-syntactic patterns. In *Proceedings of KONVENS*, volume 2012, pages 174–178. ÖGAI, 2012.

Tan Pang-Ning, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Addison-Wesley, 2006.

Patrick Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006.

Athanasios Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, 1965.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

Reinhard Rapp. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322, 2003.

Reinhard Rapp. A freely available automatically generated thesaurus of related words. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, 2004.

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, IJCAI'95, 1995.

Martin Riedl and Chris Biemann. Scaling to large3 data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 884–890. ACL, 2013.

Alan Ritter, Stephen Soderland, and Oren Etzioni. What is this, anyway: Automatic hypernym discovery. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, pages 88–93, 2009.

Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54, 2008.

Magnus Sahlgren. Distributional semantics. University Lecture, 2012. URL HTTP://WWW.GAVAGAI.SE/DISTRIBUTIONAL_SEMANTICS.PHP.

Magnus Sahlgren and Jussi Karlgren. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering*, 11(03):327–341, 2005.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, page 38, 2014.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, volume 12, pages 44–49, 1994.

Barbara C. Scholz, Francis Jeffry Pelletier, and Geoffrey K. Pullum. Philosophy of linguistics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Summer 2014 edition, 2014.

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

CE Shannon. A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423, 1948.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems (NIPS 2004)*, volume 17, pages 1297–1304, 2004.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics, 2006.

Jeff Speaks. Theories of meaning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2014 edition, 2014.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

Peter D Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006.

Peter D Turney. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 905–912, 2008.

Peter D Turney. Analogy perception applied to seven tests of word comprehension. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(3):343–362, 2011.

Peter D Turney and Saif M Mohammad. Experiments with three approaches to recognizing lexical entailment. *Natural Language Engineering*, pages 1–40, 2013.

Peter D Turney, Patrick Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010.

Amos Tversky. Features of similarity. *Psychological Review*, 84(4), 1977.

Tim Van de Cruys. A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering*, 16(04):417–437, 2010.

Julie Weeds, David Weir, and Diana McCarthy. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1015. Association for Computational Linguistics, 2004.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics*, 2014.

Ludwig Wittgenstein. *Philosophische Untersuchungen / Philosophical Investigations.* Oxford: Blackwell, 1953.

Ludwig Wittgenstein. *The Blue and Brown Books.* Basil Blackwell, Oxford, 1958.