Masterarbeit zur Erlangung des Mastertitel der
Technischen Fakultät der
Albert-Ludwigs-Universität Freiburg im Breisgau

Mitigating Feature Exclusion to Improve Hypernymy Recognition

Max Lotstein

17.07.2015

Albert-Ludwigs-Universität Freiburg im Breisgau
Technische Fakultät
Institut für Informatik

Dekan
Prof. Dr. Hannah Bast

Referenten
Prof. Dr. Joschka Bödecker
Datum der Promotion (only necessary for final publication)
07.01.2015

"I, for one, welcome our new computer overlords."
-Ken Jennings, February 16, 2011

"Oops."
-Rick Perry, November 9, 2011

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Elmar Haußmann, Michael Roth, Luis Lastras and all friends whom I have subjected to conversations about my thesis.

Abstract

Hypernymy, which is the sort of relationship exemplified in the pair $\langle dog, animal \rangle$ when both take their conventional senses, lies at the heart of many applications in artificial intelligence and natural language processing. In this work, I investigate automatic methods of identifying such sense-pairs. Many existing models for this task implement *feature inclusion*, which is the intuition that senses like *dog* are semantically narrower than senses like *animal* in that the properties one can apply to the former can always conceivably be applied to the latter, but not the inverse. State-of-the-art performance of these models has been limited by the *Feature Exclusion Problem*, which is that in the sparse feature vectors representing a sense like *dog*, in which a non-zero feature is interpreted as an applicable one, there are many non-zero features that should also be non-zero in the representation of *animal* but aren't. This work begins by exploring in greater detail the nature of this problem before exploring a post-processing technique designed to mitigate it.

Zusammenfassung

Grundlage für viele Applikationen aus den Bereichen Künstliche Intelligenz sowie natürliche Sprachverarbeitung ist die „Hyperonymi", eine semantische Relation, die beispielsweise das Verhältnis zwischen dem Wortpaar $\langle Hund, Tier \rangle$ beschreibt.

In der vorliegenden Arbeit werden Methoden untersucht, mit denen eine automatische Erkennung derartiger Wortpaare erfolgen kann. Eine Reihe derzeit existierender Modelle, die diesen Methoden zugrunde liegen, berücksichtigen dabei die *feature inclusion*, die davon ausgeht, dass Wörter wie „Hund" semantisch betrachtet enger sind als Wörter wie „Tier". Die Semantische Enge besagt, dass Eigenschaften, die auf das Wort „Hund" zutreffen, immer auch auf das Wort „Tier" anwendbar sind, wohingegen die umgekehrte Schlussfolgerung nicht zulässig ist.

Die Leistungsfähigkeit derartiger dem aktuellen Stand der Wissenschaft entsprechenden Modelle wird oftmals durch das Feature Exclusion Problem limitiert, welches darin besteht, dass viele Eigenschaften, die dem Wort „Hund" zugeordnet werden können, eben nicht auch automatisch Bestandteil der Eigenschaften des Wortes „Tier" sind, sondern ausschließlich auf das Wort „Hund" zutreffen. Im Rahmen dieser Arbeit wird zunächst das eben geschilderte Problem detaillierter betrachtet werden. Anschließend wird eine postprozedurale Vorgehensweise untersucht, die eine Lösung des Problems ermöglichen soll.

1 Introduction

1.1 Introduction

Smart technologies that promise to make interacting with technology via language as fluid as interacting with a human being are becoming ubiquitous but despite their spread, the technology is far from mature. In fact, the failure of software with limited language-interpretation ability has become a cultural touchstone, with web communities[1] where people share their favorite examples of software acting deranged as it tries to interpret a human.

While some of these failures are no doubt the result of software bugs, many of them are the result of the sheer difficulty of natural language understanding, the formal name for the problem of enabling a computers to understand human communication.

Contributing to the solution of the natural language understanding problem is the goal of many different fields. One such field is Distributional Semantics (DS), one goal of which is create models of language meaning (Distributional Semantic Models or DSMs) that are trained automatically when provided with many examples of human communication. The appeal of such models is that, if they work, they are trivial to scale.

One task toward which DSMs have been applied is the identification of pairs of senses[2] $\langle A, B \rangle$ such that A is a kind of B, a task I refer to as Hypernymy Recognition (HR). HR is important for many downstream applications, such as Question Answering (QA), Information Retrieval (IR), Information Extraction (IE), Machine Translation (MT), Strong AI and others. If a QA system were asked a question like, 'Does Barack Obama have a pet?', it would have to infer the answer from related information, e.g. 'Barack Obama has a dog named Bo'. IR systems face a similar challenge in that queries for a category can be satisfied with examples of that category but not vice-versa. In IE, the organization of extracted information into tables requires recognizing when an entity is an instance of a category. In MT, given a target word to translate, there may not be an equivalent foreign language word but there may be an equivalent to a word entailed by the target word. And a Strong AI system would be expected to understand that if A can sleep, A could theoretically also snore.

[1] such as http://www.damnyouautocorrect.com/

[2] *Sense* here refers to the meaning of word and it is necessary to use in light of the fact that polysemous words may have many senses only one of which may be appropriate.

Hypernymy is an asymmetric lexical relation in which senses acting as A does in 'A is a kind of B' are referred to as *subordinates* or *hyponyms* and sense acting as B does are referred to in this field as *hypernyms*.

The most common way of modeling hypernymy in DSMs is based on *feature inclusion*, according to which the prototypical set of permissible features of the hypernym *includes* the set of permissible features of the hyponym. For example, *fish* can have the property of *swim* and *animal*, a hypernym of *fish*, can as well. However, *fish* can not have the property *gallop*, though *animal* can. Thus, the set of features that *animal* can have *includes* the set of features of *fish* (if we pretend that *fish* can have only the one feature).

Expressing feature inclusion is trivial in DSMs when each word is represented by a sparse feature[3] vector[4], which is a common practice: with sparse feature vectors, feature inclusion is equivalent to a subset. Many researchers have created models of hypernymy using DSMs and feature inclusion [Lenci and Benotto, 2012, Roller et al., 2014, Weeds et al., 2014a]. Recently, however, problems limiting the effectiveness of these models have become apparent.

The problem is that, rather than being a subset, the features of a hyponym's vector are very often not members of the set of features of hypernym's vector. I refer to features that are found in the hyponym's vector but not the broader term's as *excluded features* and this problem generally as the Feature Exclusion Problem (FEP).

To address the FEP, two procedures for enriching the representations of DSMs are proposed and their effects analyzed. The hypothesis is that these procedures will boost precision in hypernymous pairs more than non-hypernymous ones, and thus improve the performance of feature inclusion models on the HR task. The results suggest that, while the procedures fail to improve performance on HR, something needs to be done to address the FEP.

1.2 Structure

The remainder of the work is structured as follows: Chapter 2 reviews background information on DS and DSMs; Chapter 3 reviews the semantics of lexical relations and the design of WordNet, a popular lexical database, which is used extensively in this work; Chapter 4 reviews contemporary approaches to the task of HR; Chapter 5 describes the nature of the FEP in greater detail; Chapter 6 describes the proposal for addressing the FEP; Chapter 7 consists of exploratory analyses of the effects of the proposed solution; Chapter 8 examines the extent to which the proposed solution improves HR performance; Chapter 9 concludes.

[3] *Feature* here refers to the dimensions of the space in which a given vector lives; it is also synonymous with *column*.

[4] A sparse feature vector is one in which most of the elements are 0.

For readers already familiar with DSMs, lexical relations and WordNet, Chapter 4 may be a reasonable starting point.

2 DS and DSMs

In this chapter, I will review the definition of DS and explain how DSMs implement the theory in order to construct sparse feature vectors that serve as representations of meaning. A deep understanding of how non-zero features in a feature vector acquire their weight is important to appreciate the analysis of the Feature Exclusion Problem (FEP) in later chapters. However, before starting on the details of DS and DSMs, I will first describe the central idea behind representing meaning using space.

It should be noted that not all DSMs *do* represent meaning as sparse feature vectors constructed from frequency statistics. A prominent and recent alternative to the sparse feature-space model is the so-called neural embedding model, exemplified by the work of Mikolov [Mikolov et al., 2013, Le and Mikolov, 2014], a more accessible review of which can be found in [Levy and Goldberg, 2014]. The decision to only review sparse feature spaces reflects the consensus among current work in HR; most approaches to HR use approaches similar to what is described here. This consensus is possibly because feature subsets, which are crucial to contemporary models of HR, are more easily expressed when features can be present or absent, as in a sparse feature space, which criteria does not hold in a typical dense vector in a neural embedding.

2.1 The Geometric Metaphor for Meaning

As already stated, many models for HR are implemented as sparse feature spaces. Independent of this task, however, sparse feature-space models are among the most popular sort of DSM. The reasons for their popularity are not universally agreed upon. One could argue, as Erk [2012] does, that these models have computational advantages, in that we already have mathematical libraries that accept and manipulate vectors. However, I think the appeal of these models reflects the more general appeal of space as a figurative framework.

The ways in which space serves as a source domain for metaphor are myriad. Many adjectives with spatial connotation can be used in a figurative way. Despair is deep, for example. We talk about emotional well-being as a space ("I'm in a good place right now"), or processes as journeys through space ("On the road to recovery"). Lakoff and Johnson identify many such metaphors [Lakoff and Johnson, 1997, 1999] and among the simplest of these is similarity-is-proximity. This metaphor is so well-rooted that it is hard to think about similarity and not think about proximity

[Lakoff and Johnson, 1999]. In the context of DSMs, this idea is referred to as the *geometric metaphor of meaning* [Sahlgren, 2006] and the central idea for the representation of words (when we assume each word represents a single sense) in a DSM is to let words be entities in some space, whose locations we can encode using vectors, the components of which are a function of observable phenomena, and to treat the distance between points in this space as being inversely proportional to their semantic similarity.

However, it is important to note that the similarity-is-proximity metaphor is imperfect in that many aspects of the source domain, space, cannot actually be applied to meaning. We cannot say that one meaning is in front of another, for instance. Moreover, there are problems interpreting psychological accounts of semantic similarity as distance. Distance is metric, and therefore symmetric and subject to the triangle inequality, whereas empirical measurements and thought experiments suggest semantic similarity is neither. Tversky [1977] presented this example: reversing the order of the arguments can change a statement's meaning, e.g., "A man is like a tree" implies that man has roots; "A tree is like a man" implies that the tree has a life history. This suggests that similarity is order-dependent. In contrast, the statement "The man is 5 feet from the tree" is the same when the order of its arguments is reversed. Tversky also presented this example:

1. Jamaica is similar to Cuba (because of geographical proximity), and
2. Cuba is similar to Russia (because of political affinity), but
3. Jamaica and Russia are not similar at all*

which shows that the triangle inequality[1] and transitivity do not constrain semantic similarity. Nevertheless, if Lakoff and Johnson are right, space may be the only way to think about meaning that is psychologically palatable.

2.2 The Finished Product

It is helpful to consider what the sparse feature-space in a typical DSM looks like and to establish the terminology I will use. Sparse feature-spaces are also sometimes referred to as *coocurrence matrices* because their non-zero values capture statistically noteworthy *cooccurrences*. The magnitude of the values in each vector is a function of the frequency with which a *row element* (thing being modeled) and *basis element* (feature, very often a word or a word annotated with some grammatical or syntactic information) cooccur in the corpus used as input. Fig. 2.1 shows a toy example of such a cooccurrence matrix. The parameters that determine the makeup of the row and basis elements, and the means by which the values of each component are calculated, will be reviewed in the coming sections.

[1] The triangle inequality imposes the constraint that if A is similar to B and B is similar to C, then A cannot be dissimilar to C.

$$\begin{array}{c} \\ w_1 \\ w_2 \\ ... \end{array} \begin{array}{c} \begin{array}{ccc} b_1 & b_2 & ... \end{array} \\ \begin{bmatrix} & a_{b_2}^{w_1} & \\ a_{b_1}^{w_2} & & \\ & & \end{bmatrix} \end{array}$$

Figure 2.1: A sample cooccurrence matrix, in which 0 values are omitted. The row elements $w_1, w_2...$ are from the set W, the things being modeled. The basis elements $b_1, b_2, ...$ are from the set B, the set of features in the space. The values $a \in A$ are the result of applying an Association Measure to cooccurrence frequency statistics.

2.3 Defining Distributional Semantics

DS is a theory of meaning. A fundamental part of any theory of meaning is the set of conditions under which two senses can be said to be semantically similar. As Lenci [2008] writes, the hallmark of DS is to meet this requirement through the Distributional Hypothesis (DH) [Harris, 1954]. The DH can be summarized by a slogan popularized by one of its original proponents: "You shall know a word by the company it keeps!" [Firth, 1957]. If we interpret the word *company* to be *lexical company*, then as Lowe [2001] writes, we can know the semantic character of a word by examining its associated words. Thus, the DH provides a means by which large collections of human communication can serve as the sole input to a model of meaning; one simply needs to examine associated words in the corpus to arrive at a word's meaning. More formally and broadly, the DH states that "the degree of semantic similarity between two words (or other linguistic terms) can be modeled as a function of the degree of overlap among their linguistic contexts" [Baroni and Lenci, 2010]. Thus, we see that the DH is a definition of meaning, not just for words, but for any linguistic item the modeler wants to treat as atomic or non-compositional; the DH also doesn't prescribe the sort of company one considers, and any linguistic context can serve as company. Though the DH can be interpreted as a cognitive hypothesis about the origin of semantic representations, this work adopts an alternate interpretation of the DH as a quantitative method for semantic analysis and lexical resource induction [Lenci, 2008].

2.4 DSMs and their Parameters

In order to implement the DH, DSMs must define *overlap* and *linguistic context* (as in the Baroni and Lenci definition above) and they do so by defining a number of parameters. Lowe [2001] identified a few of the most important parameters. I adapt the Lowe formulation and formalize DSMs as a six tuple $\langle R, B, C, A, S, M \rangle$, where R is the set of Row Elements, B is the set of Basis Elements, C is the definition of cooccurrence, A is the Association Measure, S is the Similarity Measure and M

is the Transformation or Smoothing, following [Turney and Pantel, 2010]. I will address each of these in the coming sections.

2.4.1 Row Elements

Choosing the right set of row elements (the set linguistic items being modeled), can be surprisingly challenging, even when the objective (lexical semantics) is clear. It would seem obvious to choose the set of words as the set of items whose company we must observe. In practice though, the boundaries of this set are not at all clear, even when ignoring the fact that many words have more than one sense, and that our efforts from the start will be compromised by having a single representation for more than one sense.

One might say a word is a sequence of letters bordered by whitespace. Or, as Evert [2005] points out, is it *white-space*? Or *white space*? All of these can be found in common usage. The only way to discriminate words with spaces from groups of words that are often used together is to take into account word meaning. And thus it seems there is a chicken-and-egg problem: in order to achieve a lexical semantics, we need to have one already. For the remainder of this work, I will use 'words' to refer to uninterrupted sequences of characters that may include hyphens, and refer to cases like 'white space' as multi-word expressions. This definition of *word* only applies to English and other languages with fairly regular or simple morphological systems. The challenge of preparing words for processing is the responsibility of the tokenizer and this issue will be revisited in Sec. 2.4.7. I use the word *sense* to refer to the meaning a word can have.

2.4.2 Basis Elements

The set of basis elements in the model, which is another name for the set of features in the space, is one of the most interesting and important parameters in the DSM. The set of possible basis element sets can be conceived as a spectrum of intentionality. At one end are sets of basis elements consisting entirely of intentionally selected or constructed basis elements. At the other end of the spectrum are sets of basis elements that have not been selected intentionally, but which arise as a consequence of a model of cooccurrence being applied to corpus. The historical trend has been to move from extremely intentional basis element sets toward far less intentional ones. Recently, however, researchers have proposed models that represent a compromise, with a mixture of basis elements reflecting *a priori* beliefs about what is informative as well as basis elements that arise from cooccurrence. The procedures described here, which add new basis elements to an existing feature space, can be seen as part of this trend.

One of the first attempts to model meaning with space was in the 1950s, by Osgood [1952]. Osgood and his colleagues devised a set of 50 features along which they

imagined senses might vary, each of which was defined by a scale with two extrema (e.g., small-large, weak-strong). They then solicited ratings on a seven-point scale from human subjects for a substantial set of words, and composed the ratings for each word into a feature vector, such that each word occupied a point in 50-dimensional feature-space. Other researchers since have attempted to hand-construct the set of features along which it was theorized words or concepts vary [Waltz and Pollack, 1985]. However, a major problem with hand-selecting features is that it obliges one to determine not just the number of features, but what the features are, and the cost of failing is a semantic space in which the measured distance and the true semantic distance are discrepant. And more problematic, Osgood's methodology is impractical to scale for larger sets of words.

Alternately, the set of basis elements can emerge as a cooccurrence model is applied to a corpus. In the 1960s, the information retrieval community first began to use documents as features to construct term-document matrices [Salton and McGill, 1983] but it was Schütze's 1992 word-word cooccurrence matrices that bear the greatest degree of similarity to contemporary DSMs. Rather than using documents or features selected by hand, Schütze let the words themselves serve as the features of the space. Schütze specified the width of a window in terms of number of characters and by sliding this window (automatically) through a corpus and treating all words within it at one time as co-occurring, was able to compute the frequency with which words occurred in the window at the same in the corpus.

More recently, researchers have begun to filter and annotate the words used as basis elements. For example, Weeds et al. [2014a] used only open-class words (nouns, verbs, adjectives and adverbs) as basis elements. Baroni and Lenci [2010] annotated words with both part-of-speech and grammatical dependency[2] information, e.g., *ball-n_NSBJ*[3], which translates to the noun *ball* governed by an *NSBJ* dependency relation. Baroni and Lenci's TypeDM model is also notable for its usage of abstract basis elements based on linguistic research. For example, the sequence of words *such as* has been shown by Hearst [1992] to indicate a hypernymy relation, e.g., *animals such as cats*, and one basis element in TypeDM model corresponds to $\langle word, suchas, word \rangle$. Attribute nouns (color, size, etc.) are likely to indicate a property [Veale and Hao, 2008], e.g. *the color of strawberries is red*, where *color* is an attribute noun, and this too can be exploited in the choice of basis elements. The present work represents another example of incorporating linguistic knowledge, in which this knowledge is used to modify the feature space by adding basis elements whose value is a function of other basis elements, rather than grouping different sorts of cooccurrences as is done in TypeDM.

[2]Grammatical dependencies are asymmetric relationships between pairs of words in a sentence. In a dependency relation, one word is said to be *governed by* its governor and one implicit governor, *ROOT*, governs the entire sentence.

[3]*NSBJ* is an example of a dependency from the Stanford typed dependencies [De Marneffe and Manning, 2008]. It represents the relationship between a nominal subject and the clause that governs it.

2.4.3 Types of Cooccurrence

Given a choice of B, the set of basis elements, and R, the set of row elements, the next step is to define C, what it means for a row and basis element to co-occur in the same context.

2.4.3.1 Surface Coocurrence

One option is to define cooccurrence in terms of what Evert [2008] calls *surface distance*. Surface distance considers distance in terms of either words (consecutive sequences of characters, in contrast to multi-word expressions) or characters. Under this definition, there are two parameters that can be varied: the size of the span and its direction. The size of the span, when it refers to words, typically takes a value between 2 and 5 words [Lapesa and Evert, 2014], although in some cases, it can be in the hundreds [Evert, 2008]. With regard to direction, the span can extend in both directions or it can be asymmetric [Sahlgren, 2006]. Importantly, no additional processing of the text is needed to recognize surface cooccurrences.

2.4.3.2 Textual Cooccurrence

Textual cooccurrences are partially a response to criticism that using surface cooccurrences requires one to choose a span size, the optimal value of which might be language or even task dependent. Consider these three cases with *give* and *speech*:

1. to give a speech
2. to give an excellent speech
3. a speech was not and will not be given at this time

No value of span size but one sufficient to include (3) would capture *give* and *speech* in these examples, and yet setting the span so large might introduce far more noise than signal. Instead of a threshold defined in terms of proximity, textual cooccurrences use textual units as thresholds, for instance sentential boundaries or documents themselves. Salton and McGill [1983] used textual cooccurrences to construct their term-document matrices. Textual cooccurrences then capture weaker relationships.

2.4.3.3 Syntactic Cooccurrence

Finally, *syntactic cooccurrences*, like surface cooccurrences, also use a notion of proximity, but instead of measuring with respect to a number of graphemic words they measure with respect to some linguistic interpretation, such as clauses or a syntactic representation. For example, a verb and its direct object are adjacent in a syntactic-link graph of a sentence. Although Evert [2008] defines these cooccurrences as requiring the components to be directly adjacent, researchers have also

considered non-adjacent cooccurrences based on paths within syntactic-link graphs [Baroni and Lenci, 2010, Padó and Lapata, 2007]. Syntactic cooccurrences can be particularly useful for examining long-distance relationships within the sentence, which might otherwise be obscured by the noise introduced with a larger span size. While syntactic cooccurrences offer more precision, they also require preprocessing of the input, which can be both computationally expensive and error-prone, and which may impose assumptions.

2.4.4 Association Measures

The choice of C defines how the cooccurrence frequency statistics are compiled. These frequency statistics, in turn, can be used as one of the inputs to an Association Measure. Association Measures ensure that the value of the feature in a vector is meaningful, the necessity of which will be explained in Sec. 2.4.4.2.

The other input to the Association Measure is the *expected* number of cooccurrences. This value is computed by the statistical model. The statistical model is not a free parameter and is entirely determined by C, the choice of cooccurrence type. Nevertheless, different statistical models yield different expected values.

2.4.4.1 The Statistical Model

Essentially, the statistical model is a description of the sorts of instances under consideration using class variables. For example, if a set of people varied according to weight (fat/thin) and height (short/tall), the class variables would be *weight* and *height*. If we are given the total number of people for each *weight* and for each *height*, we could compute the expected number of *tall thin people* under the assumption that *thin people* were evenly distributed between *short* and *tall*. In the case of cooccurrence statistics, the set of classes we choose depends on the definition of cooccurrence. The input to the statistical model, however, is always the same. The input, called the frequency signature, consists of four different numbers: the observed frequency, the marginal (class) frequencies, and a sample size. In terms of notation, I use that of [Evert, 2008]: O will stand for observed frequency of the cooccurrence, f_1 and f_2 will stand for marginal frequencies for the row element and the basis element respectively and N will stand for the sample size.

As the details of the statistical model are not essential to this work, I review only the statistical model for one type of cooccurrence, syntactic coccurrences, to give a general sense of the procedeure. For a thorough review of statistical models, see [Evert, 2008].

The statistical model for syntactic cooccurrences treats every sort of unique syntactic relation as independent, and thus the marginals (sum of frequencies for each class) for each syntactic relation are independent as well. Fig. 2.2 shows the procedure for

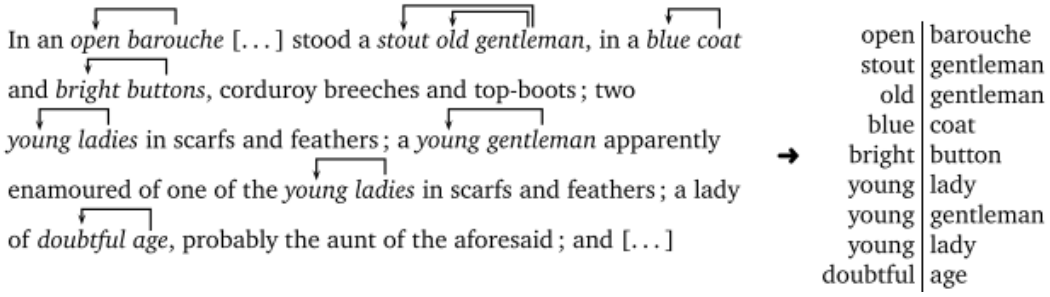

Figure 2.2: Figure illustrating the syntactic cooccurrences of nouns and their prenominal adjectives, from [Evert, 2008]. The arrows point from nouns to prenominal adjectives. The table collects all of these cooccurrences to facilitate the calculation of the frequency signature. For the pair $\langle young, gentleman \rangle$, O is 1, f_1 and f_2 are 3 and 3 and N is 9.

calculating the frequency signature for $\langle gentleman, amod - young \rangle$, where *amod* is a syntactic label denoting adjectival modification. The observed frequency O is the number of times *young* is modifies *gentleman* in a manner that would be parsed as *amod*, which is 1.[4] The marginal frequency f_1 is the number of times *gentleman* is modified by any adjective via *amod*, which is 3. The marginal frequency f_2 of $amod-young$ is the number of times *young* modifies *any* word via *amod*, which is 3 in this case. Finally, the sample size N is the number of times any adjective modifies any noun via *amod*, which is 9. Thus the frequency signature of $\langle gentleman, amod-young \rangle$ is (1,3,3,9).

The frequency signature contains all of the information required to compute the expected value. For syntactic cooccurrences, one computes the expected value using E_2 in Fig. 2.3, which can also be used with textual cooccurrences, though the arguments will be computed differently. The formula for computing the expected value for surface cooccurrences, E_1, is only slightly different.

2.4.4.2 Using the Statistical Model

As alluded to earlier, Association Measures perform a critical function, by transforming raw frequency into a quantity that is statistically robust. Because the meaning of feature weights will be an important part of discussion later, I review the motivation for using Association Measures.

Why *not* use frequency? While it is tempting to interpret the raw number of cooccurrences between a row element and a basis element as a measure of relatedness, there are two reasons why this number is not meaningful. Firstly, the raw number of cooccurrences is a function of the corpus from which the data are drawn,

[4]While in the case of an adjective modifying a noun, there may only be way in which the two could be syntactically related, for other combinations of parts-of-speech, there may be more than one way, in which case the specific syntactic relation is important.

$$E_1(f_1, f_2, N) = \frac{k \times f_1 \times f_2}{N}$$
$$E_2(f_1, f_2, N) = \frac{f_1 \times f_2}{N}$$

Figure 2.3: Figure showing how to compute the expected value E for all sorts of cooccurrences, where f_1 and f_2 refer to the marginal frequencies of the two classes and N refers to the population size. E_1 is the formula for computing the number of expected cooccurrences for surface cooccurrence and k represents the span size. E_2 is the formula for computing the number of expected cooccurrences for textual and syntactic cooccurrences. For a more thorough derivation of these formula, see [Evert, 2008].

whereas what is needed instead is information that generalizes to all potential sample corpora, and which is thus statistically robust. Secondly, the raw number of cooccurrences is highly sensitive to the raw frequencies of both individual terms, independent of their cooccurrence with each other. For example, in the Contemporary Corpus of American English [Davies, 2008], the coocurrence $\langle is, the \rangle$ occurs roughly 90,000 times, making it a very common co-occurrence. However, both words independently are also very frequent: *the* occurs 11.5 million times and *is* occurs 4.2 million times. If the corpus were the result of some random process, and there were, as a result, no meaningfully related pairs of words, we would expect to see $\langle is, the \rangle$ about 100,000 times, which is only about 10% more often than chance alone would predict.

The number 100,000 is not a subjective judgment but is actually a mathematical result; it is the output of a statistical model like the one derived in Sec. 2.4.4.1. In this case, these numbers were computed using the following procedure. The term *the* occurs roughly 54 times for every 1000 words. If there were no relationship between *the* and *is*, then every time *is* occurred, there would be a 54 in a 1000 chance that the next would be *the*. Thus, the expected number of $\langle is, the \rangle$ cooccurrences is the number of times *is* occurs multiplied by the rate at which *the* occurs. In the case of $\langle is, the \rangle$, there is in fact no relationship between the two at all, as the observed number of cooccurrences and the expected number under the assumption of independence are close. In contrast to $\langle is, the \rangle$, the cooccurrence $\langle burn, victim \rangle$ occurs 48 times more often than chance would predict, suggesting that there is far more evidence to reject the hypothesis that the two words are not meaningfully related.

What is instead needed to replace raw frequency is some sort of comparison between the observed frequency and what we would expect if there were no relationship between the basis element and the row element. In this section, the responsibility

$$PPMI(O,E) = max(MI(O,E),0)$$
$$MI(O,E) = \log_2 \frac{O}{E}$$
$$simple-ll(O,E) = 2\Big(O \times \log_2 \frac{O}{E} - (O-E)\Big)$$

Figure 2.4: The formula for various Association Measures, each of which accepts as input O, the number of observed cooccurrences, and E, the number of expected cooccurrences under independence.

of the Association Measure is explained in greater detail and examples of different design choices are reviewed.

The responsibility of the Association Measure is to separate the signal of meaning from the noise. This means that ideally, the measure would indicate both (1) the strength of the relationship between the row and basis element and (2) how statistically significant the strength is. As Evert [2008] explains, these two ideas are related but not the same. Many true cooccurrences will both have a strong positive relationship and occur statistically more often than chance; however, it is also possible for the weak co-occurrence of infrequent words to be highly significant (small highly significant effect) and for the number of observed cooccurrences to vastly exceed their expected value, even while still being statistically insignificant (big insignificant effect).

In practice, association measures must make a trade-off between these criteria. Those that focus on the former are referred to as *effect-size measures* whereas those that focus on the latter are *significance measures*. Effect-size measures fail to account for sampling variation and thus over-estimate the importance of a cooccurrence when E is small, while significance measures tend to exaggerate the importance of relatively small differences between O and E when O is large [Evert, 2008]. A survey of all potential association measures is beyond the scope of this work; for such a review, see [Evert, 2008]. Instead, I explain how a few common measures of association are constructed and the trade-offs they make with respect to effect size and significance.

All association measures must somehow relate O, the number of observed cooccurrences, and E, the number of expected cooccurrences. The simplest way is to use their ratio; however, this quantity is problematic when E is small, which is often the case with rarely occurring row or basis elements, as the resulting ratio of O and E becomes very large. Consequently, it is practical to take the logarithm of their ratio, which shrinks the value of the ratio monotonically. The pointwise mutual information measure (MI) uses the base-2 logarithm, the motivation for which stems from information theory. The resulting value can be interpreted as the number of bits of "shared information" [Church and Hanks, 1990] and is always a real number. It is

collocate	f	f_2	simple-ll
water	184	37012	1083.18
a	590	2164246	449.30
spade	31	465	342.31
plastic	36	4375	247.65
size	42	14448	203.36
slop	17	166	202.30
mop	20	536	197.68
throw	38	11308	194.66
fill	37	10722	191.44
with	196	658584	171.78

collocate	f	f_2	t-score
a	590	2164246	15.53
water	184	37012	13.30
and	479	2616723	10.14
with	196	658584	9.38
of	497	3040670	8.89
the	832	6041238	8.26
into	87	157565	7.67
size	42	14448	6.26
in	298	1937966	6.23
record	43	29404	6.12

collocate	f	f_2	MI
fourteen-record	4	4	13.31
ten-record	3	3	13.31
multi-record	2	2	13.31
two-record	2	2	13.31
a-row	1	1	13.31
anti-sweat	1	1	13.31
axe-blade	1	1	13.31
bastarding	1	1	13.31
dippermouth	1	1	13.31
Dok	1	1	13.31

collocate	$f \geq 5$	f_2	MI
single-record	5	8	12.63
randomize	10	57	10.80
slop	17	166	10.03
spade	31	465	9.41
mop	20	536	8.57
oats	7	286	7.96
shovel	8	358	7.83
rhino	7	326	7.77
synonym	7	363	7.62
bucket	18	1356	7.08

Figure 2.5: Collocates of *bucket* in the British National Corpus according to the association measures simple-ll, t-score, MI, and MI with frequency threshold $f \geq 5$, from [Evert, 2008]. t-score is another significance test and it is defined as $\frac{O-E}{\sqrt{E}}$.

common practice after [Bullinaria and Levy, 2007] to let this value range from 0, by taking the max of MI and 0, removing from the dataset any "anti-collocations" [Evert, 2008] (row and basis elements that seem to repel each other). The resulting quantity is called the positive pointwise mutual information (PPMI). In practice, MI often awards greater-than-desired significance to low-frequency word pairs when $E << 1$.

This tendency to assign high association scores to cooccurrences where $O >> E$ marks MI as an effect size measure; MI does not actually weigh the amount of evidence. In contrast, simple log-likelihood is an association measure that measures significance on a standardized scale known as the chi-squared distribution with one degree of freedom.

The choice of association measure can have a profound effect on association scores, as is apparent by comparing the different sets of top-ranked features in Fig. 2.5. Only three collocates (*water*, *a*, and *spade*) appear in more than one list. Additionally, the tendency for MI to reward rare events is clearly evident in the difference between the two bottom-most tables.

$$Minkowski(\vec{u},\vec{v}) = \left(\sum_{i=1}^{n} |u_i - v_i|^N\right)^{\frac{1}{N}}$$
$$F(\vec{u}) = \{u_1, u_2, u_3, ...u_n\}$$
$$Jaccard(\vec{u},\vec{v}) = \frac{|F(\vec{u}) \cap F(\vec{v})|}{|F(\vec{u}) \cup F(\vec{v})|}$$
$$cos(\vec{u},\vec{v}) = \frac{u \cdot v}{|u| \times |v|}$$

Table 2.1: A small sample of similarity measures. The input to each similarity measure is a pair of vectors, $\vec{u}$ and $\vec{v}$.

2.4.5 Similarity Measures

The next DSM parameter is S, the similarity (or, equivalently distance) measure, which accepts two row vectors and returns a quantity representing how similar they are. Many such functions have been empirically compared [Weeds, 2003] and interactions between the choice of similarity measure and other DSM parameters can be significant [Lapesa and Evert, 2014]. Typically, in comparing the quality of a given similarity measure, one tests the degree to which the geometric metaphor of meaning holds in the space. To do this requires pairs of rows with either known similarity or known relative similarity. A number of well-established benchmark tasks meet this description. The TOEFL task introduced by Landauer and Dumais [1997] requires models to identify a target word's synonym in a list of candidates. The similarity judgment task, for which there are many benchmarking datasets [Bruni and Gatica-perez, 2013, Hill et al., 2014, Rubenstein and Goodenough, 1965], compares the similarity scores generated by comparing two row vectors using a similarity measure against human judgments on a Likert scale (or, in the case of MEN, using a slightly different procedure) and models are compared with respect to the correspondence between their scores and the humans'.

There are many potentially suitable functions that can be used to measure similarity and Table 2.1 lists a small sample. The most common is cosine similarity, which is the inner product of each vector after it has been normalized. The advantage of cosine similarity is that it limits the effect of large values, which, as already mentioned, often occur when the expected value is very small in MI derived association measures. Set-theoretic distance functions like Jaccard's Coefficient, which ignore magnitude entirely and count shared and non-shared features were also considered, prior to the ascent of cosine similarity, as was the family of geometric distance functions, such as the Minkowski distance functions, which are noteworthy for their sensitivity to association score magnitude. Euclidean distance is a special case of the Minkowski distance function when $N = 2$.

2.4.6 Smoothing Methods

Smoothing is useful for a number of different reasons. One reason is that it can increase the speed at which vectors can be compared by decreasing the density of the cooccurrence matrix. Lin [1998] showed that there was little loss of precision in the similarity scores of vectors even after removing all association scores below a threshold.

Smoothing can also be treated as a way of improving the semantic properties of the vectors in the space and, because the procedures described in Chapter 6 can be interpreted as acting in a similar manner, I review one popular method of smoothing.

SVD is a popular method of smoothing in both information retrieval, where it is known as Latent Semantic Indexing [Deerwester et al., 1990], and in computational semantics, where it is known as Latent Semantic Analysis [Landauer and Dumais, 1997]. Deerwester et al. [1990] used truncated Singular Value Decomposition (SVD) on a term-document matrix, and showed that it improved semantic similarity characteristics. Landauer and Dumais [1997] applied the truncated SVD to a synonym detection task, and showed that models improved to near-human performance.

Turney and Pantel [2010] suggest four ways of interpreting the impact of truncated SVD on DSMs, two of which can already be used to interpret the procedures described in Chapter 6. The first of these is as a means of identifying high-order co-occurrence. While two vectors that share a feature in the original space must have occurred with the *same* basis element, two vectors in the reduced space that share a feature occurred with *similar* basis elements. This proves invaluable when the set of basis elements are inter-related. The final way of interpreting SVD is a method of sparsity reduction. The sparsity of cooccurrence matrices may be a consequence of limited data, so SVD can be seen as a way of simulating missing text.

2.4.7 Other Parameters

There are many additional parameters beyond the six tuple. Several important parameters govern steps prior to the statistical calculations. For example, the raw frequency matrix is often filtered to remove rows or columns with small marginals.

The corpus itself is a parameter, as are the pre-processing steps typically done to it before it is used by the DSM. The pre-processing of corpora consists of one or more processes including tokenization, normalization, and annotation.

Tokenization divides the corpus into tokens. In English, as we've already seen, whitespace is often but always the delimiter between tokens. In some languages (e.g., Chinese), words are not delimited by whitespace.

Normalization consists of two different processes: case folding and lemmatization. Case folding converts words to lower case. Lemmatization removes grammatical

suffixes, leaving words in their root form, e.g., *chews* $\mapsto$ *chew*. Case folding can throw away potentially valuable information when capitalization is an informative marker as in many capitalized acronyms or proper nouns. Lemmatization is fairly accurate in English, which has a simpler and more regular morphological system than many other languages; lemmatization is more complicated in languages with complex morphological systems, where single words take on more complex meanings equivalent to a sequence of words in English [Turney and Pantel, 2010].

Annotation adds additional information to the tokens, such as part-of-speech tags, sense tags (which require a sense inventory and an accurate means of word sense disambiguation), and syntactic information.

3 Lexical Relations and WordNet

3.1 What are Lexical Relations?

There are many different sorts of lexical relations. Given that the task at hand is the identification of instances of lexical relations, it's relevant to review what a lexical relation is. To understand what a lexical relation is, Cruse [2004] points out, it is helpful to consider what it *isn't*. Cruse provides some guidelines that, while not precise, are still helpful. For example: why is the relationship between dog and banana, which we might call *dogbananonymy*, not a lexical relation?

Firstly, *dogbananonymy* doesn't recur often enough, and only relations that occur sufficiently often are worth assigning a name. You could argue that only relations between two frequently co-occurring words would meet this criteria, but this ignores the fact that, unlike *dog* and *banana*, other relations can be said to belong to classes of identical relations; thus, even though *cat* is related to *banana* in a similar fashion as *dog* is, this relationship is still not the same.

Conversely, a relation which was universal, and held between all pairs of words, would have no discriminatory power. Cruse provides an example of a relation that doesn't discriminate: "can occur in the same English sentence as..." This relation holds between *all* pairs of words because there are no words that cannot be written in the same sentence.

Taken together, these criteria paint an information-theoretic picture of relations: with regard to what qualifies as a relation, a relation must have non-zero entropy and higher entropy is better.

3.2 Hyponymy and Taxonomies

Given that this thesis is focused on HR, the most important lexical relations are hyponymym and co-hyponymy, which are paradigmatic lexical relations. Hyponymy is the relation of kinds, as in *dogs are* ***a kind of*** *animal*, in which *dog* is the hyponym and *animal* its superordinate or hypernym. Hyponymy is thus an asymmetric relation. Extensionally speaking, hypernyms are broader in that they refer to a broader set of things (animals includes within it dogs). Intensionally, superordinates are less informative [Murphy, 2002] and have more properties. Hyponymy serves as the backbone of many ontologies and is thus sometimes referred to as the taxonomy

relation [Cruse, 2004], although the taxonomy relation is technically not transitive. In some instances, sentences with hyponyms entail other sentences with hypernyms, e.g., *I saw a dog* entails *I saw an animal*. But, *I did not see a dog* does not entail *I did not see an animal*.

In practice, defining in which sorts of sentences hyponymy entails and does not entail is hard [Cruse, 2004]. At the level of words, hyponymy should always be transitive but occasionally it isn't. For example,

1. A hang-glider is a type of glider
2. A glider is a type of airplane, but
3. A hang-glider is a type of airplane

What the individual steps in this inference really rely on is implicit prototypicality, e.g.,

1. A prototypical hang-glider is a type of glider
2. A prototypical glider is a type of airplane
3. A prototypical hang-glider is a type of airplane

the last example which fails because a hang-glider is not a prototypical glider. Thus, hyponymy is itself prototypically transitive, but not always so.

Because it is generally entailing, hyponymy can be used to construct a taxonomy, which is a directed acyclic graph used to express nested categories. In a well-formed taxonomy, each subtree's nodes are mutually exclusive with respect to sibling-level subtrees and, consequently, there should only be one path to the root for every node. In practice, when building taxonomies of actual every-day objects, this condition may be impossible to meet, for reasons we'll soon see. Furthermore, oftentimes, we find ourselves in need of a word that doesn't exist. This tension between taxonomies in Platonic form and taxonomies in practice bears similarity to competing psychological theories of category. Prior to Wittgenstein [1972], the dominant theory of categories was the Aristotelian account, which stipulated that categorical membership could be defined in term of necessary and sufficient criteria. Wittgenstein famously demonstrated the limitations of this approach by asking for a definition of the category *game*, examples of which lack a single set of criteria. Prototype Theory [Rosch, 1973], instead, defines category membership as graded, and a function of distance to a best example, or prototype.

Cruse [2004] demonstrates the challenge these constraints pose to constructing a real taxonomy using a toy example, cutlery, and contrasting it with a more representative example, clothing. A very simple taxonomy of cutlery might use *cutlery* as the root, *fork*, *knife*, and *spoon* nested just below *cutlery* and *teaspoon*, *tablespoon* and *soup spoon* at a third level, nested below *spoon*. At each level of the taxonomy, the set of referents to which the nodes might refer in the world are mutually exclusive; *forks* can't be *knives* or *spoons*, and vice-versa.

In contrast, it is not nearly so simple to construct a taxonomy for clothing, within which domain there are terms whose sets of possible referents overlap. Let's say the set of terms we'd like to include in the taxonomy consists of *hat*, *shoe*, *goggles*, *t-shirt*, *sneakers*, *sandals*, which are easy to visualize, and *headwear*, *footwear*, and *sportswear* which aren't.

This characteristic of being easy to visualize has an indirect effect on the ease with which these sorts of words can be incorporated into a taxonomy. The difference between easy-to-visualize words, which Cruse [2004] calls 'basic-level', and hard-to-visualize words, which are called restricted perspective-terms, is that within the former, the set of referents are uniform enough to have some sort of prototypical example, which we can then visualize, whereas the latter group refers to a set whose members are required only to share a few defining features. For example, all headwear can be worn on the head.

The problems begin when trying to add these restricted perspective-terms to the taxonomy. No matter which is added first, there will be a problem. If, after adding the basic-level terms, *headwear* and *footwear* are added then where would be no way to add *sportswear*; if can refer to things that would be considered both *headwear* and *footwear* (e.g., *helmet* and *cleat*), and thus there is no way to maintain the exclusivity of categories *and* have *sportswear* below *headwear* and *footwear*. But, if *sportswear* were added first, the same problem would arise as there are also certainly referents of *headwear* and *footwear* that are not *sportswear*.

The problem is caused by the fact that terms like *headwear* and *sportswear* are not proper subsets of each other, but rather have both shared and unshared elements. Also interesting to note is the absence of a term to describe clothing not on one's head or foot (torsowear, one cannot say), which is the default, and what is assumed when no specification supplied. As a result of these practical challenges, taxonomies, like WordNet, which will be introduced shortly, are often not well-formed.

3.3 Co-hyponymy

Co-hyponymy is closely related to hypernymy and is predicated upon the incompatibility of concepts. It is also known as Cruse [2004] co-taxonomy. Co-hyponyms are sometimes also called coordinates Baroni and Lenci [2011]. For example, the pair $\langle yacht, sailboat \rangle$ are co-hyponyms that share a semantically close hypernym, *boat*. The formal definition of co-hyponymy varies and in some cases its distinction from other relations is hard to define. For example, in the BLESS dataset, co-coordinates are only those words whose shared semantic class is very similar and the relation *random* is used to refer to word pairs that have a more distant shared semantic class; however, the threshold between what constitutes a *random* pair and what isn't is not explicitly stated.

POS	Unique Strings	Synsets	# Word-Sense Pairs
Noun	117798	82115	146312
Verb	11529	13767	25047
Adjective	21479	18156	30002
Adverb	4481	3621	5580
Totals	155287	117659	206941

Table 3.1: The coverage of WordNet's data, Miller and Beckwith [1990]

3.4 WordNet

In the last twenty years, many resources have arisen to meet the needs of researchers in fields related to linguistics and WordNet is one such resource. WordNet [Miller and Beckwith, 1990] is a large lexical database for English that records, among other things, lexical relations between words. Conceptually, WordNet can be thought of as two different data-structures. Firstly, there is a map between words and sets of senses. For example, the word *hug* maps to a set consisting of the nominal sense hug_1 (gloss: a tight or amorous embrace) and two different verb senses, hug_2 (to hold (someone) tightly in your arms, usually with fondness) and hug_3 (to fit closely or tightly). Each sense of *hug* belongs to a different synset, which consists of all of the words that connote that sense. For example, the synset to which hug_1 belongs includes also $clinch_5$ and $squeeze_7$. Synsets, in turn, belong to a rich, graph-like structure, in which they comprise the nodes and semantic relations comprise the edges. These edges are directed for asymmetric relations and undirected for symmetric relations.

The designers of WordNet recognized that each part-of-speech class had its own unique characteristics. This fact is reflected in the graph: the edges used for each synset vary as a function of the part-of-speech class, as does the high-level structure of the nodes of that part-of-speech class. Nouns are organized into a hierarchical taxonomy, the root of which is *entity*. The topmost levels of the noun taxonomy are semantically empty and are hard to lexicalize, but their inclusion allows all nouns to fit into the same structure. Below these levels lie 25 beginner trees corresponding to generally (though not entirely) mutually exclusive concepts. Fig. 3.1 presents a small piece of the WordNet taxonomy, showing how some more semantically specific words would be nested beneath more general terms.

By traversing the graph, it is trivial to follow entailed senses transitively toward the root. Additionally, though Fig. 3.1 does not reflect this, some senses have more than one hypernym, which means that there may be more than one path between nodes in the graph (just as *helmet* is both *headwear* and *sportswear*). Typically, the shortest path is used for computing distances in the graph. In this work, I frequently present results as a function of naïve generality (NG), which refers to the number of edges between a synset and the root of the noun taxonomy, under the naïve [Resnik,

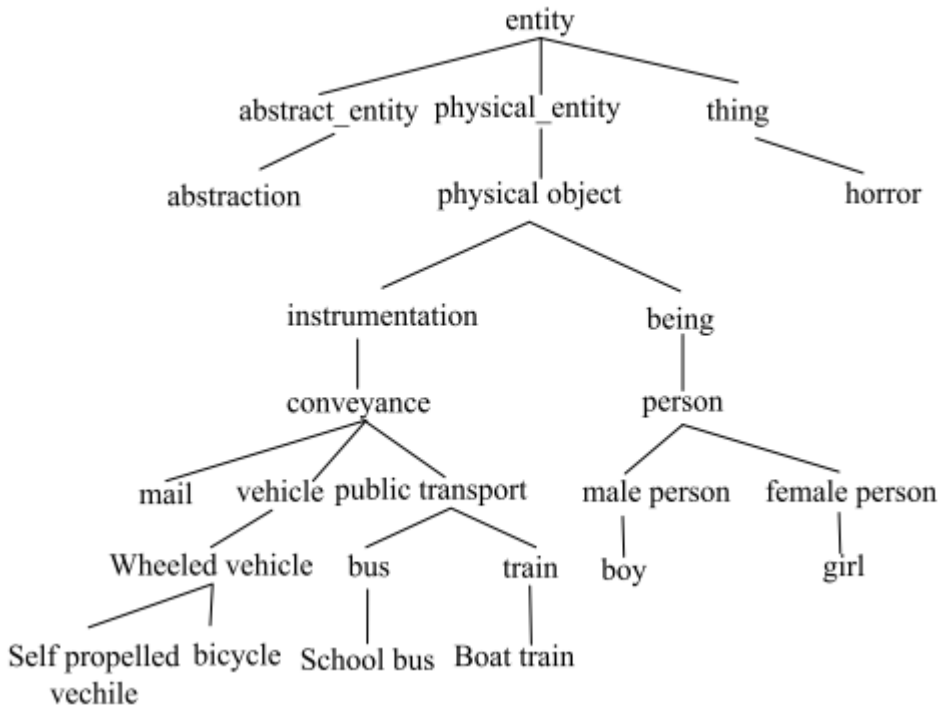

Figure 3.1: A portion of the WordNet noun taxonomy, Miller and Beckwith [1990]

2011] assumption that all edges are of equal semantic length.

The edges of verb synsets are similar to noun synsets in that entailment is still supported but the sort of entailment reflected in the graph of verb synsets extends beyond hypernymy to include more general entailment relations. More important for our purposes is the fact that the high-level structure of verb synsets is not a single taxonomy but rather a number of separate taxonomies, each of each which is shallower and has a shape that has a bulge, which is the depth at which there is the most lexicalization.

Adjectives and adverbs do no support entailment and so cannot be exploited to the degree that verbs and adjectives are. However, they both still support synonymy.

4 Recognizing Lexical Relations

In this chapter, I explain in detail the Distributional Inclusion Hypothesis (DIH), which is the theoretical basis for all models of Hypernymy Recognition (HR) that use feature inclusion. I then survey models that implement the DIH and measure their performance, to illustrate how the status quo does not outperform crude baseline models. The Feature Exclusion Problem (FEP), which I consider to be an important factor limiting performance, is presented and analyzed in Chapter 5.

4.1 The Distributional Inclusion Hypothesis

The task of identifying asymmetric relations like hyperny has been attempted using a variety of approaches, including semi-supervised approaches based on shallow lexico-syntactic features [Hearst, 1992], supervised learning models [Levy et al., 2015, Roller et al., 2014, Santus et al., 2014, Snow et al., 2006, Weeds et al., 2014a] and unsupervised models [Clarke, 2009, Geffet and Dagan, 2005, Kotlerman et al., 2010, Santus et al., 2014, Szpektor and Dagan, 2008].

One of the most prominent hypotheses guiding the design of many supervised and unsupervised models is the DIH [Weeds, 2003, Weeds and Weir, 2005], which is supported empirically by the observation that more general words tends to occur in a larger variety of contexts than do more specific words. The DIH is also consistent with definitions of hypernymy as feature inclusion in semantics [Cruse, 2004]. The DIH was formalized by Geffet and Dagan as follows: Given words $\langle A, B \rangle$, and $f(w)$, a function that determines for a sense its most important features, and that $A \rightarrow B$ denotes that A entails B, then $A \rightarrow B \equiv f(A) \subset f(B)$.

Models operationalizing the DIH typically consider not only the proportion of features shared by both narrower and broader terms but also the proportion of excluded features from the broader term, the idea being that not only are the characteristic features of the narrower term a subset of the broader term's, but they are a much smaller subset relative to the size of the broader term's characteristic feature set. This goes beyond the Geffet and Dagan formalization, in that (1) it implicitly establishes the root of the taxonomy (the word which itself has no hypernym but which hypernym to all other words) as a word whose characteristic contexts consist of all possible features and (2) by not discriminating objectively on the basis of the number of features in either the narrower or broader term (but instead discriminating with respect to proportions of features), it implicitly rewards extremely narrow

terms as candidate hyponyms. While semantically this latter idea might seem plausible, in practice it is problematic in a distributional setting, as narrowness is a consequence of not just specificity of meaning but also rarity of usage. Kotlerman et al. [2010] consider the practical challenges of sparse feature vectors in their list of desired properties for asymmetric measures of distributional similarity, and note that sparse feature vectors are less reliable. Their measure, which will be reviewed shortly, is among the few to penalize extremely sparse feature vectors.

4.1.1 Challenges in Interpreting Feature Weight Semantically

The Geffet and Dagan formalization doesn't specify the details of the characteristic feature function $f()$ and this may be because researchers in DS are not in complete agreement about what the features in the sparse feature vector of a DSM actually signify. The vast majority of researchers interpret the magnitude of a feature's weight as an important piece of information and typically also interpret this weight as being proportional to the feature's importance to the sense that the vector is intended to represent (insofar as a feature's weight is typically directly proportional to the output of most similarity measures). In qualitative examinations of top-ranked features, such as was done in Fig. 2.4, the hypothetical gold standard consists of features that are intuitively related to the sense being represented.

This intuitive standard may not be a goal worth striving for. Semantics and the DIH are not intended human-specific theories; their essence is independent of humanity. However, the representations in a DSM exhibit many artifacts of human cognition, as do our intuitions. T

It is important to mention that not only is this view of feature weight one of many, but that the issue of how to interpret feature weight bears remarkable similarity to a similar debate that took place among semanticists and philosophers about the nature of conceptuality. In one view of concepts, which I refer to as the Set-Theoretic view, concepts are defined by a set of necessary features that are of equal importance. Alternately, there is the Prototype View of concepts, in which the importance of features to a concept varies, and there may be no single set of sufficient features. Thus, while the prevailing view is that features vary in importance to their vectors, one could also treat features as having equal importance, a stance that embraces a Set-Theoretic View.

Despite the consensus that feature weight is important somehow, there is substantial disagreement about how it should be assigned and used. There is still debate about which Association Measure is best (though positive point-wise mutual information is popular, it is not universally dominant). Also, a number of researchers have explored transformations of feature weight that, while monotonic, and thus order-preserving, completely change the magnitude of features. And these researchers argue that these transformations improve the semantic properties of vectors. Lapesa and Evert [2014] consider the impact of log and square-root and find that these transformations

improve the scores on standard benchmarks. There are also researchers who take a more conservative approach to feature weight and conceptuality, and use feature *rank* rather than the weight itself. Kotlerman et al. [2010] rank features by weight, and use the ranks as indicators of importance. They note in defense of ranking as a more stable measure that two features of consecutive rank may have very different weights and similarly, features of the same rank in different vectors may have very different weights.

It remains to be seen how the field will view feature weight in the future. However, as will be clear after reviewing the performance of some baseline models, at least in HR, feature weight does not contribute that much.

4.1.2 Defining the Characteristic Function

All models of HR must implicitly define $f()$, the characteristic function that returns a word's most important contexts. Most models adopt the Prototype View and treat the magnitude of feature weight as proportional to importance. There are data in HR that seem to show that feature weight should be considered. Kotlerman et al. [2010] showed that the weight of features can be used to distinguish between entailing and non-entailing pairs of vectors. See Fig. 4.1. Two baseline models I include, $\mathrm{P}_{\mathrm{Set}}$ and $\mathrm{R}_{\mathrm{Set}}$, ignore feature weight and treat every non-zero feature as having equal importance.

4.2 Models of the DIH

In this section, I describe many recent or historically important HR models that use feature inclusion to discriminate positive and negative instances of hypernymy.

The first model to use feature inclusion is that of Weeds and Weir [2005] and views feature inclusion from an Information Retrieval (IR) perspective, in terms of precision and recall. In IR, a system returns a set of documents in response to a query and we can evaluate the system's quality based on the number of true positives (relevant documents returned), false positives (irrelevant documents returned), false negatives (relevant documents not returned) and true negatives (irrelevant documents not returned). Under this view, non-zero features are akin to relevant documents. However unlike with precision/recall in IR, the Weeds and Weir model adopts the Prototype View and considers relevance to be graded. Given that $\vec{u}$ and $\vec{v}$ represent the hyponym and hypernym respectively, and $F(\vec{u})$ is set of non-zero features for $\vec{u}$,

Both $\mathrm{P}_{\mathrm{Weeds}}$ and $\mathrm{R}_{\mathrm{Weeds}}$ effectively measure proportions of total importance. Thus the entire model can be summarized as predicting that the percentage of importance from shared features in the narrower term should be greater than the percentage of

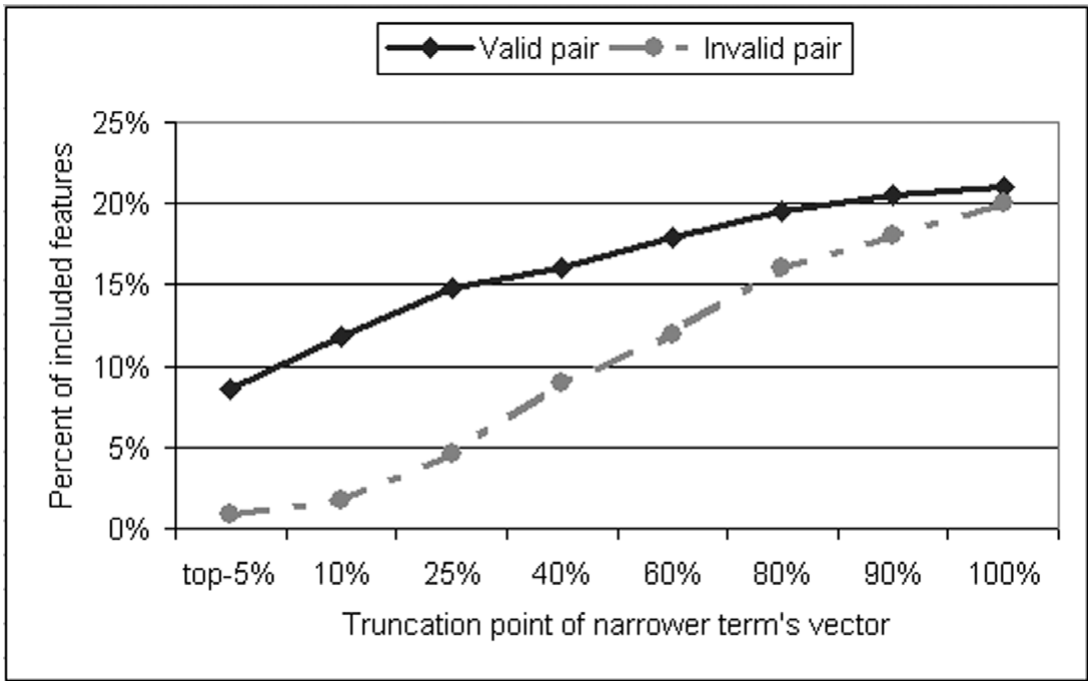

Figure 4.1: This figure presents feature inclusion, the extent to which features of the vector representing the narrower term are shared by the broader term's vector, as a function of feature rank within the narrower term for two pairs of vectors, one pair of which represent an entailing pair of words, *election* $\rightarrow$ *vote*, while the other represent a non-entailing pair of words *election* $\not\rightarrow$ $reform$. The difference between the lines decreases with feature rank, suggesting that high-rank features are useful for discriminating between entailing and non-entailing pairs. From Kotlerman et al. [2010].

$$F(\vec{u}) = \{u_1, u_2, u_3, ...u_n\}$$
$$w(f \in F(u), u) = PPMI(u_f)$$
$$\mathrm{P}_{\mathrm{Weeds}}(\vec{u}, \vec{v}) = \frac{\sum_{f \in F(u) \cap F(v)} w(f, u)}{\sum_{f \in F(u)} w(f, u)}$$
$$\mathrm{R}_{\mathrm{Weeds}}(\vec{u}, \vec{v}) = \frac{\sum_{f \in F(u) \cap F(v)} w(f, v)}{\sum_{f \in F(v)} w(f, u)}$$
$$\mathrm{WeedsDiff}(\vec{u}, \vec{v}) = \mathrm{P}_{\mathrm{Weeds}}(\vec{u}, \vec{v}) > \mathrm{R}_{\mathrm{Weeds}}(\vec{u}, \vec{v})$$

Figure 4.2: The composition of WeedsDiff, $\mathrm{P}_{\mathrm{Weeds}}$ and $\mathrm{R}_{\mathrm{Weeds}}$. $F(\vec{u})$ is the feature weight function, which returns the set of non-zero weights of a feature vector and $w(f, n)$ is the feature weight function, which returns the PPMI feature weight of a given component of a feature vector.

$$P_{Clarke}(\vec{u},\vec{v}) = \frac{\sum_{f \in F(u) \cap F(v)} min(w(f,v), w(f,u))}{\sum_{f \in F(u)} w(f,u)}$$

$$R_{Clarke}(\vec{u},\vec{v}) = \frac{\sum_{f \in F(u) \cap F(v)} min(w(f,v), w(f,u))}{\sum_{f \in F(u)} w(f,v)}$$

$$ClarkeDiff(\vec{u},\vec{v}) = P_{Clarke}(\vec{u},\vec{v}) > R_{Clarke}(\vec{u},\vec{v})$$

Figure 4.3: The composition of Clarke's P_{Clarke} and R_{Clarke} models

relevance from shared features in the broader term. In evaluations, the Weeds model was shown to be 71% accurate, but not significantly better than a näive model based on word frequency Weeds and Weir [2005].

Clarke [2009] proposed a variation on the Weeds model which treats feature vectors as approximations of frequency distributions. Under such a view, the intersection of any two vectors represents the maximal number of times both words occurred together within each context and, because the maximal number of times clearly cannot exceed the number of times a single word occurred with a particular context (i.e., the value of a particular component in one vector), the intersection vector is 0 for excluded features and takes the minimum value for shared features. Applying this view of shared features to the feature vectors in a DSM, the components of which are not actually frequencies but which have been transformed by a measure of association, yields slightly different definitions of precision and recall: In their evaluation, Weeds et al. [2014a] found $ClarkDiff$ to have comparable or worse performance to the Weeds model.

Lenci and Benotto [2012] compared precision and recall in a different way, and their model was found to be better at discriminating hypernyms from other relations on the BLESS dataset [Baroni and Lenci, 2011], although in similar tests by Weeds et al. [2014a], its performance was neither the best nor significantly better than other unsupervised models. Instead of comparing the magnitude of a precision and recall term, they hypothesize that precision should be high and recall should be low, which is analogous to the idea that narrower terms should be much narrower. Again, the recall function rewards relative narrowness between the two terms, not objective narrowness with respect to general vector width. Their model is:

Szpektor and Dagan [2008] recognized the potential of the Weeds model to reward extremely narrow terms and attempted to mitigate this problem by taking the geometric mean of precision and a similarity measure. The thought was that extremely narrow terms would also exhibit low similarity. This model penalizes distantly related hypernymous pairs.

$$\text{InvCL}(\vec{u},\vec{v}) = \sqrt[2]{P_{Clarke}(u,v) \times (1 - R_{Clarke}(u,v))}$$

Figure 4.4: The composition of the InvCL model

$$BInc(\vec{u},\vec{v}) = \sqrt[2]{Lin(u,v) \times \text{P}_{\text{Weeds}}(u,v)}$$

Figure 4.5: The composition of the *Binc* model

Kotlerman et al. [2010] presented a measure, BalAPInc, that builds upon the Szpektor and Dagan [2008] model. They use the same balancing procedure as Szpektor and Dagan, taking the geometric mean of a measure of similarity with a measure of that captures degree of entailment, but use a new measure of degree of entailment, *APInc*. *APInc* is derived from Average Precision (AP) [Voorhees and Harman, 1998], a measure from IR, which averages precision as a function of order in the returned documents list. *APInc* differs from P_{Weeds} Weeds [2003] in that (1) it models the list of documents (which in this case are contexts) as an ordered list rather than as a set, and (2) it models importance as an ordinal variable that is a function of feature weight, rather than using feature weight itself. Additionally, rather than rewarding high precision and low recall, as did InvCL, *APInc* rewards both high precision and high recall.

To test the importance of feature weight, I include two models that ignore it completely. Their performance may provide insight into the viability of other models.

$$APInc(\vec{u},\vec{v}) = \frac{\sum_{r=1}^{|F(u)|} P(r,\vec{u}) \times rel(f,\vec{u})}{|F(v)|}$$

$$P(r,\vec{u}) = \frac{\sum_{t=1}^{r} t^{th} feature \in F(u)}{r}$$

$$rel(f,\vec{u}) = \begin{cases} 1 - \frac{rank(f,F(u))}{|F(u)|+1} & \text{if } f \in F(u) \\ 0 & \text{if } f \notin F(u) \end{cases}$$

$$\text{BalAPInc}(\vec{u},\vec{v}) = \sqrt[2]{Lin(u,v) \times APInc(u,v)}$$

Figure 4.6: The composition of BalAPInc

$$\mathrm{P}_{\mathrm{Set}}(\vec{u}, \vec{v}) = \frac{|F(u) \cap F(v)|}{|F(u)|}$$
$$\mathrm{R}_{\mathrm{Set}}(\vec{u}, \vec{v}) = \frac{|F(u) \cap F(v)|}{|F(v)|}$$

Figure 4.7: The composition of two Set-Theoretic models, $\mathrm{P}_{\mathrm{Set}}$ and $\mathrm{R}_{\mathrm{Set}}$.

Recently, another hypothesis concerning distributional features distinguishing hypernyms from hyponyms was proposed. Santus et al. [2014] designed a measure, SLQS, as a measure of semantic generality, and not specifically as a model of hypernymy. The measure exploits the fact that more specialized words (narrower terms) tend to take more informative arguments. The measure compares the median entropy of typical arguments for both the proposed narrower and broader terms. ' the fact that it is a model of generality and not hypernymy, SLQS was shown to outperform $\mathrm{P}_{\mathrm{Weeds}}$ at both discriminating hypernym pairs and recognizing the direction of entailment on pairs extracted from the BLESS dataset [Baroni and Lenci, 2011]. However, this thesis is concerned with improving performance on DIH-inspired models.

Thus far, the models that have been reviewed have been unsupervised and have been operationalized in unreduced space, where feature vectors are sparse. There have also been attempts to express the DIH in dense, reduced space (the space after smoothing with, for example, truncated SVD) and to use supervised learning methods to induce a model that is more generalizable and better performing. Doing so requires the modeler to make a decision about the input to the model. Weeds et al. [2014a] experimented with various features, including binary and unary operators between pairs of vectors, as inputs to their supervised models, and were able to achieve 15% reduction in error compared with unsupervised approaches. Roller et al. [2014] use the normalized difference between vectors as features for a support vector machine and were able to achieve state-of-art results on the BLESS dataset [Baroni and Lenci, 2011]. However, as per both the analysis by Roller et al. [2014] as well as additional work done by Levy et al. [2015], there is reason to suspect the generalizability of what these supervised models actually learn. As per Roller et al. [2014], the features learned by their model do not reflect a general comprehension of feature inclusion, but instead are dependent upon the dataset. For example, a classifier that sees that many hypernyms have the feature animal will grant that feature additional weight. BLESS, in fact, seems a dataset particularly encouraging of this sort of mistake, as its instances are not evenly distributed throughout semantic space but are actually concentrated in a small number of semantically related categories (e.g., birds, appliances). Levy et al. [2015] tested whether supervised models are actually lexically memorizing by comparing the performance of models provided only with lexical features and models provided with both contextual features and lexical features, and they found that the difference, though favoring the latter models, was small. In a subsequent test designed to test whether the model was learning to recognize prototypical hypernyms (words close to the root of the

taxonomy, that tend to be hypernyms by virtue of their generality) and to a lesser extent, prototypical instances (words at the bottom of the taxonomy, that tend to be hyponyms for this reason), Levy et al tested a trained classifier on a synthetic test set in which hyponyms were paired randomly with hypernyms and showed a high linear correlation between recall of the synthetic (and presumably incorrect) instances and error in the actual test set.

4.3 Measuring State-of-the-Art Performance

In this section, I measure the performance of the models described in Sec. 4.2 on the HR task using the Weeds et al. [2014b] dataset, the virtues of which dataset are discussed in greater detail in Sec. 8.1.1. The performance of these models is high, but not significantly better than baseline models that ignore feature inclusion.

The models used in this test are all unsupervised but because in most cases a parameter is needed to define a decision boundary, training was done through five-fold cross-validation. The final parameter used is an average of the optimal parameter from each fold.

4.3.1 Vector Representations

Given the impact of cooccurrence definition (surface, syntactic, etc.) on a DSM, and that in the status quo, optimal parameter settings are still a matter of debate, two DSMs are constructed and used as the basis for further modification and analysis. The first of these, **U**, models coccurrence using surface distance whereas the other, **Y**, models coccurrence using distance in a syntactic representation. Both **U** and **Y**

Model Name	Description
$\mathrm{P}_{\mathrm{Weeds}}$	$\mathrm{P}_{\mathrm{Weeds}}(\vec{u},\vec{v}) > p$
$\mathrm{R}_{\mathrm{Weeds}}$	$\mathrm{R}_{\mathrm{Weeds}}(\vec{u},\vec{v}) > p$
$\mathrm{P}_{\mathrm{Set}}$	$\mathrm{P}_{\mathrm{Set}}(\vec{u},\vec{v}) > p$
$\mathrm{R}_{\mathrm{Set}}$	$\mathrm{R}_{\mathrm{Set}}(\vec{u},\vec{v}) > p$
InvCL	$\mathrm{InvCL}(\vec{u},\vec{v}) > p$
WeedsDiff	$\mathrm{WeedsDiff}(\vec{u},\vec{v})$
BalAPInc	$\mathrm{BalAPInc}(\vec{u},\vec{v}) > p$
SingleWidth	$\lvert F(\vec{u})\rvert > p$
WidthDiff	$abs(\lvert F(\vec{v})\rvert - \lvert F(\vec{u})\rvert) > p$
Cosine	$cosine(\vec{u},\vec{v}) > p$

Table 4.1: A list of HR models. For models that do not return a prediction, a parameter p is required as a threshold. To compute an optimal value for p for each model, five-fold cross validation was used.

use the same corpus, a concatenation of Wackypedia and ukWaK [Ferraresi et al., 2008]. The corpora are lemmatized and POS tagged using TreeTagger [1] and the sentences are parsed using MaltParser [2].

4.3.1.1 U

U is meant to represent a rather standard, surface distance-based DSM. It uses a symmetric context window spanning two words and ignoring sentence boundaries. The set of basis elements are lemmatized, POS-tagged, open-class words (nouns, adjectives, adverbs, and verbs) and the set of row elements are all nouns. The space was filtered from its original size to include only the top 100,000 features by frequency and only rows with corpus frequency greater than 300. The features were weighted using PPMI.

4.3.1.2 Y

Y is meant to represent a rather standard, syntax-based DSM. Cooccurrence were considered to have occurred between two open-class words connected by dependency from the set of major open-class dependencies (*nsubj*, *dobj*, *iobj*, *conj*, *amod*, *nmod*). Additionally, all cooccurrence is treated as being direction-independent, but direction is encoded with an additional tag, e.g. *nsubj-r*. The basis elements are a concatenation of a lemmatized, POS-tagged, open-class word and a dependency from this set e.g., $nsubj\_ball - n$. As in **U**, the space is filtered to include only the top 100,000 features and all rows with frequency greater than 300. The features were weighted using PPMI.

The results are consistent with Weeds et al. [2014a] and show that simple models like WidthDiff are comparable in performance to far more complex models, like WeedsDiff, and even better performing than other BalAPInc. Also surprising is the competitiveness of the P_{Set} model, which is both simple and ignores feature weight completely.

A closer look at performance as a function of the absolute difference in naïve generality[3] (NG) suggests some of the causes for low performance. In Table 4.3, the WeedsDiff model's accuracy is shown for **U** as a function of the absolute difference in NG of the pair of words in the problem. Pairs of words with the same NG are classified 22% less accurately than problems where the words differ in generality. The performance drop cannot be explained in terms of similarity, as the cosine similarity of the word pair seems nearly independent of absolute difference in NG.

[1] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/
[2] http://www.maltparser.org/
[3] Naïve generality refers to the number of edges between a synset and the root of the noun taxonomy in WordNet.

Model	**U**	**Y**
BalAPInc	58	51
WeedsDiff	68	70
Cosine	54	55
InvCL	58	66
P_{Set}	62	70
R_{Set}	49	49
SingleWidth	66	66
P_{Weeds}	63	69
R_{Weeds}	49	49
WidthDiff	67	70
Totals	59	61

Table 4.2: Percentage Correct for all models and DSMs

Abs. Diff. in NG	Percentage Correct	Sample Size	Cosine Similarity
0	50	462	0.09
1	67	780	0.11
2	75	541	0.10
3	75	227	0.09
4	87	104	0.10
5+	75	71	0.11

Table 4.3: Percentage of correctly classified problems as a function of the absolute difference in NG of the input for the WeedsDiff model

As has been noted in other places, NG is crude and its crudeness reveals itself here, as some of the word pairs with no difference in NG between are actually pairs that should be classified as hypernyms, which is impossible for words at the same depth in the taxonomy. After dividing the problems by their actual lexical relationship, as recorded in the dataset, over 80% of pairs with absolute difference in NG of 0 are listed as not hypernymous.

Additionally, as per Table 4.4, many of the models reveal themselves to have much higher positive than negative predictive values, where positive predictive value (PPV) is defined as the number of true positives over the number of true positives plus the number of false positives and negative predictive value (NPV) is defined as the number of true negatives over the number of true negatives plus the number of false negatives. This phenomenon might be explained by relative homogeneity: hypernymy is much rarer than co-hyponymy, and thus, compared to the much larger set of co-hyponyms, which is on $O(N)$ where N is the number of words in the taxonomy, the set of hypernyms maybe more homogeneous. The recall-based models also seem to be equivalent to a baseline model that assumes all instances are non-hypernymous.

Model	Negative Predictive Value	Positive Predictive Value	Percentage Correct
BalAPInc	0.42	0.73	0.58
WeedsDiff	0.62	0.73	0.68
Cosine	0.45	0.63	0.54
InvCL	0.55	0.60	0.58
P_{Set}	0.58	0.67	0.62
R_{Set}	0.99	0.00	0.49
SingleWidth	0.64	0.68	0.66
P_{Weeds}	0.54	0.72	0.63
R_{Weeds}	0.99	0.00	0.49
WidthDiff	0.62	0.73	0.67

Table 4.4: Positive and Negative Predictive Values for various models on an HR task

5 Analysis of the Problem

In this chapter, I describe and analyze the causes of the Feature Exclusion Problem (FEP), beginning with the magnitude of the problem, which is large enough to substantiate the claim that in a conventional feature space, models based on feature inclusion are bound to fail.

5.1 What is Feature Exclusion?

Feature exclusion is the phenomenon wherein, given a sparse feature vector representing word w, the features of w are predominantly not shared by h, a sparse feature vector representing a hypernym of w. I refer to features in w but not h and in h but not w as *excluded*. In contrast, features shared by both w and h are referred to as *conserved*, a more general notion than *shared* that can be generalized to any number of entailed hypernyms. Fig. 5.1 presents a visual example of excluded, conserved and semi-conserved features. Feature Exclusion is problematic for the Distributional Inclusion Hypothesis (DIH) because the DIH predicts that the features of the narrower term will be a subset of the broader term's features.

5.2 The Size of the Problem

Before explaining some of the reasons that these features are excluded, it is worth considering the scale of the problem. How often are features actually excluded in the narrower term? Sec. 5.2.1 considers this question by looking at rows in feature-space. Sec. 5.2.2 considers this question by looking at features in feature-space.

5.2.1 Feature Conservation By Rows

This section presents an analysis of feature conservation by rows and shows that when following a sequence of hypernymy relations, few features conserved at one step will be conserved at a subsequent step. Table 5.1 shows the percentage of features as a function of their *degree of conservation* over a sequences of hypernymously related words w_1, w_2, and w_3, such that w_1 is a hyponym of w_2 and w_2 is a hyponym of w_3. The sequences of words used in this sample was limited to

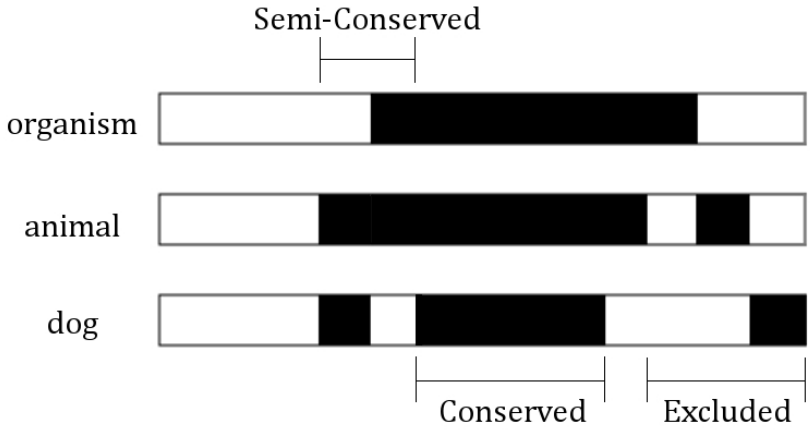

Figure 5.1: A picture depicting three hypothetical sparse feature vectors, where dark areas are non-zero features and white areas are 0. The features shared by all three vectors are labeled *conserved*; some (not all) features shared by more than one vector but not all are labeled *semi-conserved* and features that only occur in one vector are labeled *excluded*.

sequences in which w_1, w_2 and w_3 are found in adjacent levels of the WordNet taxonomy. This constraint controls for semantic distance at the expense of sample size. The sample includes 26,054 sequences from **U** and 21,094 sequences from **Y**[1]. *Degree of conservation* refers to the categories of features in Fig. 5.1. In terms of notation, degree of conservation is represented as a binary sequence, beginning at the hyponym and terminating at the most distant hypernym in the analysis. Thus, the degree of conservation of a conserved feature would be a sequences of ones: 111. The degree of conservation of a feature shared by all but the most distant hypernym would be a sequence of ones followed ultimately by a zero: 110. Because in a given sequences $\{w_1, w_2, w_3\}$, over 90% of features are likely to be zero for all three words (because the space is sparse), and because such features are omitted from the table, the percentages in each row of Table 5.1 do not add up to 100%.

A number of facts are striking about Table 5.1. Firstly, the proportion of features that are conserved is tiny: on average between only 20 and 30 out of 100,000 columns are conserved across the taxonomic span in this sample; had I extended this analysis to a more distant hypernym, the number of conserved features could not possibly have increased and would likely have further *decreased*. Secondly, the percent of Excluded features is significantly greater than either the Conserved or Semi-Conserved features. While theoretically, if the most distant hypernym is at the top of the

[1] **U** and **Y** are DSMs constructed from the same corpus but using different definitions of cooccurrence; **U** uses surface cooccurrence while **Y** uses syntactic cooccurrence. As such, their feature sets are also different; **U**'s features are part-of-speech annotated words while **Y** uses words annotated with *both* part-of-speech tags and a dependency and a direction tag. See Sec. 4.3.1 for more details.

Space	Conserved	Semi-Conserved			Excluded		
		110	011	101	001	100	010
U	0.027	0.097	0.331	0.060	2.853	1.093	2.712
Y	0.001	0.002	0.012	0.002	0.067	0.009	0.054

Table 5.1: The percentage of features as a function of conservation type for **U** and **Y** for words $\{w_1, w_2, w_3 | \, w_1 \rightarrow w_2 \wedge w_2 \rightarrow w_3\}$. The percentage of features that are zero in all three words is omitted.

Space	Conserved	Semi-Conserved		Excluded
		110	101	100
U	3.6	9.5	5.2	81.7
Y	9.2	15.3	12.5	63.0

Table 5.2: The percentage of feature weight with respect to w_1 as a function of conservation type for **U** and **Y** for words $\{w_1, w_2, w_3 | \, w_1 \rightarrow w_2 \wedge w_2 \rightarrow w_3\}$.

taxonomy, then 001-Features don't represent evidence to contradict the DIH, 010-Features and 100-Features are problematic for the DIH regardless of their position in the taxonomy. In total, Excluded features account for over 92% of non-zero **U** columns[2]. If we assert that Conserved > Semi-Conserved > Excluded, then there is an inverse relationship between degree of conservation and the percentage of features in the sample.

The problems with the DIH persist when adopting the Prototype View of features, in which feature weight is interpreted as importance or relevance, and which view, when applied to the DIH, leads to the prediction that the *feature weight* of a hyponym should be a subset of the *feature weight* of the hypernym. In Table 5.2, which uses the same vector space as above, the percentage of feature weight is with respect to w_1. Thus, an 010-Feature (a feature present only in w_2) would have a feature weight percentage of 0 with respect to w_1 and for this reason, those sorts of features are omitted from the table. With regard to problems in the DIH, the proportion of relevance that is excluded is clearly dominant (as indicated by the relative size of 100-Relevance) (Wilcoxon signed rank test, v = 1745221, p < .0001). Additionally, if Conserved > Semi-Conserved > Excluded, there is an inverse relationship between conservation type and feature weight.

5.2.2 Feature Conservation

One can also consider feature conservation by examining the features directly. This analysis is based on the intuition that we can measure the taxonomic span over

[2]The calculation for **U** is $\frac{2.853+1.093+2.712}{0.027+0.097+0.331+0.060+2.853+1.093+2.712} = 0.093$. This quantity for **Y** is even greater.

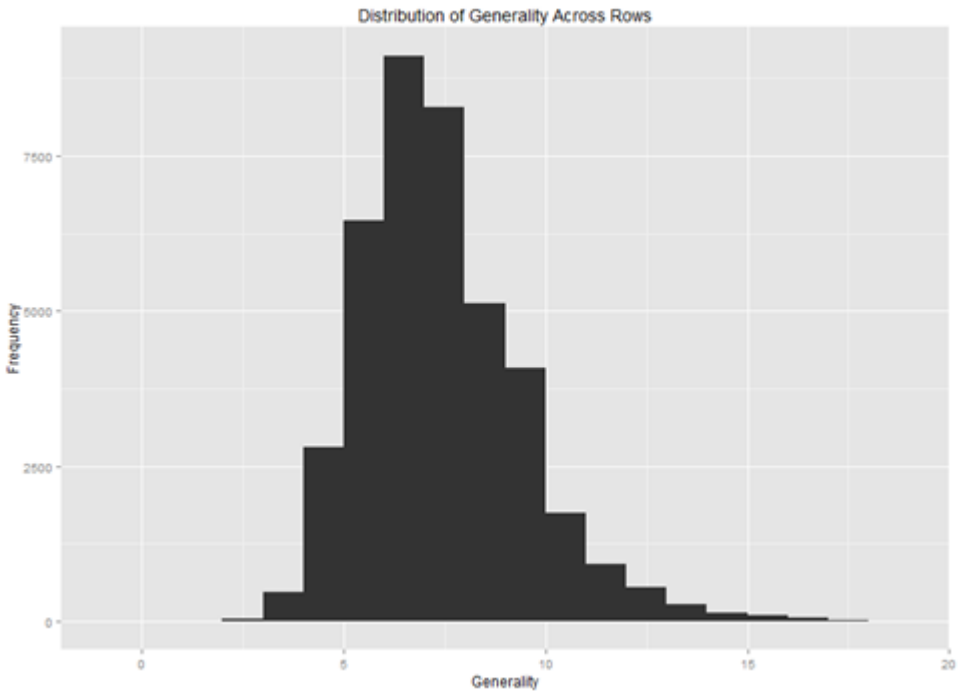

Figure 5.2: Histogram of the naive generality of all rows in **U** showing a high degree of lexicalization around 7 edges.

which a feature is used by looking at the naïve generalities[3] (NGs) (the number of edges between a noun and the root of the WordNet noun taxonomy) of the set of rows in which that feature is non-zero. A feature for which this set has high variance is, in a sense, more conserved than a feature for which this set has low variance. However, there is a bias that makes this analysis problematic: taxonomies are not lexicalized evenly. Instead, as Fig. 5.2 shows, the distribution of NG over rows is peaked around 7 edges from the root. So, the variance in NG of a particular feature's non-zero rows might exhibit lower variance because of the distribution of NG in the population, and not because the feature is used only within a small taxonomic span. Consequently, in order to test whether a given feature is conserved, I test whether its set of NGs is significantly different from the set of NGs for all rows.

The set of NGs for 1084 randomly selected columns were compared against the distribution of NGs for all rows. In 86% of columns sampled, the distribution of NGs of the column was significantly different than the NGs for the set of all rows (p-value $\leq$ .001, Welch Two Sample t-test). More intuitively, Fig. 5.3, which was generated from the 99,998 columns in **U**, shows that the vast majority of columns have a standard deviation less than 2.03, which is the standard deviation of NG from the set of all rows.

[3]See Sec. 3.4.

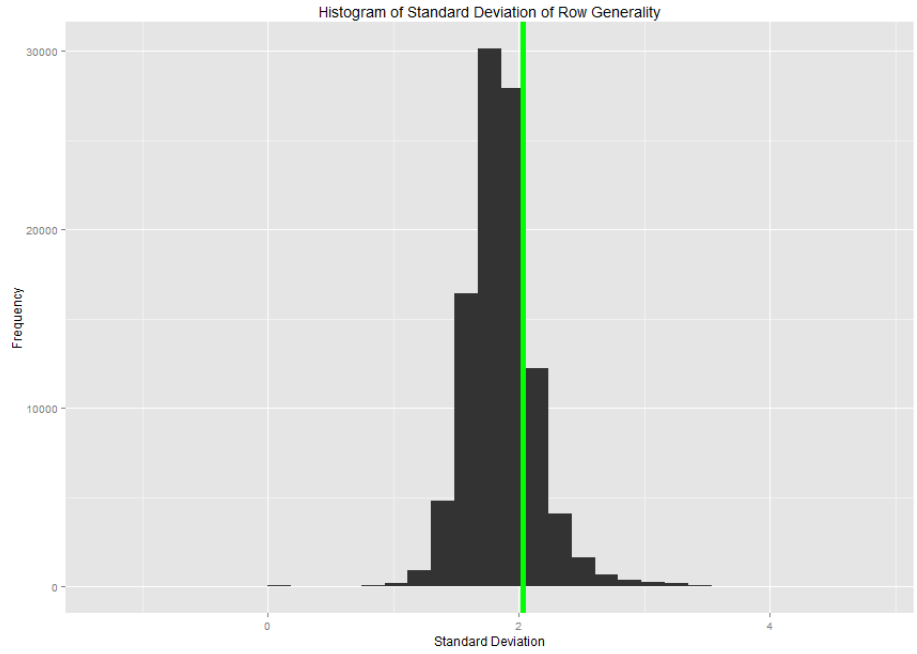

Figure 5.3: Histogram of standard deviations of NG of the non-zero rows for features in **U** showing that most features exhibit a lower standard deviation than the population from which they are drawn, here represented by a green line at $x = 2.03$.

It should be noted that this analysis is imperfect. Firstly, it relies heavily on NG, which is unreliable, both in the sense that its assumptions are demonstrably not true (not all edges can be treated as having the same semantic distance) and because one must ignore the fact that words are polysemous when computing their NG. The alternative to using polysemous rows isn't a viable option as monosemous words are concentrated at specific levels of the taxonomy.

5.3 Causes of Feature Exclusion

Having established the scale of the feature exclusion problem, we can examine some of its causes. There are three main causes of feature exclusion, one of which concerns the way in which human beings communicate, and the other two of which stem from how meaning is represented in DSMs.

5.3.1 Feature Exclusion and Human Communication

The broadest cause of Feature Exclusion has to do with the tendency for speakers and writers of natural language to adhere to Grice's [Grice, 1975] Cooperative Principle of conversation. Grice proposed four descriptive maxims for how people tend to communicate, two of which have an impact on feature inclusion: the Maxim of Quantity and the Maxim of Manner. The Maxim of Quantity states that utterances should be neither too specific nor too general. The Maxim of Manner posits that speakers should be sensitive to the capabilities and needs of their conversational partners, and should express themselves as briefly, clearly and in as organized a fashion as possible. Adherence to these maxims manifests in (1) a tendency to use words within a certain span of generality and (2) a tendency not to introduce superfluous arguments that attest to things that can be safely assumed.

Adherence to these maxims is easily demonstrable by example. Consider the following statements, *I went to the zoo and saw the entities* and *I went to the zoo and saw the Amazonian anaconda and the rattlesnake and the puff adder and the copperheads and the orangutan and the gorillas and the chimpanzees etc.* The former is too general, and tells little about the experience. The latter, which sounds like something a young child might say, tells everything about the experience, but assumes the reader wants a very granular response, or alternately, has the time to read such a response. Instead, we expect a response that is a compromise between informativeness and brevity, a la *I went to the zoo and saw the animals* or perhaps *I went to the zoo and saw the reptiles and primates.*

In a related case, consider this unlikely statement: *The tangible, mammalian, carnivorous, mobile, two-eyed dog wagged its tail.* With respect to canonical examples of dogs, the adjectives that precede the word *dog* are completely uninformative, as almost all such dogs are tangible mammals that eat meat, can move, and have two

eyes. However, what makes this case likely to lead to feature exclusion is that these properties are informative when applied to *animal*, a hypernym of *dog*.

Another cause of feature exclusion is the non-compositionality of some word pairs. Non-compositionality refers to the fact that the meaning of a pair of words is not the sum of its parts, but something else entirely. For example, the word-pair red apple is compositional: it can refer to an object that is both red and an apple. But, a red herring need neither be red nor a herring. These cases are also referred to as collocational, as in stiff drink. In a collocation, one element, the base, retains its normal connotation whereas the other element, the collocate, contributes a sense unlike its sense in other contexts [Evert, 2005]. Non-compositionality is problematic for DSMs as representations of lexical meaning generally, and is consequently indirectly problematic when using DSMs for HR. It is important to add the qualifier 'as representations of lexical meaning' because non-compositionality is not problematic when trying to identify multi-word expressions. Indeed, non-compositionality is crucial for such a task and red herring can be considered a multi-word expression. But for lexical semantic representations, the central supposition of feature vectors is that the relationship between a row and basis element is somehow illustrative of some property of the row element. This assumption holds for compositional pairs, but doesn't for non-compositional ones. And because a non-compositional feature is not illustrative of any property of a row element and is essentially unique, it is far less likely to be shared by semantically related words and is thus likely to be excluded.

In a similar case that is difficult to distinguish on the basis of association scores, sometimes row and basis elements combine compositionally but are nevertheless still highly associated with particular contexts, and thus are unlikely to be conserved. For example, lion can take the feature mane, but few other words can. In contrast, the broader term animal, a hypernym of lion, is unlikely to take the feature mane but can take the feature hair. I refer to this case as Compositional Non-Entailing because of the relationship between mane and hair, which is asymmetric: mane entails hair but hair does not entail mane. Feature exclusion may also occur when features are mutually entailing. For example, frog may take the feature gullet, and animal, a hypernym of frog, may take a synonymous feature esophagus. Unlike the earlier case, in which mane entailed hair but hair did not entail mane, these two features are mutually entailing and any occurrence of gullet with frog should be substitutable with esophagus without changing the truth conditions of the sentence. The problem is a really a probabilistic one: hypernyms are less likely to use the same feature as their hyponym, although this tendency isn't as strong nor as systematic as the Non-Entailing case. I refer to this case as Compositional Entailing.

Exclusion can also result from semantically equivalent features that are spelling variants of each other, e.g., colour and colour, which can also be considered as a special case of Compositional Entailing.

It is important to note that some of the cases of feature exclusion described above are

Type	Example
Non-Compositional	red herring
Compositional Non-Entailing	mane, hair
Compositional Entailing	gullet, esophagus
Spelling Variation	
Capitalization	colour, color
Case-Folding	
Gricean	*I went to the zoo and saw the entities.

Table 5.3: Types of Feature Exclusion

debatable, and determined by one's objective in modeling and one's interpretation of the DH. If the goal is to measure contextual similarity, then clearly collapsing features from different contexts is likely to have a deleterious effect. However, I argue that for the task of HR, these examples are worthy of correction.

5.3.2 Feature Exclusion and DSM Design

Two aspects of DSM design actually *contribute* to feature exclusion. The first of these is perhaps unavoidable, as it concerns the means by which we ensure the statistical robustness of the representation. Recall that the raw coocurrence matrix is weighted using an Association Measure that takes into account both the observed number of cooccurrences and the expected number, under a statistical model of independence. In fact, it is this very process that contributes to feature exclusion.

Consider a column in a typical raw cooccurrence matrix. The more conserved this column is, the higher its marginal frequency. The higher its marginal frequency, the greater the expected value for cells in that column. The greater the expected value, the greater the number of observations required for a positive association score. The greater the requirement, the less likely it is to be met, and thus the less likely those cells will become features in the sparse feature space.

Of course, this characteristic of typical feature spaces is actually by design. The traditional view of feature weight is based on the idea that the more statistically novel a feature is, the more discriminative power it should have for semantic similarity, and thus the greater the weight should be.

There are objective and subjective reasons for using discriminative power as the basis for feature weight, at least for applications that depend on semantic similarity. Objectively, models using discriminative power to weight features, which includes the status quo, have performed well on semantic similarity benchmarks. However, it's not clear the extent to which benchmark performance is necessary for high performance in HR, and it is unlikely that benchmark performance will be more important than minimizing feature exclusion, which is crucial to HR.

More subjectively, these feature weighting schemes seem to reward features that correspond with psychological salience, albeit imperfectly. For example, in U the verb bark is the third largest feature in the vector dog, which is consistent with an intuitive judgment that barking is important to the concept of dogness. Correspondence with intuitive judgments is often used as a crude qualitative benchmark of Association Measures, as in Fig. 2.5. However, there is reason to believe that the correspondence between feature weights and psychological salience is something that should be *avoided* rather than sought as the most salient features are rarely good candidates for conservation. Rather, salient features tend to be either special characteristics or characteristics that are typical of a near hypernym, meaning they will be conserved over a short span, but not a longer one. For example, the ordered list of 16 words most associated with eagle in the Edinburgh Associative Thesaurus [Kiss et al., 1976] consists of *bird*, *golden*, *nest*, *hawk*, *comic*, *fly*, *star*, *wings*, *America*, *eye*, *eyrie*, *mountain*, *soar*, *talons*, *air*, and *beak*. With the exception of *eye*, none of these are likely to be conserved across any great span, though *fly*, *wings*, *talons*, *bird*, and *beak* are characteristics of many birds of prey.

The second aspect of DSM design that contributes to feature exclusion stems from modeling meaning as space. In Cartesian space, we treat the dimensions as an orthornormal basis. But, not all spaces are treated this way. Frequently in machine learning, dimensions of the feature space are assumed to be interdependent. Nevertheless, in DSMs and semantic models, the basis elements of semantic spaces are treated as independent; operations on vectors typically only compare components with the matching component of the other vector.

Thinking of features as independent of each other means that two words can either share a feature or they don't; there is no *degree* of sharing, as there is no notion of similarity or dependence between features. But intuitively, if one word has the feature *claw* and another word the feature *talon*, these words are arguably far more comparable than two words that have no features in common. This treatment of feature similarity falsely treats as unrelated many features that are in fact related in known ways. As will become clear after the proposed procedures are described, this problem may be correctable.

6 Entailed Features

In Chapter 5, I established the scope and causes of the Feature Exclusion Problem (FEP). In this chapter, I present a proposal for mitigating the FEP by increasing feature conservation. I begin by establishing the goals for this form of representation before describing the proposal and finally the proposal's theoretical motivation.

6.1 A New Goal for Representation

When considering feature inclusion as a model used for Hypernymy Recognition (HR), the ideal feature space is very different from what is typically constructed in the status quo. In the ideal feature space, there is no feature exclusion whereas, as already demonstrated, in typical feature spaces, feature exclusion abounds. Because there would be no feature exclusion, there would be perfect nestedness with respect to features of word vectors and their hypernyms, as in Fig. 6.1. Furthermore, feature weight would follow a similar pattern, with far more significant portions conserved over greater spans of semantic distance.

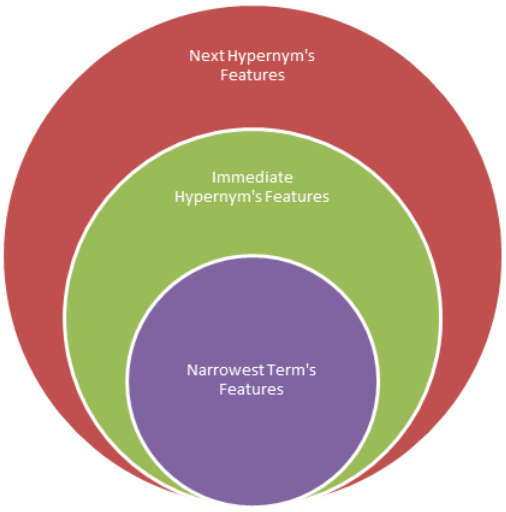

Figure 6.1: A Venn diagram depicting the nestedness between features in an ideal DIH-conforming feature space

6.2 Proposal

While the ideal feature space may not be achievable, there are ways by which a conventional feature space can be modified to make it exhibit more desirable characteristics. I describe in detail two different procedures in this section.

Structurally, Procedures 1 and 2 are identical: both construct mappings offline between features in the original space and lists of features in the modified space; both apply these maps by iterating over the vectors in the original space and modifying each vector independently. The essential difference is that whereas the mapping in Procedure 1 exploits synonymy information, the mapping in Procedure 2 exploits hypernymy.

Both procedures are also conservatively applied, in the sense that only a subset of features in the original space are modified. This subset of feature are those that are composed of a *monosemous* word [1]. Monosemous is operationalized as (1) being in WordNet and (2) having only one entry for that particular part-of-speech class[2]

Procedure 1 attempts to mitigate the FEP by dealing with cases like gullet/esophagus and colour/color in Table 5.3. The intuition here is that if $\vec{v_1}$ has feature *gullet* but not *esophagus* and $\vec{v_2}$ has the feature *esophagus* but not *gullet*, then $\vec{v_1}$ and $\vec{v_2}$ are more related than if neither of them had these features. Thus, the map that is constructed offline is between a monosemous feature f_1 and a list of monosemous features $\{f_2, ...f_n\}$ such that all features are collectively synonymous.

Procedure 2 attempts to mitigate the feature exclusion cases like mane/hair in Table 5.3, wherein one feature is a hyponym of the other or two features share a common hypernym. The intuition here is that, given that (1) feature f_1 is a hyponym of f_2, (2) that $\vec{v_1}$ has feature f_1 but not f_2, and (3) $\vec{v_2}$ has feature f_2 but not f_1, the vectors are more comparable for having these features than not having them. As with Procedure 1, a map of features to sets of features is created offline and then this map is used by a function that can be called on each feature vector in the space independently. However, in this case, instead of reflecting synonymy, the keys and values of the map meet the constraint that the key is a hyponym of all members of the set.

The application of each map to the feature space also differs with respect to how feature weight is composed. Procedure 1 uses the function putOrMax(), which sets all synonymous features to take the maximum feature weight of the features in the set of features in the map. This ensures that if a vector has any of the synonymous features, it effectively has all of them. Procedure 2 uses the function putOrAdd(),

[1] Features in **U** are composed of a word and part-of-speech tag. Features in **Y** are composed of a word, a part-of-speech tag and dependency relation and a direction.

[2] When searching in WordNet via the web, there is no way to control the part of speech of the query. When using a library to search WordNet locally, oftentimes the query is an object that includes a part-of-speech tag. In either case, the result of the query is a set of matching synsets in which that word occurs.

input : o, a map between features and association scores representing the original feature vector
k, a feature map $\{f_a \mapsto \{f_b, f_c, f_d, ...\}\}$, where $f_i \in Features$
output : n, a map between features and association scores representing the new vector
$n \longleftarrow \emptyset$
for *Entry* $e \in$ `EntrySet(`o`)` **do**
  if $e_{key} \in$ `KeySet(`k`)` **then**
    for $f \in$ `get(`k, e_{key}`)` **do**
      `putOrMax(`n, f, e_{value}`)`
    end
  else
    `putOrMax(`n, e_{key}, e_{value}`)`
  end
end
return n

Figure 6.2: Procedure for applying a feature map to a feature space using putOrMax(). putOrMax() can be replaced with the function putOrAdd() and used with Procedure 2's map without changing the algorithm. EntrySet(*map*) is equivalent to the function of the same name in Java, which returns a list of all key-value pairs in a map. KeySet(*map*) returns all keys in a map.

which causes feature weight to become proportional to semantic generality, such that the most general words take the most feature weight.

The iterative procedure is described in Fig. 6.2.

6.3 Theoretical Justification

The design of the procedures is justified on many levels. Firstly, the procedures can be seen as a means of constructing higher-order features. Unlike in truncated SVD, in which the resulting feature set is a set of latent features that were not part of the original feature set, the resulting space from the procedures is a concatenation of the original space with some number of higher-order features. These higher-order features arise naturally by considering entailed features that are not in the space. For example, the feature *dog* entails *organism*. If *organism* were not already a feature in the space but the feature *dog* was, *organism* would be added after applying the procedure. The appeal of higher-order features lies in their ability to capture indirect relationships: we no longer need to observe two words with the same feature in order to compare them, but instead need only observe them with similar features, as alluded to in Sec. 5.3.2. Failure to consider the similarity features reduces the the granularity of semantically distant comparisons, obscuring meaningful relations

between words. However, though the appropriateness of considering synonymous features and features that are immediate hypernyms, it is less clear that *distant* hypernyms, such as those at the very tops of taxonomies, need also be included.

Additionally, these procedures may be seen as a sparsity reduction measure. As described in Sec. 5.3.1, the FEP is a consequence of the way we communicate . The procedures here can be seen to simulate a feature space in which Gricean principles didn't constrain lexical choice and speakers used all permissible combinations of words.

The choice of the putOrMax() and putOrAdd() functions in Procedures 1 and 2 respectively are also theoretically motivated. In Procedure 1, the putOrMax() functions effectively designates synonymous features as equivalent. While this seems intuitively obvious, it actually contradicts the conventional theory about feature weight. Conventionally, feature vectors are compared on the basis of contextual similarity and, from this perspective, collapsing contexts that may be semantically equivalent means throwing away information. However, in this thesis, the object is a semantic taxonomy and so contextual differences are relatively less important.

In Procedure 2, the decision to use putOrAdd() is also theoretically motivated. The choice of the putOrAdd() operator treats feature weight as independent evidence that can be *combined*. Thus, there is greatest support for features for which there is the most evidence. Alternately, one could view the proportionality of feature weight and semantic generality as creating a universal semantic foundation.

6.4 Related Work

The idea to improve the semantic characteristics of a vector space through some sort of post-processing is not a new one. Indeed, Landauer and Dumais's [Landauer and Dumais, 1997] LSA can be seen as one example of this. However, unlike LSA, the procedures described here are task-oriented and use semantic knowledge, rather than mathematical principles, to guide the process.

Geffet and Dagan [2009] observed that the vectors in a typical cooccurrence matrix, the magnitude of feature weights often does not always correspond with how characteristic a feature is. Instead of rewarding features that are characteristic of a word, association measures reward statistically novel features, and while this is a reasonable proxy for characteristic-ness, it is remains imperfect. This observation lead Geffet and Dagan to implement a bootstrapping procedure that re-weighted features based on the sum of the similarity of neighboring vectors that also have that feature. These weights can then be used to reduce the size of the matrix simply by choosing the top N weighted features. Their bootstrapped vectors exhibited a number of positive characteristics. They were found to perform better on a lexical entailment task. When nearest neighbors were selected with a similarity measure,

the precision of the resulting set, with respect to entailment, increased by 50% relative to the baseline. Additionally, reduced bootstrapped vectors were shown to perform best when only the top 100 features were used. As in this approach, this bootstrapping is useful in that it can be applied to all vectors in the space. However, because it is based on proximity, the vectors most in need of bootstrapping (the narrowest vectors) may also be least impacted, as their nearest neighbors will likely be farther away than vectors are that less sparse.

Faruqui et al. [2015] implemented a post-processing procedure that performed an optimization that minimized changes to vectors while also minimizing their distance to semantically related neighbors. In experiments, they showed that retrofitted vectors constructed using a variety of different methods performed better on a variety of semantic similarity benchmarks and tasks. Unlike the present work, their retrofitting procedure can only be applied to vectors representing words in a lexical resource.

7 Exploratory Experiments

In this section, I present a number of exploratory experiments in order to understand the impact of applying the procedures from Chapter 6 on **U** and **Y**, the two feature spaces intended to represent the state-of-the-art[1]. This chapter does not explore the impact of the procedures on Hypernymy Recognition, which is the subject of Chapter 8.

Because the impact of Procedure 1 was minimal, I collapse the effects of both Procedure 1 and Procedure 2 into a single variable and I refer to spaces modified by these procedures with the m subscript. Thus, the modified feature spaces of **U** and **Y** are referred to as $\mathbf{U}_m$ and $\mathbf{Y}_m$. Additionally, I refer to both procedures from Chapter 6 collectively as *the procedure*.

7.1 Qualitative Impact

I begin with a qualitative analysis of feature ranks (by weight) for the vectors representing *dog* in both the **U** and **Y**. Because the procedure tends to make the largest features nouns (because nouns have the deepest taxonomies), it is more informative to separate features by part-of-speech. Furthermore, I relegate the analysis to features in the original space, and exclude those added. Finally, because the ranks of adjectives and adverbs change little (a consequence of the fact that entailment-based modification was only applied to noun and verb features), I exclude these from this analysis.

What is clear from the top-ranked features in Table 7.1 and Table 7.2 is that characteristics that are relatively unique to dogs, e.g., *musher*, are demoted while characteristics that are more general are promoted.

However, the sort of general features that are promoted in the feature spaces **U** and **Y** seem to differ. In the **U** feature space, with only two notable exceptions, *bird* and *ungulate*, the top noun features in the modified feature space $\mathbf{U}_m$ seem fairly close to the sorts of properties we would need to relate *dog* to other animals, but insufficient to relate to more semantically distant words like *hammer*. Many of the top-ranked verb features (*bark*, *yap*, *yelp*, and *wag*) in $\mathbf{U}_m$ are also top-ranked verb features from **U**. However, with the exception of *bark*, these features take lower ranks in $\mathbf{U}_m$. As with the top-ranked noun features, the top-ranked verb features include many

[1]For details on how these two DSMs differ, see Sec. 4.3.1.

Rank	Noun		Verb	
	$\mathbf{U}$	$\mathbf{U}_m$	$\mathbf{U}$	$\mathbf{U}_m$
1	sledding	animal	bark	bark
2	sniffer	dog	foul	breed
3	sled	mammal	yap	sterilize
4	musher	food	kennel	treat
5	turd	carnivore	wag	walk
6	rehome	waste	muzzle	feed
7	whelk	bird	sledge	frighten
8	leash	ungulate	salivate	foul
9	crossbreed	disease	rehomed	jiggle
10	cat	canine	rehoming	eat
11	Alsatian	hound	neuter	sleep
12	kennel	guardian	yelp	yelp
13	mongrel	primate	cross-breed	yap
14	poo	goody	snarl	kennel
15	prairie	guard	herd	wag

Table 7.1: The top ranked features by POS in both $\mathbf{U}$ and $\mathbf{U}_m$

that are suitable for comparison with other animals, e.g. *breed*, *feed*, and *sleep*, but none that would be useful for comparison with semantically distant words.

Analysis of the top ranked features in $\mathbf{Y}$ and $\mathbf{Y}_m$ shows a similar pattern, with the promotion of features that are useful for comparison with semantically similar words, e.g. *NMOD-R_beast* and *NMOD-R_animal*. In addition to these sorts of features, a substantial proportion of the top ranked noun features are far more general, e.g. *NMOD-R_matter* and *NMOD_act*, and are useful for comparisons with semantically more distant words. Many of the top ranked noun features in $\mathbf{Y}_m$ seem to relate to personhood, for example *NMOD-R_someone*, *NMOD_somebody*, *NMOD_person*, and *NMOD_someone*. This is may be a consequence of a subtle feature of dogness, namely that dogs are domesticated animals that have become important to humanity. Unlike the top ranked verb features in $\mathbf{U}_m$, the top ranked verb features in $\mathbf{Y}_m$ do not include any from the set of top ranked verb features in $\mathbf{Y}$. As in $\mathbf{U}_m$, many of the top ranked verb features, e.g. *NMOD-R_sleep*, *NMOD-R_excrete* and *NMOD-R_rest*, are suitable for comparison with other animals.

7.2 Quantitative Impact

In this section, I explore the effect of the procedure with methods that are better suited to quantification.

Rank	Noun		Verb	
	$\mathbf{Y}$	$\mathbf{Y}_m$	$\mathbf{Y}$	$\mathbf{Y}_m$
1	NM-R_sniffer	NM-R_someone	NM-R_bark	NM-R_sterilize
2	NM_sledding	NM-R_whole	NM-R_foul	OBJ_treat
3	NM-R_Mutt	NM-R_beast	SBJ_bark	NM-R_walk
4	NM_turd	NM-R_animal	SBJ_foul	OBJ_stroke
5	NM-R_Pogo	NM-R_creature	OBJ_skewer	NM-R_sleep
6	NM-R_sled	NM_somebody	OBJ_wag	SBJ_excrete
7	NM_faeces	NM_person	NM-R_sledge	NM-R_steal
8	NM_poo	NM_someone	OBJ_bark	OBJ_frighten
9	NM_bollocks	NM-R_dog	OBJ_rehoming	NM-R_treat
10	NM_excrement	NM-R_matter	NM-R_snarl	NM-R_arouse
11	NM_sled	NM_act	SBJ_neuter	SBJ_run
12	NM-R_beagle	NM-R_act	OBJ_pet	NM-R_excrete
13	NM_poop	NM_animal	SBJ_lick	NM-R_wound
14	NM-R_collie	NM_creature	NM-R_pant	NM-R_injure
15	NM_kennel	NM_beast	OBJ_neuter	NM-R_rest

Note: NMOD is here abbreviated NM, for formatting reasons.

Table 7.2: The top ranked features by POS in $\mathbf{Y}$ and $\mathbf{Y}_m$

	$\mathbf{U}$	$\mathbf{U}_m$	$\mathbf{Y}$	$\mathbf{Y}_m$
Number of Rows	99110	99110	61364	61364
Number of Features	99997	102129	99999	103747
Density	0.8%	1.1%	0.8%	1.0%

Table 7.3: Dimensions and density of feature spaces. Density is computed by dividing the number of non-zero entries by the total area of the matrices.

7.2.1 Density and Dimensions

Perhaps the simplest way to assess the effect of the procedures is to ignore the magnitude of feature weights and to consider the number of features in the space and the number of non-zero entries. As Table 7.3 shows, the modified feature spaces have about 2-4% more features and are about 70% denser than the original spaces.

7.2.2 Number of Non-Zero Entries

We can also examine what *sorts* of entries are non-zero by grouping different entries together. For this analysis, I use naïve generality[2] (NG) to group entries. Because each row and feature are associated with a word, we can calculate the NG of rows and features and use this number to assign for each entry an NG_{row} and NG_{feat}.

[2]See Sec. 3.4.

	Not in WordNet	$NG_{feat} \leq 7$	$NG_{feat} > 7$	Total
Not in WordNet	1.04	5.70	1.31	3.05
$NG_{row} > 7$	1.03	4.19	1.25	2.48
$NG_{row} \leq 7$	1.05	8.50	1.29	4.10
Total	1.04	6.60	1.28	3.41

Table 7.4: The multiplicative factor by which the number entries in **U** would need to multiplied to be equivalent to the number of entries in $\mathbf{U}_m$ for various combinations of NG rows and features.

The dependent variable in this analysis is the factor by which the number of entries of a certain sort in the original space would need to be multiplied to be equal to the number of entries of the same sort in the modified space. Because the procedure only adds entries, this factor is always greater than 1.

There are two strong trends in the data in Table 7.4, one of which is an obvious consequence of the procedures and the other less so. The marginal for entries with $NG_{feat} \leq 7$ (more semantically general features) is substantially greater than the marginal for entries with $NG_{feat} > 7$, indicating that the change in density in semantically more general features as a result of the procedures was substantially greater than the change in density in less general features. More surprising is that the marginal of $NG_{row} \leq 7$ (more semantically general rows) is also substantially greater than the marginal of $NG_{row} > 7$ rows. This indicates that the change in density in semantically more general rows as a result of the procedures was substantially greater than the change in density in less general rows. This may be caused by the fact that, as per [Weeds, 2003], semantically more general words tend to be observed in a larger number of contexts and thus, because those vectors are much bigger to begin with, and because the procedures increase the number of features at a rate proportional to the original vector's size, they increase more.

Table 7.5 shows f_{Y,Y_m}, and seems to exhibit the same patterns as Table 7.4, though not as strongly.

Together, tables Table 7.4 and Table 7.5 suggest that semantically more general columns are more likely to be conserved in modified spaces than unmodified ones, because they are far denser.

7.2.3 Feature Conservation

An analysis of feature conservation[3] provides additional evidence that the procedure mitigates feature exclusion with respect to both the number of conserved features as well as the amount of conserved feature weight. Table 7.6 presents the data from an analysis of feature exclusion by number of features, in which the percentage of

[3]Feature conservation was first described in Sec. 5.2.1.

	Not in WordNet	$NG_{feat} \leq 7$	$NG_{feat} > 7$	Total
Not in WordNet	2.08	2.09	1.34	1.68
$NG_{row} > 7$	1.40	1.65	1.25	1.42
$NG_{row} \leq 7$	2.41	2.73	1.50	2.03
Total	1.99	2.25	1.39	1.77

Table 7.5: The multiplicative factor by which the number entries in $\mathbf{Y}$ would need to multiplied to be equivalent to the number of entries in $\mathbf{Y}_m$ for various combinations of NG rows and features.

Space	Conserved	Semi-Conserved			Excluded		
		110	011	101	001	100	010
$\mathbf{U}$	0.03	0.10	0.33	0.06	2.85	1.09	2.71
$\mathbf{U}_m$	0.18	0.18	0.60	0.13	3.31	1.22	3.13
$\frac{\mathbf{U}_m}{\mathbf{U}}$	6.75	1.83	1.81	2.22	1.16	1.12	1.15
$\mathbf{Y}$	0.001	0.002	0.012	0.002	0.067	0.009	0.054
$\mathbf{Y}_m$	0.003	0.003	0.018	0.002	0.073	0.009	0.058
$\frac{\mathbf{Y}_m}{\mathbf{Y}}$	2.598	1.287	1.479	1.393	1.089	1.023	1.079

Table 7.6: Percent of features by degree of feature conservation.

features that are zero is excluded. Table 7.6 reveals that the percentage of features that are conserved increased by a multiplicative factor of 6.75 and 2.60 in $\mathbf{U}$ and $\mathbf{Y}$ respectively as a result of applying the procedures. The percentage of semi-conserved features (i.e., features shared by two of the three words in the triple, but not all three) increased by a multiplicative factor of 1.95 and 1.39 in $\mathbf{U}$ and $\mathbf{Y}$ respectively. In contrast, the percentage of excluded features increased by a multiplicative factor of 1.14 and 1.06 in $\mathbf{U}$ and $\mathbf{Y}$ respectively. Thus, the proportion of conserved and semi-conserved features relative to the percentage of excluded features increased, but this increase is mostly attributable to an overall increase in the number of non-zero features and excluded features still account for the vast majority of features.

With regard to feature weight conservation, the procedures increase the amount of feature weight in conserved, and to a lesser extent, semi-conserved features, while decreasing the proportion of feature weight that is excluded. Table 7.7 shows that the procedures increase the proportion of conserved feature weight by a multiplicative factor of 9.80 and 3.48 in $\mathbf{U}$ and $\mathbf{Y}$ respectively. There may be a small effect within the Semi-Conserved Features, such that the weight of features useful for more distant comparisons in increasing, as demonstrated by the difference between 1.34 and 1.05 and 1.05 and 0.92 for $\mathbf{U}$ and $\mathbf{Y}$ respectively. In contrast, the proportion of feature weight for excluded features is reduced significantly.

Space	Conserved	Semi-Conserved		Excluded
		110	101	100
$\mathbf{U}$	0.04	0.10	0.05	0.82
$\mathbf{U}_m$	0.35	0.10	0.07	0.48
$\frac{\mathbf{U}_m}{\mathbf{U}}$	9.80	1.05	1.34	0.58
$\mathbf{Y}$	0.09	0.15	0.12	0.63
$\mathbf{Y}_m$	0.32	0.14	0.13	0.41
$\frac{\mathbf{Y}_m}{\mathbf{Y}}$	3.48	0.92	1.05	0.65

Table 7.7: Proportion of feature weight as a function of degree of feature conservation.

NG	$\mathbf{U}$ vs. $\mathbf{U}_m$	$\mathbf{Y}$ vs. $\mathbf{Y}_m$
Not in WordNet	0.44	0.60
$NG_{row} \leq 7$	0.37	0.53
$NG_{row} > 7$	0.36	0.56
Total	**0.41**	**0.58**

Table 7.8: Aggregate cosine similarity for pairs of vectors representing the same word in both $\mathbf{U}$ and $\mathbf{Y}$

7.2.4 Semantic Similarity and Naïve Generality

As described earlier, cosine similarity is a convenient and empirically useful way of comparing semantic feature vectors. In contrast to most applications, where the objective is to compare vectors in the same space, in this analysis the object is to compare vectors in *different* spaces, the better to understand the impact of the procedures. Cosine similarity can be used for this purpose with one caveat: cosine similarity effectively treats the vectors, which in this case belong to features spaces with different feature sets, as belonging to a single feature space whose feature set is the union of the features of the vectors being compared. This depresses the similarity scores lower than if non-shared features were discarded.

Table 7.8 presents data from a comparison of vectors representing the same word in $\mathbf{U}$ and $\mathbf{U}_m$ and $\mathbf{Y}$ and $\mathbf{Y}_m$, respectively. The data suggest that, generally, the effect of the procedures is significant: the similarity between vectors representing the same word is low. Additionally, the procedure had a stronger effect on $\mathbf{U}$ than on $\mathbf{Y}$ and the effect on similarity seems independent of NG, as indicated by the relatively small difference between the $NG_{row} \leq 7$ rows and $NG_{row} > 7$ rows. However, words that are not in WordNet, and which consequently do not have a naïve generality, seem to be less strongly affected, as indicated by the greater similarity between the original and modified vectors.

Another important consideration for a semantic vector space is the degree to which it satisfies the geometric metaphor of meaning, which stipulates that distance in space

Space	Semantic Similarity (Pearson's ρ)				Synonym Identification (Correctness)	
	WS-353	MEN	RG	Simlex-999	ESL	TOEFL
$\mathbf{U}$	0.54	0.73	0.71	0.29	0.50	0.11
$\mathbf{U}_m$	0.27	0.57	0.53	0.21	0.29	0.06
$\mathbf{Y}$	0.47	0.70	0.70	0.32	0.54	0.10
$\mathbf{Y}_m$	0.12	0.43	0.34	0.13	0.17	0.04

Table 7.9: Pearson correlation coefficients and Correctness Percentage for Semantic Similarity and Synonym Detection tasks

should be equivalent to distance in meaning, as satisfying the geometric metaphor of meaning, is part of the more general claim that the space exhibits semantic characteristics. Data from a variety of standard semantic similarity and synonym detection benchmarks actually suggest that the procedures substantially *worsen* performance on these benchmarks. This result suggests that, whether or not the procedures improve performance on HR, they will likely only be useful as task-specific post-processing phase, and not for more general purposes.

Table 7.9 shows the results from a variety of semantic similarity benchmarks. The Semantic Similarity columns are all tests in which the similarity of vectors is compared with human judgments of semantic similarity and the overall score is the Pearson correlation coefficient. The Synonym Identification columns are tasks in which, for a candidate vector v, the closest vector, v_i, from a set of vectors V, is predicted to be the synonym of v. For a more in-depth review of these individual benchmarks, see [Baroni and Lenci, 2010] and [Hill et al., 2014].

One possible explanation for this result is that by conserving features and feature weight, the differences between vectors is decreased. And, indeed, whereas the mean cosine similarity of vectors in the WS-353 task for $\mathbf{U}$ and $\mathbf{Y}$ is .1, the mean cosine similarity for this same task for $\mathbf{U}_m$ and $\mathbf{Y}_m$ is .7.

8 Experiments

In this section, I review various Hypernymy Recognition (HR) datasets and then report the effect of the procedure described in Chapter 6 on performance. As in Chapter 7, I use $\mathbf{U}$ and $\mathbf{Y}$, the two feature spaces intended to represent the state-of-the-art[1] and refer to the spaces that have been modified as $\mathbf{U}_m$ and $\mathbf{Y}_m$.

8.1 Hypernymy Recognition Datsets

I use three different datasets in this analysis, each of which affords different advantages and drawbacks for this task. Each dataset entry set consists of a word pair $\langle x, y \rangle$ and class label, which in this case is binary, denoting either *x is hyponym of y* or *x is not a hyponym of y*.

8.1.1 Weeds Dataset

The Weeds et al. dataset [Weeds et al., 2014b] is considered the gold standard for this thesis, as its many good properties outweigh the drawbacks. The Weeds dataset consists of 2515 instances (input/output pairs), each consisting of a pair of words and a class value. All words in all instances are both frequently occurring and are likely to be monosemous[2]. It has an equal number of positive and negative instances. The authors controlled for semantic similarity, such that the average path distance in WordNet between pairs in the sets of positive and negative instances are comparable. Finally, the set of instances does not include any pairs that can be inferred from any other groups of pairs, thus preventing models from learning trivial and ungeneralizable facts.

One property that may be beneficial for supervised HR is that the average path distance between pairs *overall* is quite small, which means that, if it is the case that the training instances exist in a low-dimensional manifold, the instances fall near the decision boundary and there is a good chance that models trained with this dataset will generalize to pairs that are farther apart.

[1] For details on how these two DSMs differ, see Sec. 4.3.1.

[2] *Monosemous* means having only one sense.

8.1.2 BLESS Dataset

The BLESS dataset [Baroni and Lenci, 2011] was actually created to examine lexical semantic relations, which includes many relations in addition to hypernymy. As such, the dataset must be adapted in order to be used for HR. The relations included in the dataset include many standard relations for lexical inference and some that are relatively unique. *Hypernymy*, *coord*, which the authors define to mean words that share a semantically close hypernym, and *meronymy* are all commonplace in lexical inference. BLESS also includes a catch-all relation called *random*, which holds between pairs of words that have a common distant ancestor in the taxonomy. Finally, BLESS includes relations such as *attri* and *event* which have to do with selectional preferences of adjectives and verbs respectively. Because it was designed for learning so many lexical semantic relations, the number of hypernym instances in BLESS is much smaller than the number of non-hypernym instances.

In this work, I use a subset of BLESS created by Levy et al. [2015] which includes only instances applicable to lexical inference and which codes hypernymous pairs as positive instances and all other relations as negative instances. The Levy subset is not balanced. It consists of 14547 instances, 13210 of which are negative and 1337 of which are positive.

Using the Baroni and Lenci terminology, each instance in the dataset consists of a *concept*, *relation* and *relatum*, e.g. *concept is a kind of relatum*, which can represent an instance with the hypernymy relation. The concepts, of which there are 200, were selected to be frequently occurring, and neither ambiguous nor highly polysemous. The small number of concepts and much larger total number of instances means that each concept occurs in many instances, another characteristic that makes it different from the Weeds dataset. Critically, each concept belongs to one of 17 broader classes, of which the concepts can be either typical or atypical instances. The relata were selected from a variety of difference sources and as such, are representative of a broad swath of semantic information.

Although it's less of an issue for unsupervised models, which have no means of prioritizing different dimensions of the input space, the usage of a small number of concepts that belong to well-defined semantic groups proves critically detrimental to supervised models. As Roller et al. [2014] observed, linear supervised models trained using BLESS learn to weight features that are clearly not generalizable, but are instead related to the broader classes to which the concepts belong.

8.1.3 Entailment Dataset

The Entailment dataset [Baroni et al., 2012], referred to in [Baroni et al., 2012] as $N_1 \models N_2$, was constructed specifically for HR. It is balanced. There were no explicit checks against polysemy, though the authors removed both hypernyms with many hyponyms (which are typically very abstract words like *entity*) and checked all pairs

to ensure correctness. The positive instances were generated from WordNet. After generating the positive instances, the negative instances were generating by either reversing positive instances (33% of negative instances) or randomly selecting other words in the dataset, after checking to see that they do not constitute an entailing pair. The result is 2770 instances. This procedure

8.1.4 Challenges of Dataset Construction

The HR setting presents many challenges to constructing a dataset. Firstly, there is the issue of balance vs. breadth. If one considers a comprehensive taxonomy, the set of all positive instances is significantly smaller than the set of all negative instances. The set of all positive instances is the set of all pairs that can be generated by considering the path to the root from every node. Given that taxonomies typically display "bulges" of heavy lexicalization, we can approximate the total number of pairs produced by this method as $k \times N$, where k is the level at which the bulge occurs. (In WordNet, this bulge occurs at around 6 edges from the root of the noun taxonomy.) In constrast, the set of all negative instances is proportional to $N \times (N - k)$, which, given that $N >> k$, is essentially N^2.

To adopt a balanced approach means throwing away the vast majority of negative instances. To adopt a broad approach means having a dataset that is extremely unbalanced.

8.2 Experiments

8.2.1 Experiment 1

In Experiment 1, I assess the effect of the procedures using all three datasets, a variety of models of feature inclusion, and both $\mathbf{U}$ and $\mathbf{Y}$. Most models tested require a decision boundary, which is a free parameter. To determine the optimal value of these parameters for both the Weeds and Entailment datasets, I use 5-fold cross-validation, and set the parameter value by maximizing accuracy. In the case of the BLESS dataset, in order that the parameter not exploit the dataset's unbalancedness, I use the parameters from the Weeds dataset.

8.2.1.1 Results

Table 8.2, Table 8.1 and Table 8.3 present the results from the experiment from the Weeds, Entailment and BLESS datasets, respectively. Because both the Weeds

Model	**U**	$\mathbf{U}_m$	$\mathbf{U}_m^{th}$	**Y**	$\mathbf{Y}_m$	$\mathbf{Y}_m^{th}$
BalAPInc	0.78	0.64	0.65	0.71	0.50	0.50
WeedsDiff	0.67	0.69	0.69	0.70	0.72	0.72
Cosine	0.75	0.71	0.71	0.74	0.64	0.64
InvCL	0.73	0.65	0.66	0.82	0.79	0.80
$\mathrm{P}_{\mathrm{Set}}$	0.77	0.77	0.77	0.82	0.80	0.80
$\mathrm{R}_{\mathrm{Set}}$	0.65	0.54	0.56	0.50	0.50	0.50
SingleWidth	0.64	0.64	0.63	0.68	0.68	0.68
$\mathrm{P}_{\mathrm{Weeds}}$	0.78	0.76	0.75	0.82	0.80	0.80
$\mathrm{R}_{\mathrm{Weeds}}$	0.67	0.57	0.59	0.50	0.50	0.50
WidthDiff	0.67	0.67	0.67	0.71	0.71	0.71

Table 8.1: Accuracy of models for Experiments 1 and 2 using the Entailment dataset

Model	**U**	$\mathbf{U}_m$	$\mathbf{U}_m^{th}$	**Y**	$\mathbf{Y}_m$	$\mathbf{Y}_m^{th}$
BalAPInc	0.58	0.49	0.49	0.51	0.49	0.50
WeedsDiff	0.68	0.68	0.68	0.70	0.70	0.70
Cosine	0.54	0.52	0.52	0.55	0.51	0.51
InvCL	0.58	0.61	0.61	0.66	0.69	0.69
$\mathrm{P}_{\mathrm{Set}}$	0.62	0.66	0.66	0.70	0.70	0.70
$\mathrm{R}_{\mathrm{Set}}$	0.49	0.49	0.49	0.49	0.49	0.49
SingleWidth	0.66	0.66	0.66	0.66	0.66	0.66
$\mathrm{P}_{\mathrm{Weeds}}$	0.63	0.63	0.62	0.69	0.68	0.68
$\mathrm{R}_{\mathrm{Weeds}}$	0.49	0.49	0.49	0.49	0.49	0.49
WidthDiff	0.67	0.67	0.67	0.70	0.69	0.69

Table 8.2: Accuracy of models for Experiments 1 and 2 using the Weeds dataset

dataset and the Entailment dataset are balanced, and because BLESS is not, Table 8.2 and Table 8.1 present accuracy percentages and Table 8.3 presents F_1 scores[3].

Broadly, the experimental procedures seem to have little to no effect; the marginals for each space in all three tables suggest that accuracy and F_1 scores decrease by a small percentage or remain the same, and this seems to hold true for all individual models as well.

However, a closer examination of the Weeds results reveals that the procedures did have an effect on performance. Both precision models, P_{Set} and P_{Weeds}, one or the other of which is an essential component in all other models, seemed to be effected dramatically by the procedures. Table 8.4 shows that the procedures, rather than boosting precision for positive instances, seem to have boosted precision for *all* instances. In the case of P_{Set}, the difference in precision between positive and

[3] F_1 is the harmonic mean of precision and recall, which in this case refers to instances classified and not shared and unshared features.

Model	**U**	$\mathbf{U}_m$	$\mathbf{U}_m^{th}$	**Y**	$\mathbf{Y}_m$	$\mathbf{Y}_m^{th}$
BalAPInc	0.22	0.02	0.03	0.21	0.00	0.00
WeedsDiff	0.18	0.18	0.18	0.19	0.20	0.20
Cosine	0.22	0.22	0.22	0.23	0.20	0.20
InvCL	0.20	0.16	0.16	0.23	0.21	0.21
$\mathrm{P}_{\mathrm{Set}}$	0.21	0.21	0.20	0.23	0.22	0.22
SingleWidth	0.19	0.19	0.19	0.21	0.21	0.21
$\mathrm{P}_{\mathrm{Weeds}}$	0.21	0.19	0.19	0.23	0.21	0.21
WidthDiff	0.18	0.18	0.18	0.19	0.19	0.19

Table 8.3: F_1 score for Experiments 1 and 2 using the BLESS dataset. The R_{Set} and R_{Weeds} models are omitted because the models classify all instances as negative and thus have an F_1 score of 0.

Model	Instance Type	**U**	$\mathbf{U}_m$	$\mathbf{U}_m^{th}$	**Y**	$\mathbf{Y}_m$	$\mathbf{Y}_m^{th}$
P_{Set}	0	0.10	0.18	0.16	0.15	0.19	0.18
	1	0.16	0.27	0.24	0.27	0.35	0.33
P_{Weeds}	0	0.11	0.39	0.36	0.12	0.31	0.29
	1	0.18	0.46	0.43	0.25	0.44	0.42

Table 8.4: The output of P_{Set} and P_{Weeds} in all spaces, aggregated by instance class

negative instances seems to have increased slightly as a result of the procedures. In the case of P_{Weeds}, the difference in precision between positive and negative instances didn't change at all.

This same effect is also observed when grouping instances by the naïve generality[4] (NG) of the input pair. Fig. 8.1 and Fig. 8.2 present the output of the P_{Set} and P_{Weeds} as a function of the NG of both input words. While this means of grouping instances is imperfect due to polysemy, it is still broadly accurate. A high-performing model should exhibit striation along lines with slope of 1, as instances in which the NG of the first word is less than the second cannot possibly be positive. While all of the models *do* seem to exhibit such a characteristic, critically, the modified spaces do not seem to exhibit it to a greater degree, as evidenced by the fact that the plots of the modified spaces are merely lightened versions of the originals.

8.2.2 Experiment 2

The results of Experiment 1 suggest that the procedures boost precision in all instances, rather than just in positive instances. One explanation for this result is that entailed features are not useful for HR. An alternate explanation that I test is whether a subset of all possible entailed features is useful for HR. While testing all

[4]See Sec. 3.4.

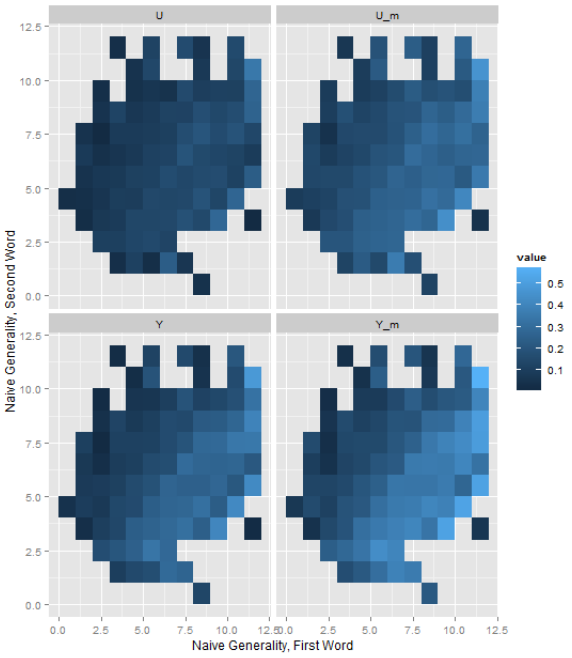

Figure 8.1: Heatmaps for all spaces of the output of the P_{Set} model for the Weeds dataset

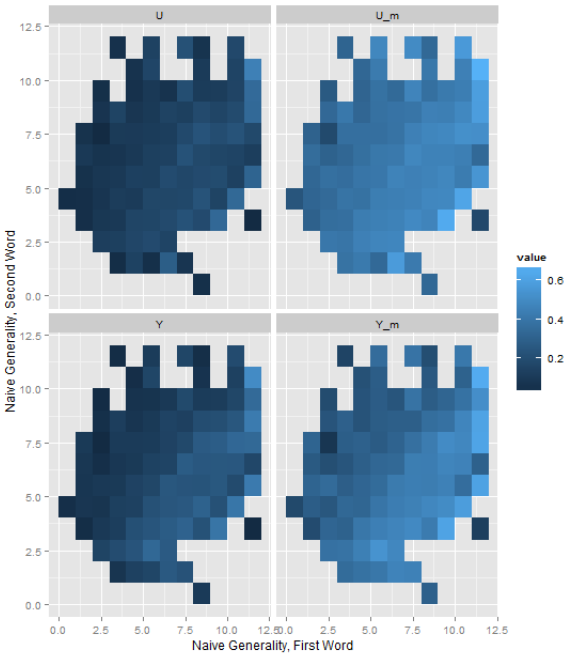

Figure 8.2: Heatmaps for all spaces of the output of the P_{Weeds} model for the Weeds dataset

	$\mathbf{U}$	$\mathbf{U}_m$	$\mathbf{U}_m^{th}$	$\mathbf{Y}$	$\mathbf{Y}_m$	$\mathbf{Y}_m^{th}$
$\mathbf{U}$	1.00	0.80	0.81	0.83	0.79	0.79
$\mathbf{U}_m$		1.00	0.94	0.76	0.82	0.82
$\mathbf{U}_m^{th}$			1.00	0.75	0.81	0.81
$\mathbf{Y}$				1.00	0.88	0.88
$\mathbf{Y}_m$					1.00	0.97
$\mathbf{Y}_m^{th}$						1.00

Figure 8.3: The Pearson correlations in the decisions of the $WeedsDiff$ model applied to the spaces

	Disagree	Agree	Total
Close	184	1599	1783
Far	30	372	402
Total	214	1971	2185

	Disagree	Agree	Total
Negative	85	994	1079
Positive	129	977	1106
Total	214	1971	2185

Table 8.5: A comparison of the decisions of $WeedsDiff$ trained on U and U_m as a function of the absolute difference in NG of the instance and whether the instance was a positive or negative.

subsets is impossible, Experiment 2 tests whether the applying the procedures to only the top half of features, as determined by feature weight, proves more effective than applying the procedures to all features.

The procedure is identical to Experiment 1, except that the algorithms are modified to only consider the top half of features, by weight.

8.2.2.1 Results

The results from Experiment 2, which are in Table 8.2, Table 8.1 and Table 8.3, are essentially indistinguishable from Experiment 1. Fig. 8.3 presents an analysis of the decisions of the $WeedsDiff$ model in each space, applied to each instance, and shows there there is a high degree of correlation between the model trained on U_m and on U_m^{th} ($\rho = .94$) and between Y_m and Y_m^{th} ($\rho = .97$), suggesting that despite applying the procedures to only the top half of features, the result is nearly identical with respect to HR. Table 8.5 presents frequency of agreement between U and U_m using the $WeedsDiff$ model as a function of the ADNG of the pair and whether or not it was positive. Chi-squared tests reveal that there is a significant effect of positivity ($\chi^2 = 8.4$, $df = 1$, $p = 0.004$) but that there was no significant effect of ADNG ($\chi^2 = 2.7$, $df = 1$, $p = 0.10$), suggesting that the effects of the procedure were greater for positive instances than for negative instances.

9 Conclusion

To conclude, I will revisit the motivation for this work, the intuition that guided the design of the procedures and the reasons for their ineffectiveness before reviewing options for future work.

9.1 Summary

The ability to identify positive instances of hypernymy is extremely important for many tasks. While there has been some success at identifying positive instances that are explicitly referred to in text using shallow lexico-syntactic patterns, the most common approaches to HR using more scalable methods capable of identifying all sorts of instances, which are based on using a DSM to construct representations of word meaning, do not perform significantly better than baseline models; mostly, these models assume that the representations of the hyponyms and hypernyms will share many features and yet in most cases, the feature sets in these words' representations are almost mutually exclusive. I refer to this problem as the Feature Exclusion Problem (FEP).

The scale of the FEP problem is part of what makes it such an issue. Due to the small percentage of features that are typically shared between the vectors representing most hyponyms and hypernyms, and the fact that most models consider only (1) shared features and (2) excluded features in the hypernym's representation, most features in the hyponym's representation do not even participate, effectively throwing away information.

Feature exclusion is the consequence of many factors. It is partially the consequence of how we communicate, which leads us to choose a small subset of all permissible combinations of words, and which in turn limits the sorts of cooccurrences we observe. Surprisingly, the FEP is also partially a consequence of how we compose the feature vectors themselves; in the attempt to identify statistically robust cooccurrences, we also ensure that feature vectors become very sparse and that no features will be maintained across a taxonomic span and thus feature conservation will be low. Furthermore, though the vast majority of models treat feature weight as indicative of importance, there is evidence that models that ignore feature weight perform just as well, suggesting that, at least in HR, some conventional wisdom about DSM design may not apply or may be altogether wrong.

Finally, feature exclusion is a consequence in how we interpret and compare feature vectors. By treating the basis of a semantic space as orthonormal, which is an implicit when the components of vectors are only compared against the matching components in other vectors, we ignore dependencies between the dimensions of the space and thus treat similar features as unique and dissimilar. We do this despite the fact that the very pursuit of lexical relations (and the fact that most dimensions are associated with words) indicates that many dimensions are related to each other.

The procedure described here attempted to mitigate each of these problems. The procedure can be viewed as a way to simulate patterns of language usage if humans chose from the set of all permissible and coherent word-combinations, as opposed to applying Gricean filters. The procedure, which dramatically changed the feature weight of nouns and verbs, can be seen as redefining the semantics of feature weight, from importance to each vector's meaning to something to importance for comparisons with the set of all vectors in the space. Finally, the procedure can be viewed as a way of treating features that are synonymous or which share a common hypernym as similar and therefore comparable.

With respect to Hypernymy Recognition, the hope was that these procedures would boost precision in positive instances of hypernymy more than negative instances. However, this hypothesis was not borne out in the data.

Instead, Experiment 1 demonstrated that precision was boosted for all instances in the datasets and performance on HR remained the same or worsened very slightly for almost all models. Experiment 2 showed that applying the procedures to a subset that, based on feature weight, were possibly more important to each individual vector did not improve the quality of the resulting models either.

9.2 Future Work

This work raises a number of important questions and highlights some outstanding older ones. The Feature Exclusion Problem remains a serious issue for DSMs and HR. One fundamental question is whether to allow for semantic knowledge to influence how different features are compared, such that feature similarity becomes more nuanced. Given how the procedures described here boosted precision for all instances, another question concerns refinements to the method explored here. One such refinement would change the way feature weights accumulate. The proportionality of feature weight and semantic generality may have been the culprit, at least in the case of models that considered feature weight, by adding new features that were shared by all vectors and were also large in magnitude. One variation worth exploring would be to make feature weight and semantic generality *inversely* proportional.

Finally, unsupervised HR is made far more challenging than it needs to be because it provides no clear way in which to introduce semantic knowledge. If, instead of

attempting to classify instances as positive or negative, the task were to determine the most likely location for a word in an *existing* taxonomy, models could still be used to infer whether a pair of words were a positive instance of hypernymy or not and incorporating known information would be simplified. Furthermore, the model's uncertainty would be more useful: models might express uncertainty over where in the taxonomy an extremely sparse feature vector might fit. Similarly, terms that have multiple hypernyms, such as *sportswear* and other restricted-perspective terms (see Chapter 3), might be shown to also have multiple maximum-likelihood locations in the taxonomy.

Bibliography

Marco Baroni and Alessandro Lenci. Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics*, 36(4):673–721, December 2010. ISSN 0891-2017. doi: 10.1162/coli_a_00016. URL `http://www.mitpressjournals.org/doi/abs/10.1162/coli_a_00016`.

Marco Baroni and Alessandro Lenci. How we BLESSed distributional semantic evaluation. *... GEometrical Models of Natural Language Semantics*, 2011. URL `http://dl.acm.org/citation.cfm?id=2140491`.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics, 2012.

Elia Bruni and Daniel Gatica-perez. Multimodal distributional semantics Marco Baroni , Thesis Advisor. 48(December), 2013.

John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: a computational study. *Behavior research methods*, 39(3):510–526, 2007.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, 1990. ISSN 08912017. URL `http://portal.acm.org/citation.cfm?id=89095&dl=`.

Daoud Clarke. Context-theoretic Semantics for Natural Language : an Overview. (March):112–119, 2009.

DA Cruse. Meaning in language: An introduction to semantics and pragmatics. 2004. URL `http://philpapers.org/rec/CRUMIL`.

M Davies. The corpus of contemporary american english (coca): 400+ million words, 1990-present, 2008. URL `http://corpus.byu.edu/coca/`.

Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.

S Deerwester, S Dumais, T Landauer, G Furnas, and R Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Katrin Erk. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Language and Linguistics Compass*, 6(10):635–653, October 2012. ISSN 1749818X. doi: 10.1002/lnco.362. URL http://doi.wiley.com/10.1002/lnco.362.

Stefan Evert. The Statistics of Word Cooccurrences Word Pairs and Collocations. (August 2004), 2005.

Stefan Evert. Corpora and collocations. *Corpus Linguistics. An International Handbook*, pages 1–53, 2008.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah a. Smith. Retrofitting Word Vectors to Semantic Lexicons. *Proceedings of NAACL*, (i), 2015.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and Evaluating {ukWaC}, a Very Large Web-Derived Corpus of {English}. *Proceedings of the 4th {Web as Corpus} Workshop*, pages 47–54, 2008.

J.R. Firth. A synopsis of linguistic theory 1930-55. *The Philological Society*, 1952-59, 1957.

Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, pages 107–114, 2005. doi: 10.3115/1219840.1219854. URL http://portal.acm.org/citation.cfm?doid=1219840.1219854.

Maayan Geffet and Ido Dagan. Bootstrapping Distributional Feature Vector Quality. (November 2008), 2009. ISSN 0891-2017. doi: 10.1162/coli.08-032-R1-06-96. URL http://eprints.pascal-network.org/archive/00006704/.

Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics*, chapter 3: Speech, pages 41–58. Academic Press, 1975.

Zellig S. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

Marti a. Hearst. Automatic Acquisition of Hyponyms ftom Large Text Corpora. *Proceedings of the 14th conference on Computational Linguistics*, 2:23–28, 1992.

Felix Hill, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. August 2014. URL http://arxiv.org/abs/1408.3456.

G. R. Kiss, Christine Armstrong, and Robert Milroy. *An associative thesaurus of English*. Medical Research Council, Speech and Communication Unit, University of Edinburgh, 1976.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(04):359–389, October 2010. ISSN 1351-3249. doi: 10.1017/S1351324910000124. URL http://www.journals.cambridge.org/abstract_S1351324910000124.

George Lakoff and Mark Johnson. Metaphors We Live By. In Jodi O'Brien and Peter Kollock, editors, *The production of reality: essays and reading on social interaction*, chapter 12, pages 124–134. 1997.

George Lakoff and Mark Johnson. *Philosophy in the Flesh: The Embodied Mind and Its challenges to Western Thought*. Basic books, 1999. ISBN 0465056733 9780465056736 0465056741 9780465056743.

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., 1997. ISSN 0033-295X.

Gabriella Lapesa and S Evert. A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions and Model Selection. *Transactions of the Association for ...*, 2:531–545, 2014. URL `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/457`.

Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. 32, May 2014. URL `http://arxiv.org/abs/1405.4053`.

A Lenci and G Benotto. Identifying hypernyms in distributional semantic spaces. *... on Lexical and Computational Semantics-Volume 1: ...*, pages 75–79, 2012. URL `http://dl.acm.org/citation.cfm?id=2387650`.

Alessandro Lenci. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31, 2008. ISSN 11202726.

Omer Levy and Yoav Goldberg. 2014 - Dependency-Based Word Embeddings.pdf. *Proceedings of the 52nd Annual Meeting of the ...*, 2014. URL `http://www.cs.bgu.ac.il/~yoavg/publications/acl2014syntemb.pdf`.

Omer Levy, Steffen Remus, and Chris Biemann. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? *NAACL 2015*, 2015.

Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304, 1998.

Will Lowe. Towards a theory of semantic space. *Proceedings of the Cognitive Science Society*, 2001. ISSN 0305-0009. doi: 10.1017/S0305000900006309.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, pages 1–12, 2013. URL `http://arxiv.org/abs/1301.3781`.

GA Miller and Richard Beckwith. Introduction to wordnet: An on-line lexical database*. *International journal ...*, (August), 1990. URL `http://ijl.oxfordjournals.org/content/3/4/235.short`.

Gregory Leo Murphy. *The big book of concepts*. MIT press, 2002.

Charles E Osgood. Psychological Bulletin. 49(3), 1952.

Sebastian Padó and Mirella Lapata. Dependency-Based Construction of Semantic Space Models. (December 2004), 2007.

Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *arXiv preprint arXiv:1105.5444*, 11:95–130, 2011. URL http://arxiv.org/abs/1105.5444.

Stephen Roller, Katrin Erk, and Gemma Boleda. Inclusive yet selective: Supervised distributional hypernymy detection. *Proceedings of the Twenty Fifth ...*, pages 1025–1036, 2014. URL http://www.cs.utexas.edu/users/ml/papers/roller.coling14.pdf.

Eleanor H. Rosch. Natural categories, 1973.

Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy, 1965. ISSN 00010782.

Magnus Sahlgren. *The Word-Space Model Using distributional analysis to represent syntagmatic and paradigmatic relations between words*. 2006. ISBN 9171552812.

G Salton and M J McGill. Introduction to modern information retrieval. *Introduction to modern information retrieval*, 1983. ISSN 0070544840.

Enrico Santus, A Lenci, Qin Lu, and SS im Walde. Chasing Hypernyms in Vector Spaces with Entropy. *EACL 2014*, pages 38–42, 2014. URL http://www.aclweb.org/anthology/E14-4#page=58.

H. Schütze. Dimensions of meaning. *Proceedings Supercomputing '92*, 1992. doi: 10.1109/SUPERC.1992.236684.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, pages 801–808, 2006. doi: 10.3115/1220175.1220276. URL http://portal.acm.org/citation.cfm?doid=1220175.1220276.

Idan Szpektor and Ido Dagan. Learning Entailment Rules for Unary Templates. 2008. doi: 10.3115/1599081.1599188. URL http://eprints.pascal-network.org/archive/00004483/.

PD Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010. URL http://www.aaai.org/Papers/JAIR/Vol37/JAIR-3705.pdf.

Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977. ISSN 0033-295X. doi: 10.1037//0033-295X.84.4.327. URL http://content.apa.org/journals/rev/84/4/327.

Tony Veale and Yanfen Hao. Acquiring Naturalistic Concept Descriptions from the Web. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 1121–1124, 2008.

Ellen M. Voorhees and Donna Harman. Overview of the Seventh Text REtrieval Conference (TREC-7). In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 1—-24, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.4400.

D Waltz and J Pollack. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9(1):51–74, March 1985. ISSN 03640213. doi: 10.1016/S0364-0213(85)80009-4. URL http://doi.wiley.com/10.1016/S0364-0213(85)80009-4.

JE Weeds. Measures and applications of lexical distributional similarity. (September), 2003. URL http://www.sequenceserial.com/Users/juliewe/weedsthesis.pdf.

Julie Weeds and David Weir. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, (May 2004), 2005. URL http://www.mitpressjournals.org/doi/abs/10.1162/089120105775299122.

Julie Weeds, D Clarke, and J Reffin. Learning to distinguish hypernyms and co-hyponyms. *Proceedings of the ...*, 2014a. URL http://musicdoc.org.uk/Users/juliewe/semanticrelations.pdf.

Julie Weeds, David Weir, and Jeremy Reffin. Distributional Composition using Higher-Order Dependency Vectors. *Proceedings of the 2nd Workshop on ...*, 2014b. URL http://www.aclweb.org/anthology/W/W14/W14-15.pdf#page=21.

Ludwig Wittgenstein. *Philosophical Investigations*, volume 2 of *G. ˜E. ˜M. ˜Anscombe (trans.)*. Blackwell, 1972. ISBN 0631146709.