Co-occurrence Retrieval: A Flexible Framework for Lexical Distributional Similarity

Julie Weeds and David Weir*
University of Sussex

Techniques that exploit knowledge of distributional similarity between words have been proposed in many areas of Natural Language Processing. For example, in language modeling, the sparse data problem can be alleviated by estimating the probabilities of unseen co-occurrences of events from the probabilities of seen co-occurrences of similar events. In other applications, distributional similarity is taken to be an approximation to semantic similarity. However, due to the wide range of potential applications and the lack of a strict definition of the concept of distributional similarity, many methods of calculating distributional similarity have been proposed or adopted.

In this work, a flexible, parameterized framework for calculating distributional similarity is proposed. Within this framework, the problem of finding distributionally similar words is cast as one of co-occurrence retrieval (CR) for which precision and recall can be measured by analogy with the way they are measured in document retrieval. As will be shown, a number of popular existing measures of distributional similarity are simulated with parameter settings within the CR framework. In this article, the CR framework is then used to systematically investigate three fundamental questions concerning distributional similarity. First, is the relationship of lexical similarity necessarily symmetric, or are there advantages to be gained from considering it as an asymmetric relationship? Second, are some co-occurrences inherently more salient than others in the calculation of distributional similarity? Third, is it necessary to consider the difference in the extent to which each word occurs in each co-occurrence type?

Two application-based tasks are used for evaluation: automatic thesaurus generation and pseudo-disambiguation. It is possible to achieve significantly better results on both these tasks by varying the parameters within the CR framework rather than using other existing distributional similarity measures; it will also be shown that any single unparameterized measure is unlikely to be able to do better on both tasks. This is due to an inherent asymmetry in lexical substitutability and therefore also in lexical distributional similarity.

1. Introduction

Over recent years, approaches to a broad range of natural language processing (NLP) applications have been proposed that require knowledge about the *similarity* of words. The application areas in which these approaches have been proposed range from speech recognition and parse selection to information retrieval (IR) and natural language

* Department of Informatics, University of Sussex, Falmer, Brighton, BN1 9QH, UK.

Submission received: 4 May 2004; revised submission received: 16 November 2004; accepted for publication: 16 April 2005.

generation. For example, language models that incorporate substantial lexical knowledge play a key role in many statistical NLP techniques (e.g., in speech recognition and probabilistic parse selection). However, they are difficult to acquire, since many plausible combinations of events are not seen in corpus data. Brown et al. (1992) report that one can expect 14.7% of the word triples in any new English text to be unseen in a training corpus of 366 million English words. In our own experiments with grammatical relation data extracted by a Robust Accurate Statistical Parser (RASP) (Briscoe and Carroll 1995; Carroll and Briscoe 1996) from the British National Corpus (BNC), we found that 14% of noun-verb direct-object co-occurrence tokens and 49% of noun-verb direct-object co-occurrence types in one half of the data set were not seen in the other half. A statistical technique using a language model that assigns a zero probability to these previously unseen events will rule the correct parse or interpretation of the utterance impossible.

Similarity-based smoothing (Hindle 1990; Brown et al. 1992; Dagan, Marcus, and Markovitch 1993; Pereira, Tishby, and Lee 1993; Dagan, Lee, and Pereira 1999) provides an intuitively appealing approach to language modeling. In order to estimate the probability of an unseen co-occurrence of events, estimates based on seen occurrences of similar events can be combined. For example, in a speech recognition task, we might predict that *cat* is a more likely subject of *growl* than the word *cap*, even though neither co-occurrence has been seen before, based on the fact that *cat* is "similar" to words that do occur as the subject of *growl* (e.g., *dog* and *tiger*), whereas *cap* is not.

However, what is meant when we say that *cat* is "similar" to *dog*? Are we referring to their semantic similarity, e.g., the components of meaning they share by virtue of both being carnivorous four-legged mammals? Or are we referring to their distributional similarity, e.g., in keeping with the Firthian tradition,[1] the fact that these words tend to occur as the arguments of the same verbs (e.g., *eat*, *feed*, *sleep*) and tend to be modified by the same adjectives (e.g., *hungry* and *playful*).

In some applications, the knowledge required is clearly semantic. In IR, documents might be usefully retrieved that use synonymous terms or terms subsuming those specified in a user's query (Xu and Croft 1996). In natural language generation (including text simplification), possible words for a concept should be similar in meaning rather than just in syntactic or distributional behavior. In these application areas, distributional similarity can be taken to be an approximation to semantic similarity. The underlying idea is based largely on the central claim of the distributional hypothesis (Harris 1968), that is:

> The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.

This hypothesized relationship between distributional similarity and semantic similarity has given rise to a large body of work on automatic thesaurus generation (Hindle 1990; Grefenstette 1994; Lin 1998a; Curran and Moens 2002; Kilgarriff 2003). There are inherent problems in evaluating automatic thesaurus extraction techniques, and much research assumes a gold standard that does not exist (see Kilgarriff [2003] and Weeds [2003] for more discussion of this). A further problem for distributional similarity methods for automatic thesaurus generation is that they do not offer any obvious way to distinguish between linguistic relations such as synonymy, antonymy, and hyponymy (see Caraballo [1999] and Lin et al. [2003] for work on this). Thus, one may question

1 "You shall know a word by the company it keeps."(Firth 1957)

the benefit of automatically generating a thesaurus if one has access to large-scale manually constructed thesauri (e.g., WordNet [Fellbaum 1998], *Roget's* [Roget 1911], the *Macquarie* [Bernard 1990] and Moby[2]). Automatic techniques give us the opportunity to model language change over time or across domains and genres. McCarthy et al. (2004) investigate using distributional similarity methods to find predominant word senses within a corpus, making it possible to tailor an existing resource (WordNet) to specific domains. For example, in the computing domain, the word *worm* is more likely to be used in its 'malicious computer program' sense than in its 'earthworm' sense. This domain knowledge will be reflected in a thesaurus automatically generated from a computing-specific corpus, which will show increased similarity between *worm* and *virus* and reduced similarity between *worm* and *caterpillar*.

In other application areas, however, the requirement for "similar" words to be semantically related as well as distributionally related is less clear. For example, in prepositional phrase attachment ambiguity resolution, it is necessary to decide whether the prepositional phrase attaches to the verb or the noun as in the examples (1) and (2).

1. Mary ((visited their cottage) with her brother).
2. Mary (visited (their cottage with a thatched roof)).

Hindle and Rooth (1993) note that the correct decision depends on all four lexical events (the verb, the object, the preposition, and the prepositional object). However, a statistical model built on the basis of four lexical events must cope with extremely sparse data. One approach (Resnik 1993; Li and Abe 1998; Clark and Weir 2000) is to induce probability distributions over semantic classes rather than lexical items. For example, a *cottage* is a type of *building* and a *brother* is a type of *person*, and so the co-occurrence of any type of building and any type of person might increase the probability that the PP in example (1) attaches to the verb.

However, it is unclear whether the classes over which probability distributions are induced need to be semantic or whether they could be purely distributional. If we know that two words tend to behave the same way with respect to prepositional phrase attachment, does it matter whether they mean similar things? Other arguments for using semantic classes over distributional classes can similarly be disputed (Weeds 2003). For example, it is not necessary for a class of objects to have a name or symbolic label for us to know that the objects are similar and to exploit that information. Distributional classes do conflate word senses, but in a task such as PP-attachment ambiguity resolution, we are unlikely to be working with sense-tagged examples and therefore it is for word forms that we will wish to estimate probabilities of different attachments. Finally, distributional classes may be over-fitted to a specific corpus, but this may be beneficial to the extent that the over-fitting reflects a specific domain or dialect.

Further, recent empirical evidence suggests that techniques based on distributional similarity may perform as well on this task as those based on semantic similarity. Li (2002) shows that using a fairly small corpus (126,084 sentences from the *Wall Street Journal*) and a distributional similarity technique, it is possible to outperform a state-of-the-art, WordNet-based technique in terms of accuracy, although not in terms of coverage. Pantel and Lin (2000) report performance of 84.3% using an unsupervised approach to prepositional phrase attachment based on distributional similarity

2 The Moby Thesaurus is a product of the Moby Project, which was released into the public domain by Grady Ward in 1996.

techniques. This significantly outperforms previous unsupervised techniques and is drawing close to the state-of-the-art supervised techniques (88.2%).

Having discussed why distributional similarity is important, we now turn to how to formulate it. As we have said, two words are distributionally similar if they appear in similar contexts. We therefore need to consider what is meant by *context*. For example, two words could be considered to appear in the same context if they appear in the same sentence, the same document, or the same grammatical dependency relation. The effect of the type of context used is discussed by Kilgarriff and Yallop (2000). They show that the use of sentence-level and document-level context leads to "looser" thesauri more akin to *Roget's*, whereas the use of grammatical dependency relation level context leads to "tighter" thesauri more akin to WordNet. The use of grammatical dependency relations as context gives us a tighter thesaurus because it restricts distributionally similar words to those that are plausibly inter-substitutable (Church et al. 1994), giving us the following definition of distributional similarity:

> The distributional similarity of two words is the extent to which they can be inter-substituted without changing the plausibility[3] of the sentence.

This concept of lexical substitutability highlights the relationship between distributional similarity and semantic similarity, since semantic similarity can be thought of as the degree of synonymy that exists between two words, where synonymy is defined (Church et al. 1994) as follows:

> Two words are absolute synonyms if they can be inter-substituted in all possible contexts without changing the meaning.

In our empirical work, we focus on finding semantic relationships between words such as *synonymy*, *antonymy* and *hyponymy* that might be found in a tighter thesaurus such as WordNet. Hence, the proposed framework is based on the concept of substitutability, and we use grammatical dependency relations as context. However, since the framework is based on *features*, there is no reason why someone wishing to find topical relationships between words, as might be found in *Roget's*, could not use the framework. We simply do not repeat the earlier work of Kilgarriff and Yallop (2000).

However, a number of questions still remain, which this work does investigate:

1. Is lexical substitutability and therefore distributional similarity symmetric? The concept of substitution is inherently asymmetric. It is possible to measure the appropriateness of substituting word A for word B without measuring the appropriateness of substituting word B for word A. Similarity has been defined in terms of *inter-substitutability*; but we ask whether there is something in the inherent asymmetry of substitution that can be exploited by an asymmetric measure of distributional similarity.
2. Are all contexts equally important? For example, some verbs, e.g., *have* and *get*, are selectionally weak in the constraints they place on their arguments (Resnik 1993). Should such contexts be considered on equal terms with selectionally strong contexts in the calculation of distributional similarity?

3 We use "plausible sentence" to refer to a sentence that might be observed in naturally occurring language data.

3. Is it necessary to consider the difference in extent to which each word appears in each context? Is it enough to know that both words can occur in each context, or do similar words occur in similar contexts with similar probabilities?

In order to answer these questions, we take a pragmatic, application-oriented approach to evaluation that is based on the assumption that we want to know which words are distributionally similar because particular applications can make use of this information.

However, high performance in one application area is not necessarily correlated with high performance in another application area (Weeds and Weir 2003a). Thus, it is not clear that the same characteristics that make a distributional similarity measure useful in one application will make it useful in another. For example, with regard to the question about symmetry, in some applications we may prefer a word A that can be substituted for word B in all of the contexts in which B occurs. In other applications, we may prefer a word A that can be substituted for word B in all of the contexts in which A occurs. For example, asked for a semantically related word to *dog*, we might say *animal*, since *animal* can generally be used in place of *dog*, whereas we might be less likely to say *dog* for *animal*, since *dog* cannot generally be used in place of *animal*. This preference in the direction of the relationship between the two words is not necessarily maintained when one considers language modeling in the face of sparse data. If we want to learn what other contexts *animal* can occur in, we might look at the co-occurrences of words such as *dog*, since we know that *dog* can generally be replaced by *animal*. If we want to learn what other contexts *dog* can occur in, we are less likely to look at the co-occurrences of *animal*, since we know that *animal* can occur in contexts in which *dog* cannot.

Rather than attempt to find a single universally optimal distributional similarity measure, or propose using a radically different distributional similarity measure in each possible application, we propose a flexible, parameterized framework for calculating distributional similarity (Section 2). Within this framework, we cast the problem of finding distributionally similar words as one of *co-occurrence retrieval* (CR), for which we can measure precision and recall by analogy with the way that they are measured in *document retrieval*. Different models within this framework allow us to investigate how frequency information is incorporated into the distributional similarity measure. Different parameter settings within each model allow us to investigate asymmetry in similarity. In Section 3 we discuss the data and the neighbor set comparison technique used throughout our empirical work. In Section 4 we discuss a number of existing distributional similarity measures and discuss the extent to which these can be simulated by settings within the CR framework. In Section 5 we evaluate the CR framework on a semantic task (WordNet prediction) and on a language modeling task (pseudo-disambiguation).

2. Co-occurrence Retrieval

In this section, we present a flexible framework for distributional similarity. This framework directly defines a similarity function, does not require smoothing of the base language model, and allows us to systematically explore the questions about similarity raised in Section 1. In our approach, similarity between words is viewed as a measure of how appropriate it is to use one word (or its distribution) in place of the other. Like relative entropy (Cover and Thomas 1991), it is inherently asymmetric, since we can

measure how appropriate it is to use word A instead of word B separately from how appropriate it is to use word B instead of word A.

The framework presented here is general to the extent that it can be used to compute similarities for any set of objects where each object has an associated set of **features** or **co-occurrence types** and these co-occurrence types have associated frequencies that may be used to form probability estimates. Throughout our discussion, the word for which we are finding neighbors will be referred to as the **target word**. If we are computing the similarity between the target word and another word, then the second word is a **potential neighbor** of the target word. A target word's **nearest neighbors** are the potential neighbors that have the highest similarity with the target word.

2.1 Basic Concepts

Let us imagine that we have formed descriptions of each word in terms of the other words with which they co-occur in various specified grammatical relations in some corpus. For example, the noun *cat* might have the co-occurrence types $\langle$*dobj-of, feed*$\rangle$ and $\langle$*ncmod-by, hungry*$\rangle$. Now let us imagine that we have lost (or accidentally deleted) the description for word w_2, but before this happened we had noticed that the description of word w_2 was very similar to that of word w_1. For example, the noun *dog* might also have the co-occurrence types $\langle$*dobj-of, feed*$\rangle$ and $\langle$*ncmod-by, hungry*$\rangle$. Hence, we decide that we can use the description of word w_1 instead of the description of word w_2 and are hopeful that nobody will notice. How well we do will depend on the validity of substituting w_1 for w_2, or, in other words, the similarity between w_1 and w_2.

The task we have set ourselves can be seen as **co-occurrence retrieval** (CR). By analogy with information retrieval, where there is a set of documents that we would like to retrieve and a set of documents that we do retrieve, we have a scenario where there is a set of co-occurrences that we would like to retrieve, the co-occurrences of w_2, and a set of co-occurrences that we have retrieved, the co-occurrences of w_1. Continuing the analogy, we can measure how well we have done in terms of precision and recall, where **precision** tells us how much of what was retrieved was correct and **recall** tells us how much of what we wanted to retrieve was retrieved.

Our flexible framework for distributional similarity is based on this notion of co-occurrence retrieval. As the distribution of word B moves away from being identical to that of word A, its "similarity" with A can decrease along one or both of two dimensions. When B occurs in contexts that word A does not, the result is a loss of precision, but B may remain a high-recall neighbor. For example, we might expect the noun *animal* to be a high-recall neighbor of the noun *dog*. When B does not occur in contexts that A does occur in, the result is a loss of recall but B may remain a high-precision neighbor. For example, we might expect the noun *dog* to be a high-precision neighbor of the noun *animal*. We can explore the merits of symmetry and asymmetry in a similarity measure by varying the relative importance attached to precision and recall. This was the first question posed about distributional similarity in Section 1.

The remainder of this section is devoted to defining two types of co-occurrence retrieval model (CRM). **Additive models** are based on the Boolean concept of two objects either sharing or not sharing a particular feature (where objects are words and features are co-occurrence types). **Difference-weighted models** incorporate the difference in extent to which each word has each feature. Exploring the two types of models, both defined on the same concepts of precision and recall, allows us to investigate the third question posed in Section 1: Is a shared context worth the same, regardless of the difference in the extent to which each word appears in that context?

We also use the CR framework to investigate the second question posed about distributional similarity, "Should all contexts be treated equally?," by using different weight functions within each type of model. Weight functions decide which co-occurrence types are features of a word and determine the relative importance of features. In previous work (Weeds and Weir 2003b), we experimented with weight functions based on combinatorial, probabilistic, and mutual information (MI). These allow us to define type-based, token-based, and MI-based CRMs, respectively. This work extends the previous work by also considering weighted mutual information (WMI) (Fung and McKeown 1997), the t-test (Manning and Schütze 1999), the z-test (Fontenelle et al. 1994), and an approximation to the log-likelihood ratio (Manning and Schütze 1999) as weight functions.

2.2 Additive Models

Having considered the intuition behind calculating precision and recall for co-occurrence retrieval, we now formulate this formally in terms of an additive model.

We first need to consider for each word w which co-occurrence types will be retrieved, or predicted, by it and, conversely, required in a description of it. We will refer to these co-occurrence types as the features of w, $F(w)$:

$$F(w) = \{c : D(w,c) > 0\} \tag{1}$$

where $D(w,c)$ is the weight associated with word w and co-occurrence type c. Possible weight functions will be described in Section 2.3.

The shared features of word w_1 and word w_2 are referred to as the set of True Positives, $TP(w_1,w_2)$, which will be abbreviated to TP in the rest of this article:

$$TP(w_1,w_2) = F(w_1) \cap F(w_2) \tag{2}$$

The precision of w_1's retrieval of w_2's features is the proportion of w_1's features that are shared by both words, where each feature is weighted by its relative importance according to w_1:

$$\mathcal{P}^{add}(w_1,w_2) = \frac{\sum_{TP} D(w_1,c)}{\sum_{F(w_1)} D(w_1,c)} \tag{3}$$

The recall of w_1's retrieval of w_2's features is the proportion of w_2's features that are shared by both words, where each feature is weighted by its relative importance according to w_2:

$$\mathcal{R}^{add}(w_1,w_2) = \frac{\sum_{TP} D(w_2,c)}{\sum_{F(w_2)} D(w_2,c)} \tag{4}$$

Table 1
Weight functions.

$D_{type}(w,c)$	=	$\begin{cases} 1 & \text{if } P(c\|w) > 0 \\ 0 & \text{otherwise} \end{cases}$
$D_{tok}(w,c)$	=	$P(c\|w)$
$D_{mi}(w,c)$	=	$I(w,c) = \log\left(\frac{P(c,w)}{P(c)P(w)}\right)$
$D_{wmi}(w,c)$	=	$P(c,w) \cdot \log\left(\frac{P(c,w)}{P(c)P(w)}\right)$
$D_{t}(w,c)$	=	$\frac{P(c,w)-P(c)\cdot P(w)}{\sqrt{\frac{P(c,w)}{N}}}$
$D_{z}(w,c)$	=	$\frac{P(c,w)-P(c)\cdot P(w)}{\sqrt{\frac{P(c)\cdot P(w)}{N}}}$
$D_{allr}(w,c)$	=	$-2 \cdot \left(\log L\left(F(w,c), F(w), \frac{F(c)}{N}\right) - \log L\left(F(w,c), F(w), \frac{F(w,c)}{F(w)}\right)\right)$

Precision and recall both lie in the range [0,1] and are both equal to one when each word has exactly the same features. It should also be noted that the recall of w_1's retrieval of w_2 is equal to the precision of w_2's retrieval of w_1, i.e., $\mathcal{R}^{add}(w_1, w_2) = \mathcal{P}^{add}(w_2, w_1)$.

2.3 Weight Functions

The weight function plays two important roles. First, it determines which co-occurrences of w_1 and w_2 are important enough to be considered part of their description, or by analogy with document retrieval, which co-occurrences we want to retrieve for w_2 and which co-occurrences we have retrieved using the description of w_1. It is then used to weight contexts by their importance. In the latter case, $D(w_1, c)$ tells us the retrieval process's perceived relevance of co-occurrence type c, and $D(w_2, c)$ tells us the actual relevance of co-occurrence type c. The weight functions we have considered so far are summarized in Table 1. Each weight function can be used to define its own CRM, which we will now discuss in more detail.

Additive type-based CRM (D_{type}). In this CRM, the precision of w_1's retrieval of w_2 is the proportion of co-occurrence types occurring with w_1 that also occur with w_2, and the recall of w_1's retrieval of w_2 is the proportion of verb co-occurrence types (or distinct verbs) occurring with w_2 that also occur with w_1. In this case, the summed values of D are always 1, and hence the expressions for precision and recall can be simplified:

$$\mathcal{P}^{add}_{type}(w_1, w_2) = \frac{\sum_{TP} D_{type}(w_1, c)}{\sum_{F(w_1)} D_{type}(w_1, c)} = \frac{|TP|}{|F(w_1)|} \tag{5}$$

$$\mathcal{R}^{add}_{type}(w_1, w_2) = \frac{\sum_{TP} D_{type}(w_2, c)}{\sum_{F(w_2)} D_{type}(w_2, c)} = \frac{|TP|}{|F(w_2)|} \quad (6)$$

Additive token-based CRM (D_{tok}). In this CRM, the precision of w_1's retrieval of w_2 is the proportion of co-occurrence tokens occurring with w_1 that also occur with w_2, and the recall of w_1's retrieval of w_2 is the proportion of co-occurrence tokens occurring with w_2 that also occur with w_1. Hence, words have the same features as in the type-based CRM, but each feature is given a weight based on its probability of occurrence. Since $F(w) = \{c : D(w, c) > 0\} = \{c : P(c|w) > 0\}$, it follows that $\sum_{F(w)} D_{tok}(w, c) = 1$, and therefore the expressions for precision and recall can be simplified:

$$\mathcal{P}^{add}_{tok}(w_1, w_2) = \frac{\sum_{TP} D_{tok}(w_1, c)}{\sum_{F(w_1)} D_{tok}(w_1, c)} = \sum_{TP} P(c, w_1) \quad (7)$$

$$\mathcal{R}^{add}_{tok}(w_1, w_2) = \frac{\sum_{TP} D_{tok}(w_2, c)}{\sum_{F(w_2)} D_{tok}(w_2, c)} = \sum_{TP} P(c, w_2) \quad (8)$$

Additive MI-based CRM (D_{mi}). Using pointwise mutual information (MI) (Church and Hanks 1989) as the weight function means that a co-occurrence c is considered a feature of word n if the probability of their co-occurrence is greater than would be expected if words occurred independently. In addition, more informative co-occurrences contribute more to the sums in the calculation of precision and recall and hence have more weight.

Additive WMI-based CRM (D_{wmi}). Weighted mutual information (WMI) (Fung and McKeown 1997) has been proposed as an alternative to MI, particularly when MI might lead to the over-association of low-frequency events. In this function, the pointwise MI is multiplied by the probability of the co-occurrence; hence, reducing the weight assigned to low-probability events.

Additive t-test based CRM (D_t). The t-test (Manning and Schütze 1999) is a standard statistical test that has been proposed for collocation analysis. It measures the (signed) difference between the observed probability of co-occurrence and the expected probability of co-occurrence, as would be observed if words occurred independently. The difference is divided by the standard deviation in the observed distribution. Similarly to MI, this score obviously gives more weight to co-occurrences that occur more than would be expected, and its use as the weight function results in any co-occurrences that occur less than would be expected being ignored.

Additive z-test based CRM (D_z). The z-test (Fontenelle et al. 1994) is almost identical to the t-test. However, using the z-test, the (signed) difference between the observed probability of co-occurrence and the expected probability of co-occurrence is divided by the standard deviation in the expected distribution.

Additive log-likelihood ratio based CRM (D_{allr}). The log-likelihood ratio (Manning and Schütze 1999) considers the difference (as a log ratio) in probability of the observed frequencies of co-occurrences and individual words occurring under the null hypoth-

esis, that words occur independently, and under the alternative hypothesis, that they do not.

$$H_0 : P(c|w) = p = P(c|\neg w) \tag{9}$$

$$H_1 : P(c|w) = p_1 \neq p_2 = P(c|\neg n) \tag{10}$$

If $f(w,c)$ is the frequency of w and c occurring together, $f(w)$ is the total frequency of w occurring in any context, $f(c)$ is the total frequency of c occurring with any word, and N is the grand total of co-occurrences, then the log-likelihood ratio can be written:

$$\text{Log}\lambda(w,c) = -2.\log\frac{L(H_0)}{L(H_1)} \tag{11}$$

$$= -2.\begin{pmatrix} \log L\left(f(w,c), f(w), \frac{f(c)}{N}\right) \\ +\log L\left(f(c) - f(w,c), N - f(w), \frac{f(c)}{N}\right) \\ -\log L\left(f(w,c), f(w), \frac{f(w,c)}{f(w)}\right) \\ -\log L\left(f(c) - f(w,c), N - f(w), \frac{f(c)-f(w,c)}{N-f(w)}\right) \end{pmatrix} \tag{12}$$

$$\text{where} \quad L(k,n,x) = x^k(1-x)^{n-k} \tag{13}$$

In our implementation (see Table 1), an approximation to this formula is used, which we term the ALLR weight function. We use an approximation because the terms that represent the probabilities of the other contexts (i.e., seeing $f(c) - f(w,c)$ under each hypothesis) tend towards $-\infty$ as N increases (since the probabilities tend towards zero). Since N is very large in our experiments (approximately 2,000,000), we found that using the full formula led to many weights being undefined. Further, since in this case the probability of seeing other contexts will be approximately equal under each hypothesis, it is a reasonable approximation to make.

Another potential problem with using the log-likelihood ratio as the weight function is that it is always positive, since the observed distribution is always more probable than the hypothesized distribution. All of the other weight functions assign a zero or negative weight to co-occurrence types that do not occur with a given word and thus these zero frequency co-occurrence types are never selected as features. This is advantageous in the computation of similarity, since computing the sums over all co-occurrence types rather than just those co-occurring with at least one of the words is (1) very computationally expensive and (2) due to their vast number, the effect of these zero frequency co-occurrence types tends to outweigh the effect of those co-occurrence types that have actually occurred. Giving such weight to these shared non-occurrences seems unintuitive and has been shown by Lee (1999) to be undesirable in the calculation of distributional similarity. Hence, when using the

ALLR as the weight function, we use the additional restriction that $P(c, w) > 0$ when selecting features.

2.4 Difference-Weighted Models

In additive models, no distinction is made between features that have occurred to the same extent with each word and features that have occurred to different extents with each word. For example, if two words have the same features, they are considered identical, regardless of whether the feature occurs with the same probability with each word or not. Here, we define a type of model that allows us to capture the difference in the extent to which each word has each feature.

We do this by defining the similarity of two words with respect to an *individual* feature, using the same principles that we use to define the similarity of two words with respect to all their features. First, we define an **extent** function, $E(w, c)$, which is the extent to which w_1 goes with c and which may be, but is not necessarily, the same as the weight function $D(n, w)$. Possible extent functions will be discussed in Section 2.5. Having defined this function, we can measure the precision and recall of individual features. The precision of an individual feature c retrieved by w_1 is the extent to which both words go with c divided by the extent to which w_1 goes with c. The recall of the retrieval of c by w_1 is the extent to which both words go with c divided by the extent to which w_2 goes with c.

$$\mathcal{P}(w_1, w_2, c) = \frac{min(E(w_1, c), E(w_2, c))}{E(w_1, c)} \tag{14}$$

$$\mathcal{R}(w_2, w_1, c) = \frac{min(E(w_1,\ c), E(w_2,\ c))}{E(w_2,\ c)} \tag{15}$$

Precision and recall of an individual feature, like precision and recall of a distribution, lie in the range [0,1]. We can now redefine precision and recall of a distribution as follows:

$$\mathcal{P}^{dw}(w_1, w_2) = \frac{\sum_{TP} D(w_1, c) \cdot \mathcal{P}(w_1, w_2, c)}{\sum_{F(w_1)} D(w_1, c)} \tag{16}$$

$$\mathcal{R}^{dw}(w_1, w_2) = \frac{\sum_{TP} D(w_2, c) \cdot \mathcal{R}(w_1, w_2, c)}{\sum_{F(w_2)} D(w_2, c)} \tag{17}$$

Using precision and recall of individual features as weights in the definitions of precision and recall of a distribution captures the intuition that retrieval of a co-occurrence type is not a black-and-white matter. Features that are shared to a similar extent are considered more important in the calculation of distributional similarity.

Table 2
Extent functions.

$E_{type}(w,c) = P(c|w)$

$E_{tok}(w,c) = P(c|w)$

$E_{mi}(w,c) = I(w,c) = \log\left(\frac{P(c,w)}{P(c)P(w)}\right)$

$E_{wmi}(w,c) = P(c,w).\log\left(\frac{P(c,w)}{P(c)P(w)}\right)$

$E_{t}(w,c) = \frac{P(c,w)-P(c).P(w)}{\sqrt{\frac{P(c,w)}{N}}}$

$E_{z}(w,c) = \frac{P(c,w)-P(c).P(w)}{\sqrt{\frac{P(c).P(w)}{N}}}$

$E_{allr}(w,c) = -2.\left(\log L\left(f(w,c),f(w),\frac{f(c)}{N}\right) - \log L\left(f(w,c),f(w),\frac{f(w,c)}{f(w)}\right)\right)$

2.5 Extent Functions

The extent functions we have considered so far are summarized in Table 2. Note that in general, the extent function is the same as the weight function, which leads to a standard simplification of the expressions for precision and recall in the difference-weighted CRMs. For example, in the *difference-weighted MI-based model* we get the expressions:

$$\mathcal{P}^{dw}_{mi}(w_1,w_2) = \frac{\sum_{TP} I(w_1,c)\cdot\frac{min(I(w_1,c),I(w_2,c))}{I(w_1,c)}}{\sum_{F(w_1)} I(w_1,c)} = \frac{\sum_{TP} min(I(w_1,c),I(w_2,c))}{\sum_{F(w_1)} I(w_1,c)} \tag{18}$$

$$\mathcal{R}^{dw}_{mi}(w_1,w_2) = \frac{\sum_{TP} I(w_2,c)\cdot\frac{min(I(w_2,c),I(w_1,c))}{I(w_2,c)}}{\sum_{F(w_2)} I(w_2,c)} = \frac{\sum_{TP} min(I(w_2,c),I(w_1,c))}{\sum_{F(w_2)} I(w_2,c)} \tag{19}$$

Similar expressions can be derived for the *WMI-based CRM*, the *t-test based CRM*, the *z-test based CRM*, and the *ALLR-based CRM*. An interesting special case is the *difference-weighted token-based CRM*. In this case, since $\sum_{F(w)} P(c|w) = 1$, we derive the following expressions for precision and recall:

$$\mathcal{P}^{dw}_{tok}(w_1,w_2) = \frac{\sum_{TP} P(c|w_1)\cdot\frac{min(P(c|w_1),P(c|w_2))}{P(c|w_1)}}{\sum_{F(w_1)} P(c|w_1)} = \sum_{TP} min(P(c|w_1),P(c|w_2)) \tag{20}$$

$$\mathcal{R}^{dw}_{tok}(w_1, w_2) = \frac{\sum_{TP} P(c|w_2) \cdot \frac{min(P(c|w_2), P(c|w_1))}{P(c|w_2)}}{\sum_{F(w_2)} P(c|w_2)} = \sum_{TP} min(P(c|w_2), P(c|w_1))$$
$$= \mathcal{P}^{dw}_{tok}(w_1, w_2) \quad (21)$$

Note that although we have defined separate precision and recall functions, we have arrived at the same expression for both in this model. As a result, this model is symmetric.

The only CRM in which we use a different extent and weight function is the *difference-weighted type-based CRM*. This is because there is no difference between types and tokens for an individual feature; i.e., their retrieval is equivalent. In this case, the following expressions for precision and recall are derived:

$$\mathcal{P}^{dw}_{type}(w_1, w_2) = \frac{\sum_{TP} D_{type}(w_1, c) \cdot \mathcal{P}_{type}(w_1, w_2, c)}{\sum_{F(w_1)} D_{type}(w_1, c)} = \frac{\sum_{TP} \frac{min(P(c|w_1), P(c|w_2))}{P(c|w_1)}}{|F(w_1)|} \quad (22)$$

$$\mathcal{R}^{dw}_{type}(w_1, w_2) = \frac{\sum_{TP} D_{type}(w_2, c) \cdot \mathcal{R}_{type}(w_1, w_2, c)}{\sum_{F(w_2)} D_{type}(w_2, c)} = \frac{\sum_{TP} \frac{min(P(c|w_2), P(c|w_1))}{P(c|w_2)}}{|F(w_2)|} \quad (23)$$

Note that this is different from the additive token-based model because, although every token is effectively considered in this model, tokens are not weighted equally. In this model, tokens are treated differently according to which type they belong. The importance of the retrieval (or non-retrieval) of a single token depends on the proportion of the tokens for its particular type that it constitutes.

2.6 Combining Precision and Recall

We have, so far, been concerned with defining a pair of numbers that represents the similarity between two words. However, in applications, it is normally necessary to compute a single number in order to determine neighborhood or cluster membership. The classic way to combine precision and recall in IR is to compute the F-score; that is, the harmonic mean of precision and recall:

$$F = m_h(\mathcal{P}, \mathcal{R}) = \frac{2 \cdot \mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}} \quad (24)$$

However, we do not wish to assume that a good substitute requires both high precision and high recall of the target distribution. It may be that, in some situations, the best word to use in place of another word is one that only retrieves correct co-occurrences (i.e., it is a high-precision neighbor) or it may be one that retrieves all of the required co-occurrences (i.e., it is a high-recall neighbor). The other factor in each case may play only a secondary role or no role at all.

We can retain generality and investigate whether high precision *or* high recall *or* high precision *and* high recall are required for high similarity by computing a weighted

Table 3
Table of special values of β and γ.

β	γ	Special Case
-	1	harmonic mean of precision and recall (F-score)
β	0	weighted arithmetic mean of precision and recall
1	0	precision
0	0	recall
0.5	0	unweighted arithmetic mean

arithmetic mean of the harmonic mean and the weighted arithmetic mean of precision and recall:[4]

$$m_h(\mathcal{P}(w_1,w_2),\mathcal{R}(w_1,w_2)) = \frac{2.\mathcal{P}(w_1,w_2).\mathcal{R}(w_1,w_2)}{\mathcal{P}(w_1,w_2)+\mathcal{R}(w_1,w_2)} \tag{25}$$

$$m_a(\mathcal{P}(w_1,w_2),\mathcal{R}(w_1,w_2)) = \beta.\mathcal{P}(w_1,w_2)+(1-\beta).\mathcal{R}(w_1,w_2) \tag{26}$$

$$\begin{aligned} sim(w_1,w_2) = {} & \gamma.m_h(\mathcal{P}(w_1,w_2),\mathcal{R}(w_1,w_2)) \\ & +(1-\gamma).m_a(\mathcal{P}(w_1,w_2),\mathcal{R}(w_1,w_2)) \end{aligned} \tag{27}$$

where both β and γ lie in the range [0,1]. The resulting similarity, $sim(w_1,w_2)$, will also lie in the range [0,1] where 0 is low and 1 is high. This formula can be used in combination with any of the models for precision and recall outlined earlier. Precision and recall can be computed once for every pair of words (and every model) whereas similarity depends on the values of β and γ. The flexibility allows us to investigate empirically the relative significance of the different terms and thus whether one (or more) might be omitted in future work. Table 3 summarizes some special parameter settings.

2.7 Discussion

We have developed a framework based on the concept of co-occurrence retrieval (CR). Within this framework we have defined a number of models (CRMs) that allow us to systematically explore three questions about similarity. First, is similarity between words necessarily a symmetric relationship, or can we gain an advantage by considering it as an asymmetric relationship? Second, are some features inherently more salient than others? Third, does the difference in extent to which each word takes each feature matter?

The CRMs and the parameter settings therein correspond to alternative possibilities. First, a high-precision neighbor is not necessarily a high-recall neighbor (and, conversely, a high-recall neighbor is not necessarily a high-precision neighbor) and therefore we are not constrained to a symmetric relationship of similarity between

4 This is as opposed to using a standard weighted F-score (Manning and Schütze 1999), which uses just one parameter α: $F_\alpha = \frac{\mathcal{PR}}{\alpha\mathcal{R}+(1-\alpha)\mathcal{P}}$. We did not use this weighting because we wished to investigate the differences between using an arithmetic mean and a harmonic mean.

words. Second, the use of different weight functions varies the relative importance attached to features. Finally, difference-weighted models contrast with additive models in considering the difference in extent to which each word takes each feature.

3. Data and Experimental Techniques

The rest of this paper is concerned with evaluation of the proposed framework; first, by comparing it to existing distributional similarity measures, and second, by evaluating performance on two tasks. Throughout our empirical work, we use one data-set and one neighbor set comparison technique, which we now discuss in advance of presenting any of our actual experiments.

3.1 Data

The data used for all our experimental work was noun-verb direct-object data extracted from the BNC by a Robust Accurate Statistical Parser (RASP) (Briscoe and Carroll 1995; Carroll and Briscoe 1996). We constructed a list of nouns that occur in both our data set and WordNet ordered by their frequency in our corpus data. Since we are interested in the effects of word frequency on word similarity, we selected 1,000 high-frequency nouns and 1,000 low-frequency nouns. The 1,000 high-frequency nouns were selected as the nouns with frequency ranks of 1–1,000; this corresponds to a frequency range of [586,20871]. The low-frequency nouns were selected as the nouns with frequency ranks of 3,001–4,000; this corresponds to a frequency range of [72,121].

For each target noun, 80% of the available data was randomly selected as training data and the other 20% was set aside as test data.[5] The training data was used to compute similarity scores between all possible pairwise combinations of the 2,000 nouns and to provide (MLE) estimates of noun-verb co-occurrence probabilities in the pseudo-disambiguation task. The test data provides unseen co-occurrences for the pseudo-disambiguation task.

Although we only consider similarity between nouns based on co-occurrences with verbs in the direct-object position, the generality of the techniques proposed is not so restricted. Any of the techniques can be applied to other parts of speech, other grammatical relations, and other types of context. We restricted the scope of our experimental work solely for computational and evaluation reasons. However, we could have chosen to look at the similarity between verbs or between adjectives.[6] We chose nouns as a starting point since nouns tend to allow less sense extensions than verbs and adjectives (Pustejovsky 1995). Further, the noun hyponymy hierarchy in WordNet, which will be used as a pseudo-gold standard for comparison, is widely recognized in this area of research.

Some previous work on distributional similarity between nouns has used only a single grammatical relation (e.g., Lee 1999), whereas other work has considered multiple grammatical relations (e.g., Lin 1998a). We consider only a single grammatical relation because we believe that it is important to evaluate the usefulness of each grammatical relation in calculating similarity before deciding how to combine information from

5 This results in a single 80:20 split of the complete data set, in which we are guaranteed that the original relative frequencies of the target nouns are maintained.

6 The use of grammatical relations to model context precludes finding similarities between words of different parts of speech. Since we are looking at similarity in terms of substitutability, we would not expect to find a word of one part of speech substitutable for a word of another part of speech.

different relations. In previous work (Weeds 2003), we found that considering the subject relation as well as the direct-object relation did not improve performance on a pseudo-disambiguation task.

Our last restriction was to only consider 2,000 of the approximately 35,000 nouns occurring in the corpus. This restriction was for computational efficiency and to avoid computing similarities based on the potentially unreliable descriptions of very low-frequency words. However, since our evaluation is comparative, we do not expect our results to be affected by this or any of the other restrictions.

3.2 Neighbor Set Comparison Technique

In several of our experiments, we measure the overlap between two different similarity measures. We use a neighbor set comparison technique adapted from Lin (1997).

In order to compare two neighbor sets of size k, we transform each neighbor set so that each neighbor is given a rank score of $k - rank$. Potential neighbors not within a given rank distance k of the noun score zero. This transformation is required since scores computed on different scales are to be compared and because we wish to only consider neighbors up to a certain rank distance. The similarity between two neighbor sets S and S' is computed as the cosine of the rank score vectors:

$$C(S, S') = \frac{\sum_{w \in S \cap S'} s(w) \cdot s'(w)}{\sum_{i=1}^{k} i^2} \tag{28}$$

where $s(w)$ and $s'(w)$ are the rank scores of the words within each neighbor set S and S' respectively.

In previous work (Weeds and Weir 2003b), having computed the similarity between neighbor sets for each noun according to each pair of measures under consideration, we computed the mean similarity across all high-frequency nouns and all low-frequency nouns. However, since the use of the CR framework requires parameter optimization, here, we randomly select 60% of the nouns to form a development set and use the remaining 40% as a test set. Thus, any parameters are optimized over the development set nouns and performance measured at these settings over the test set.

4. Alternative Distributional Similarity Measures

In this section, we consider related work on distributional similarity measures and the extent to which some of these measures can be simulated within the CR framework. However, there is a large body of work on distributional similarity measures; for a more extensive review, see Weeds (2003). Here, we concentrate on a number of more popular measures: the Dice Coefficient, Jaccard's Coefficient, the L_1 Norm, the α-skew divergence measure, Hindle's measure, and Lin's MI-based measure.

4.1 The Dice Coefficient

The Dice Coefficient (Frakes and Baeza-Yates 1992) is a popular combinatorial similarity measure adopted from the field of Information Retrieval for use as a measure of lexical

distributional similarity. It is computed as twice the ratio between the size of the intersection of the two feature sets and the sum of the sizes of the individual feature sets:

$$sim_{dice}(w_1, w_2) = \frac{2 \cdot |F(w_1) \cap F(w_2)|}{|F(w_1)| + |F(w_2)|}$$
$$\text{where} \quad F(w) = \{c : P(c|w) > 0\} \tag{29}$$

According to this measure, the similarity between words with no shared features is zero and the similarity between words with identical feature sets is 1. However, as shown below, this formula is equivalent to a special case in the CR framework: the harmonic mean of precision and recall (or F-score) using the additive type-based CRM.

$$\begin{aligned} m_h(\mathcal{P}^{add}_{type}(w_1, w_2), \mathcal{R}^{add}_{type}(w_1, w_2)) &= \frac{2 \cdot \mathcal{P}^{add}_{type}(w_1, w_2) \cdot \mathcal{R}^{add}_{type}(w_1, w_2)}{\mathcal{P}^{add}_{type}(w_1, w_2) + \mathcal{R}^{add}_{type}(w_1, w_2)} \\ &= \frac{2 \cdot \frac{|TP|}{|F(w_1)|} \cdot \frac{|TP|}{|F(w_2)|}}{\frac{|TP|}{|F(w_1)|} + \frac{|TP|}{|F(w_2)|}} = \frac{2 \cdot |TP| \cdot |TP|}{|TP| \cdot |F(w_1)| + |TP| \cdot |F(w_2)|} \\ &= \frac{2 \cdot |TP|}{|F(w_1)| + |F(w_2)|} = \frac{2 \cdot |F(w_1) \cap F(w_2)|}{|F(w_1)| + |F(w_2)|} \\ &= sim_{dice}(w_1, w_2) \end{aligned} \tag{30}$$

Thus, when γ is set to 1 in the additive type-based CRM, the Dice Coefficient is exactly replicated.

4.2 Jaccard's Coefficient

Jaccard's Coefficient (Salton and McGill 1983), also known as the Tanimoto Coefficient (Resnik 1993), is another popular combinatorial similarity measure. It can be defined as the proportion of features belonging to either word that are shared by both words; that is, the ratio between the size of the intersection of the feature sets and the size of the union of feature sets:

$$sim_{jacc}(w_1, w_2) = \frac{|F(w_1) \cap F(w_2)|}{|F(w_1) \cup F(w_2)|} \tag{31}$$

As with the Dice Coefficient, the similarity between words with no shared co-occurrences is zero and the similarity between words with identical features is 1. Further, as shown by van Rijsbergen (1979), the Dice Coefficient and Jaccard's Coefficient are monotonic in one another. Thus, although in general the scores computed by each will be different, the orderings or rankings of objects will be the same. In other words, for all k and w, the k nearest neighbors of word w according to Jaccard's Coefficient will be identical to the k nearest neighbors of word w according to the Dice Coefficient and the harmonic mean of precision and recall in the additive type-based CRM.

4.3 The L_1 Norm

The L_1 Norm (Kaufman and Rousseeuw 1990) is a member of a family of measures known as the Minkowski Distance, for measuring the distance[7] between two points in space. The L_1 Norm is also known as the Manhattan Distance, the taxi-cab distance, the city-block distance, and the absolute value distance, since it represents the distance traveled between the two points if you can only travel in orthogonal directions. When used to calculate lexical distributional similarity, the dimensions of the vector space are co-occurrence types and the values of the vector components are the probabilities of the co-occurrence types given the word. Thus the L_1 distance between two words, w_1 and w_2, can be written as:

$$dist_{L_1}(w_1, w_2) = \sum_c |P(c|w_1) - P(c|w_2)| \tag{32}$$

However, noting the algebraic equivalence $A + B - |A - B| \equiv 2 \cdot min(A, B)$ and using basic probability theory, we can rewrite the L_1 Norm as follows:

$$\begin{aligned} dist_{L_1}(w_1, w_2) &= \sum_c |P(c|w_1) - P(c|w_2)| \\ &= \sum_c P(c|w_1) + P(c|w_2) - 2 \cdot min(P(c|w_1), P(c|w_2)) \\ &= \sum_c P(c|w_1) + \sum_c P(c|w_2) - 2 \cdot \sum_c min(P(c|w_1), P(c|w_2)) \\ &= 2 - 2 \cdot \sum_c min(P(c|w_1), P(c|w_2)) \end{aligned} \tag{33}$$

However, $min(P(c|w_1), P(c|w_2)) > 0$ if and only if $c \in TP$. Hence:

$$dist_{L_1}(w_1, w_2) = 2 - 2 \cdot \sum_{TP} min(P(c|w_1), P(c|w_2) = 2 - 2 \cdot sim^{dw}_{tok}(w_1, w_2) \tag{34}$$

In other words, the L_1 Norm is directly related to the difference-weighted token-based CRM. The constant and multiplying factors are required, since the CRM defines a similarity in the range [0,1], whereas the L_1 Norm defines a distance in the range [0,2] (where 0 distance is equivalent to 1 on the similarity scale).

4.4 The α-skew Divergence Measure

The α-skew divergence measure (Lee 1999, 2001) is a popular approximation to the Kullback-Leibler divergence measure[8] (Kullback and Leibler 1951; Cover and Thomas 1991). It is an approximation developed to be used when unreliable MLE probabilities

7 Distance measures, also referred to as divergence and dissimilarity measures, can be viewed as the inverse of similarity measures; that is, an increase in distance correlates with a decrease in similarity.

8 The Kullback-Leibler divergence measure is also often referred to as "relative entropy."

would result in the actual Kullback-Leibler divergence measure being equal to ∞. It is defined (Lee 1999) as:

$$dist_\alpha(q, r) = D(r||\alpha.q + (1 - \alpha).r) \tag{35}$$

for $0 \leq \alpha \leq 1$, and where:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{36}$$

In effect, the q distribution is smoothed with the r distribution, which results in it always being non-zero when the r distribution is non-zero. The parameter α controls the extent to which the measure approximates the Kullback-Leibler divergence measure. When α is close to 1, the approximation is close while avoiding the problem with zero probabilities associated with using the Kullback-Leibler divergence measure. This theoretical justification for using a very high value of α (e.g., 0.99) is also borne out by empirical evidence (Lee 2001).

The α-skew divergence measure retains the asymmetry of the Kullback-Leibler divergence, and Weeds (2003) discusses the significance in the direction in which it is calculated. For the purposes of this paper, we will find the neighbors of w_2 by optimizing:[9]

$$dist_\alpha(P(c|w_1), P(c|w_2)) \tag{37}$$

Due to the form of the α-skew divergence measure, we do not expect any of the CRMs to exactly simulate it. However, this measure does take into account the differences between the probabilities of co-occurrences in each distribution (as a log ratio) and therefore we might expect that it will be fairly closely simulated by the difference-weighted token-based CRM. Further, the α-skew divergence measure is asymmetric. $dist_\alpha(w_1, w_2)$ measures the cost of using the distribution of w_1 instead of w_2 and is calculated over the verbs that occur with w_2. As such, we might expect that $dist_\alpha$ will be a high-recall measure, since recall is calculated over the co-occurrences of w_2.

In order to determine how close any approximation is in practice, we compared the 200 nearest neighbors according to $dist_\alpha$ and different parameter settings within the CR framework for 1,000 high-frequency nouns and for 1,000 low-frequency nouns, using the data and the neighbor set comparison technique described in Section 3. Table 4 shows the optimal parameters in each CRM for simulating $dist_\alpha$, computed over the development set, and the mean similarity at these settings over both the development set and the test set. From these results, we can make the following observations.

First, the differences in mean similarities over the development set and the test set are minimal. Thus, performance of the models with respect to different parameter settings appears stable across different words.

Second, the differences between the models are fairly small. The difference-weighted token-based CRM achieves a fairly close approximation to $dist_\alpha$, but the

9 This is what Weeds (2003) refers to as $dist_{\alpha 1}$.

Table 4
Optimized similarities between CRMs and $dist_\alpha$ and corresponding parameter settings.

	Target Noun Frequency							
	high				low			
	Optimal Parameters		Devel. Sim.	Test Sim.	Optimal Parameters		Devel. Sim.	Test Sim.
CRM	γ	β			γ	β		
sim^{add}_{type}	0.25	0.4	0.74	0.75	0.5	0.3	0.66	0.66
sim^{add}_{tok}	0.5	0.0	0.77	0.78	0.5	0.0	0.67	0.66
sim^{add}_{mi}	0.0	0.0	0.76	0.77	0.25	0.1	0.72	0.73
sim^{add}_{wmi}	0.25	0.0	0.71	0.72	0.25	0.1	**0.79**	**0.79**
sim^{add}_{t}	0.5	0.0	**0.84**	**0.85**	0.5	0.2	0.71	0.71
sim^{add}_{z}	0.5	0.0	0.79	0.80	0.5	0.1	0.63	0.63
sim^{add}_{allr}	0.25	0.0	0.70	0.71	0.5	0.0	0.64	0.63
sim^{dw}_{type}	0.0	0.25	0.70	0.71	0.0	0.0	0.52	0.53
sim^{dw}_{tok}	—	—	0.79	0.80	—	—	0.58	0.58
sim^{dw}_{mi}	0.0	0.0	0.66	0.68	0.0	0.0	0.60	0.60
sim^{dw}_{wmi}	0.0	0.1	0.66	0.67	0.0	0.1	0.58	0.58
sim^{dw}_{t}	0.5	0.1	0.82	0.83	0.5	0.3	0.69	0.69
sim^{dw}_{z}	0.5	0.0	0.78	0.80	0.5	0.1	0.64	0.64
sim^{dw}_{allr}	0.75	0.0	0.53	0.54	0.75	0.0	0.48	0.48

overall best approximation is achieved by the additive t-test based CRM. Although none of the CRMs are able to simulate $dist_\alpha$ exactly, the closeness of approximation achieved in the best cases (greater than 0.7) is substantially higher than the degree of overlap observed between other measures of distributional similarity. Weeds, Weir, and McCarthy (2004) report an average overlap of 0.4 between neighbor sets produced using $dist_\alpha$ and Jaccard's Measure and an average overlap of 0.48 between neighbor sets produced using $dist_\alpha$ and Lin's similarity measure.

A third observation is that all of the asymmetric models get closest at high levels of recall for both high- and low-frequency nouns. For example, Figure 1 illustrates the variation in mean similarity between neighbor sets with the parameters β and γ for the additive t-test based model. As can be seen, similarity between neighbor sets is significantly higher at high recall settings (low β) within the model than at high-precision settings (high β), which suggests that $dist_\alpha$ has high-recall CR characteristics.

4.5 Hindle's Measure

Hindle (1990) proposed an MI-based measure, which he used to show that nouns could be reliably clustered based on their verb co-occurrences. We consider the variant of

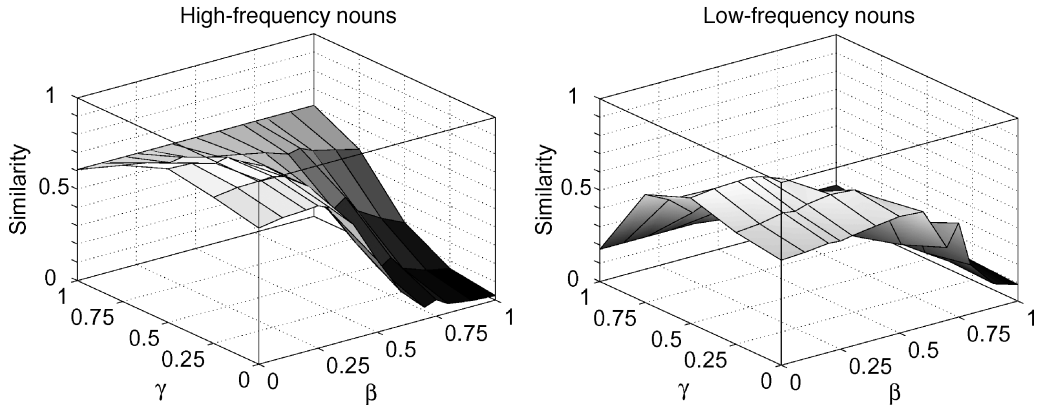

Figure 1
Variation (with parameters β and γ) in development set mean similarity between neighbor sets of the additive t-test based CRM and of $dist_{\alpha}$.

Hindle's Measure proposed by (Lin 1998a), which overcomes the problem associated with calculating MI for word-feature combinations that do not occur:

$$sim_{hind}(w_1, w_2) = \sum_{T(w_1) \cap T(w_2)} min(I(c, w_1), I(c, w_2)) \tag{38}$$

where $T(w_1) = \{c : I(c, n) > 0\}$. This expression is the same as the numerator in the expressions for precision and recall in the difference-weighted MI-based CRM:

$$\mathcal{P}^{dw}_{mi}(w_1, w_2) = \frac{\sum_{TP} I(w_1, c) \cdot \frac{min(I(w_1,c), I(w_2,c))}{I(w_1,c)}}{\sum_{F(w_1)} I(w_1, c)} = \frac{\sum_{TP} min(I(w_1, c), I(w_2, c))}{\sum_{F(w_1)} I(w_1, c)} \tag{39}$$

$$\mathcal{R}^{dw}_{mi}(w_1, w_2) = \frac{\sum_{TP} I(w_2, c) \cdot \frac{min(I(w_2,c), I(w_1,c))}{I(w_2,c)}}{\sum_{F(w_2)} I(w_2, c)} = \frac{\sum_{TP} min(I(w_2, c), I(w_1, c))}{\sum_{F(w_2)} I(w_2, c)} \tag{40}$$

since $TP = T(w_1) \cap T(w_2)$. However, we also note that the denominator in the expression for recall depends only on w_2, and therefore, for a given w_2, is a constant. Since w_2 is the target word, it will remain the same as we calculate each neighbor set. Accordingly, the value of recall for each potential neighbor w_1 of w_2 will be the value of sim_{hind} divided by a constant. Hence, neighbor sets derived using sim_{hind} are identical to those obtained using recall ($\gamma = 0, \beta = 0$) in the difference-weighted MI-based CRM.

4.6 Lin's Measure

Lin (1998a) proposed a measure of lexical distributional similarity based on his information-theoretic similarity theorem (Lin 1997, 1998b):

> The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are.

If the features of a word are grammatical relation contexts, the similarity between two words w_1 and w_2 can be written according to Lin's measure as:

$$sim_{lin}(w_1, w_2) = \frac{\sum_{T(w_1)\cap T(w_2)}(I(w_1, c) + I(w_2, c))}{\sum_{T(w_1)} I(w_1, c) + \sum_{T(w_2)} I(w_2, c)} \tag{41}$$

where $T(w) = \{c : I(w, c) > 0\}$. There are parallels between sim_{lin} and sim_{dice} in that both measures compute a ratio between what is shared by the descriptions of both nouns and the sum of the descriptions of each noun. The major difference appears to be the use of MI, and hence we predicted that there would be a close relationship between sim_{lin} and the harmonic mean in the additive MI-based CRM. This relationship is shown below:

$$m_h(\mathcal{P}^{add}_{mi}(w_1, w_2), \mathcal{R}^{add}_{mi}(w_1, w_2)) = \frac{2 \cdot \mathcal{P}^{add}_{mi}(w_1, w_2) \cdot \mathcal{R}^{add}_{mi}(w_1, w_2)}{\mathcal{P}^{add}_{mi}(w_1, w_2) + \mathcal{R}^{add}_{mi}(w_1, w_2)} \tag{42}$$

$$= \frac{2 \cdot \frac{\sum_{TP} I(w_1, c)}{\sum_{F(w_1)} I(w_1, c)} \cdot \frac{\sum_{TP} I(w_2, c)}{\sum_{F(w_2)} I(w_2, c)}}{\frac{\sum_{TP} I(w_1, c)}{\sum_{F(w_1)} I(w_1, c)} + \frac{\sum_{TP} I(w_2, c)}{\sum_{F(w_2)} I(w_2, c)}} \tag{43}$$

$$= \frac{2 \cdot \sum_{TP} I(w_1, c) \cdot \sum_{TP} I(w_2, c)}{\sum_{TP} I(w_2, c) \cdot \sum_{F(w_1)} I(w_1, c) + \sum_{TP} I(w_1, c) \cdot \sum_{F(w_2)} I(w_2, c)} \tag{44}$$

Now if $\sum_{TP} I(w_1, c) = \sum_{TP} I(w_2, c)$, it follows:

$$m_h(\mathcal{P}^{add}_{mi}(w_1, w_2), \mathcal{R}^{add}_{mi}(w_1, w_2)) = \frac{2 \cdot \sum_{TP} I(w_1, c)}{\sum_{F(w_1)} I(w_1, c) + \sum_{F(w_2)} I(w_2, c)} \tag{45}$$

$$= \frac{\sum_{TP} I(w_1, c) + I(w_2, c)}{\sum_{F(w_1)} I(w_1, c) + \sum_{F(w_2)} I(w_2, c)} \tag{46}$$

$$= \frac{\sum_{T(w_1)\cap T(w_2)}(I(w_1, c) + I(w_2, c))}{\sum_{T(w_1)} I(w_1, c) + \sum_{T(w_2)} I(w_2, c)} \text{ since } T(w) = F(w) \tag{47}$$

$$= sim_{lin}(w_1, w_2) \tag{48}$$

Thus, when the additive MI-based model is used, $\gamma = 1$ and the condition $\sum_{TP} I(w_1, c) = \sum_{TP} I(w_2, c)$ holds, the CR framework reduces to sim_{lin}. However, this last necessary condition for equivalence is not one we can expect to hold for many (if any) pairs of words. In order to investigate how good an approximation the harmonic mean is to sim_{lin} in practice, we compared neighbor sets according to each measure using the neighbor set comparison technique outlined earlier.

Figure 2 illustrates the variation in mean similarity between neighbor sets with the parameters β and γ. At $\gamma = 1$, the average similarity between neighbor rankings was 0.967 for high-frequency nouns and 0.923 for low-frequency nouns. This is significantly higher than similarities between other standard similarity measures. However, the optimal approximation of sim_{lin} was found using $\gamma = 0.75$ and $\beta = 0.5$ in the additive MI-based CRM. With these settings, the development set similarity was 0.987 for high-

frequency nouns and 0.977 for low-frequency nouns. This suggests that sim_{lin} allows more compensation for lack of recall by precision and vice versa than the harmonic mean.

4.7 Discussion

We have seen that five of the existing lexical distributional similarity measures are (approximately) equivalent to settings within the CR framework and for one other, a weak approximation can be made. The CR framework, however, more than simulates existing measures of distributional similarity. It defines a space of distributional similarity measures that is already populated with a few named measures. By exploring the space, we can discover the desirable characteristics of distributional similarity measures. It may be that the most useful measure within this space has already been discovered, or it may be that a new optimal combination of characteristics is discovered. The primary goal, however, is to understand how different characteristics relate to high performance in different applications and thus explain why one measure performs better than another.

With this goal in mind, we now turn to the applications of distributional similarity. In the next section, we consider what characteristics of distributional similarity measures are desirable in two different application areas: (1) automatic thesaurus generation and (2) language modeling.

5. Application-Based Evaluation

As discussed by Weeds (2003), evaluation is a major problem in this area of research. In some areas of natural language research, evaluation can be performed against a gold standard or against human plausibility judgments. The first of these approaches is taken by Curran and Moens (2002), who evaluate a number of different distributional similarity measures and weight functions against a gold standard thesaurus compiled from *Roget's*, the *Macquarie* thesaurus, and the Moby thesaurus. However, we argue that this approach can only be considered when distributional similarity is required as an approximation to semantic similarity and that, in any case, it is not ideal since it is not

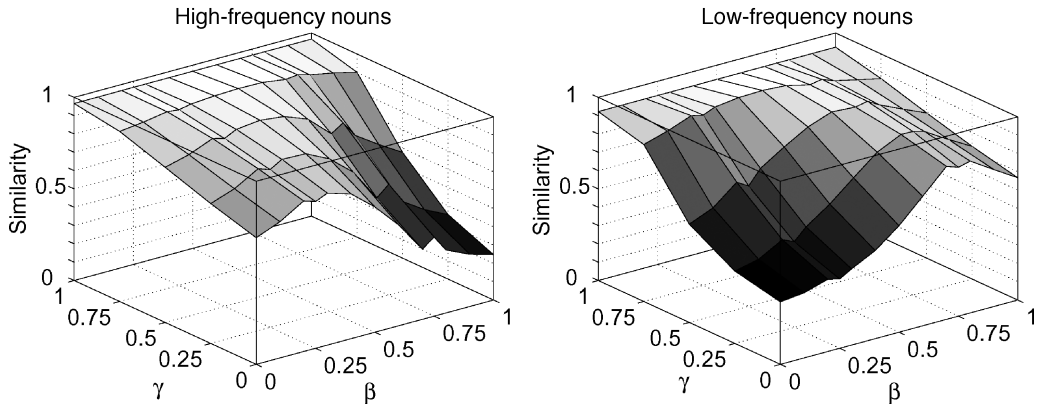

Figure 2
Variation (with parameters β and γ) in development set mean similarity between neighbor sets of the additive MI-based CRM and of sim_{lin}.

clear that there is a single "right answer" as to which words are most distributionally similar. The best measure of distributional similarity will be the one that returns the most useful neighbors in the context of a particular application and thus leads to the best performance in that application. This section investigates whether the desirable characteristics of a lexical distributional similarity measure in an automatic thesaurus generation task (WordNet prediction) are the same as those in a language modeling task (pseudo-disambiguation).

5.1 WordNet Prediction Task

In this section, we evaluate the ability of distributional similarity measures to predict semantic similarity by making comparisons with WordNet. An underlying assumption of this approach is that WordNet is a gold standard for semantic similarity, which, as is discussed by Weeds (2003), is unrealistic. However, it seems reasonable to suppose that a distributional similarity measure that more closely predicts a semantic measure based on WordNet is more likely to be a good predictor of semantic similarity. We chose WordNet as our gold standard for semantic similarity since, as discussed by Kilgarriff and Yallop (2000), distributional similarity scores calculated over grammatical relation level context tend to be more similar to *tighter* thesauri, such as WordNet, than *looser* thesauri such as *Roget's*.

5.1.1 Experimental Set-Up. There are a number of ways to measure the distance between two nouns in the WordNet noun hierarchy (see Budanitsky [1999] for a review). In previous work (Weeds and Weir 2003b), we used the WordNet-based similarity measure first proposed in Lin (1997) and used in Lin (1998a):

$$wn\_sim_{lin}(w_1, w_2) = \max_{c_1 \in S(w_1) \wedge c_2 \in S(w_2)} \left(\max_{c \in sup(c_1) \cap sup(c_2)} \frac{2 \log P(c)}{\log (P(c_1)) + \log (P(c_2))} \right) \quad (49)$$

where $S(w)$ is the set of senses of the word w in WordNet, $\sup(c)$ is the set of possibly indirect super-classes of concept c in WordNet, and $P(c)$ is the probability that a randomly selected word refers to an instance of concept c (estimated over some corpus such as SemCor [Miller et al. 1994]).

However, in other research (Budanitsky and Hirst 2001; Patwardhan, Banerjee, and Pedersen 2003; McCarthy, Koeling, and Weeds 2004), it has been shown that the distance measure of Jiang and Conrath (1997) (referred to herein as the "JC measure") is a superior WordNet-based semantic similarity measure:

$$wn\_dist_{JC}(w_1, w_2) = \max_{c_1 \in S(w_1) \wedge c_2 \in S(w_2)} \left(\max_{c \in sup(c_1) \cap sup(c_2)} 2 \log(c) - \log P(c_1) - \log P(c_2) \right) \quad (50)$$

In our work, we make an empirical comparison of neighbors derived using a WordNet-based measure and each of the distributional similarity measures using the technique discussed in Section 3. We have carried out the same experiments using both the Lin measure and the JC measure. Correlation between distributional similarity measures and the WordNet measure tends to be slightly higher when using the JC measure

Table 5
Optimized similarities between distributional neighbor sets and WordNet derived neighbor sets.

	Noun Frequency							
	high				low			
	Optimal Parameters		Devel Corr.	Test Corr.	Optimal Parameters		Devel Corr.	Test Corr.
Measure	γ	β	(C)	(C)	γ	β	(C)	(C)
sim^{add}_{type}	0.25	0.5	0.323	0.327	0.5	0.25	0.281	0.275
sim^{add}_{tok}	0.25	0.3	0.302	0.310	0.25	0.0	0.266	0.263
sim^{add}_{mi}	0.25	0.2	0.334	**0.342**	0.25	0.2	0.290	**0.283**
sim^{add}_{wmi}	0.25	0.2	0.282	0.293	0.25	0.0	0.274	0.266
sim^{add}_{t}	0.5	0.2	0.330	**0.338**	0.5	0.2	0.292	**0.286**
sim^{add}_{z}	0.5	0.1	0.324	0.332	0.5	0.1	0.280	0.276
sim^{add}_{allr}	0.25	0.2	0.298	0.304	0.25	0.1	0.272	0.267
sim^{dw}_{type}	0.0	0.4	0.306	0.310	0.0	0.0	0.221	0.219
sim^{dw}_{tok}	—	—	0.285	0.294	—	—	0.212	0.211
sim^{dw}_{mi}	0.0	0.2	0.324	0.333	0.0	0.0	0.266	0.261
sim^{dw}_{wmi}	0.0	0.1	0.273	0.281	0.0	0.1	0.223	0.220
sim^{dw}_{t}	0.5	0.2	0.328	0.333	0.5	0.3	0.289	0.282
sim^{dw}_{z}	0.5	0.1	0.324	0.329	0.5	0.2	0.280	0.276
sim^{dw}_{allr}	0.75	0.2	0.263	0.265	0.75	0.0	0.226	0.225
sim_{dice}			0.295	0.299			0.123	0.123
sim_{jacc}			0.295	0.299			0.123	0.123
$dist_{L1}$			0.285	0.294			0.212	0.211
$dist_{\alpha}$			0.310	0.317			0.289	0.281
sim_{hind}			0.320	0.326			0.267	0.261
sim_{lin}			0.313	0.323			0.192	0.186
$wnsim_{lin}$			0.907	0.907			0.884	0.883

(percentage increase in similarity of approximately 10%), but the relative differences between distributional similarity measures remain approximately the same. Here, for brevity, we present results just using the JC measure.

5.1.2 Results. As before, we present the results separately for the 1,000 high-frequency target nouns and for the 1,000 low-frequency target nouns. Table 5 shows the optimal parameter settings for each CRM (computed over the development set) and the mean similarities with the JC measure at these settings in both the development set and the test set. It also shows the mean similarities over the development set and the test set for each of the existing similarity measures discussed in Section 4. For reference, we also present the mean similarity for the WordNet-based measure wn_sim_{lin}. For ease

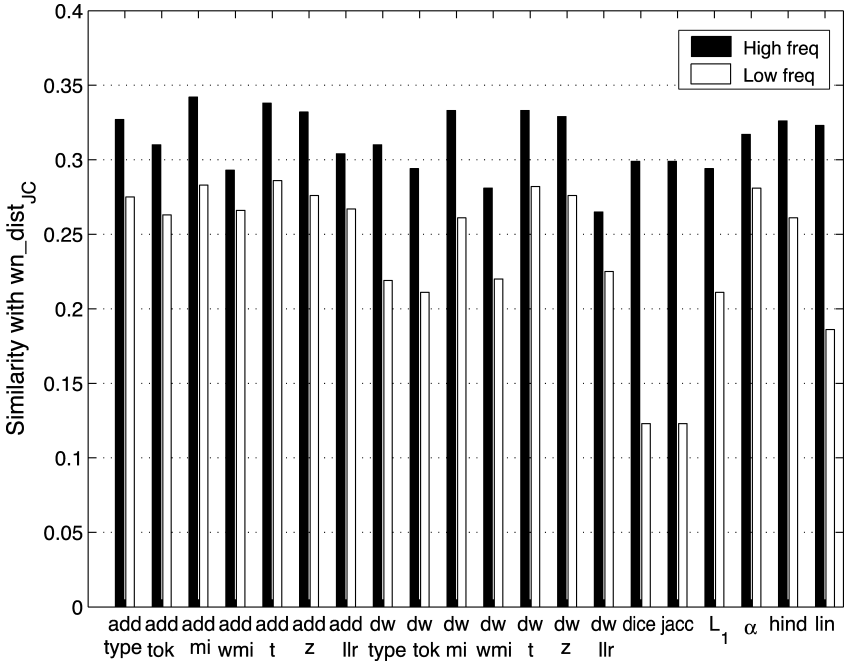

Figure 3
Bar chart illustrating test set similarity with WordNet for each distributional similarity measure.

of comparison, the test set correlation values for each distributional measure are also illustrated in Figure 3.

We would expect a mean overlap score of 0.08 by chance. Standard deviations in the observed test set mean similarities were all less than 0.1, and thus any difference between mean scores of greater than 0.016 is significant at the 99% level, and differences greater than 0.007 are significant at the 90% level. Thus, from the results in Table 5 we can make the following observations.

First, the best-performing distributional similarity measures, in terms of WordNet prediction, for both high- and low-frequency nouns, are the MI-based and the t-test based CRMs. The additive MI-based CRM performs the best for high-frequency nouns and the additive t-test based CRM performs the best for low-frequency nouns. However, the differences between these models are not statistically significant. These CRMs perform substantially better than all of the unparameterized distributional similarity measures, of which the best performing are sim_{hind} and sim_{lin} for high-frequency nouns and $dist_{\alpha 1}$ for low-frequency nouns. Second, the difference-weighted versions of each model generally perform slightly worse than their additive counterparts. Thus, the difference in extent to which each word occurs in each context does not appear to be a factor in determining semantic similarity. Third, all of the measures perform significantly better for high-frequency nouns than for low-frequency nouns. However, some of the measures (sim_{lin}, sim_{jacc} and sim_{dice}) perform considerably worse for low-frequency nouns.

We now consider the effects of β and γ in the CRMs on performance. The pattern of variation across the CRMs was very similar. This pattern is illustrated using one of the best-performing CRMs (sim^{add}_{mi}) in Figure 4. With reference to this figure and to the results for the other models (not shown), we make the following observations.

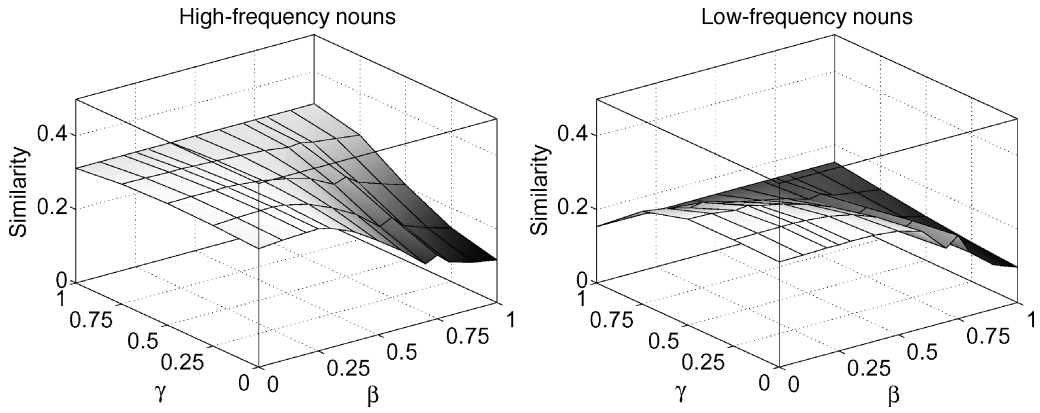

Figure 4
Variation in similarity with WordNet with respect to β and γ for the additive MI-based CRM.

First, for high- and low-frequency nouns, similarity with WordNet is higher for low values of β than for high values of β. In other words, neighbors according to the WordNet based measure tend to have high-recall retrieval of the target noun's co-occurrences. Second, a high value of γ leads to high performance for high-frequency nouns but poor performance for low-frequency nouns. This suggests that WordNet-derived neighbors of high-frequency target nouns also have high-precision retrieval of the target noun's co-occurrences, whereas the WordNet-derived neighbors of low-frequency target nouns do not. This also explains why particular existing measures (Jaccard's / the Dice Coefficient and Lin's Measure), which are very similar to a $\gamma = 1$ setting in the CR framework, perform well for high-frequency nouns but poorly for low-frequency nouns.

5.1.3 Discussion. Our results in this section are comparable to those of Curran and Moens (2002), who showed that combining the t-test with Jaccard's coefficient outperformed combining MI with Jaccard's coefficient by approximately 10% in a comparison against a gold-standard thesaurus. However, we do not find a significant difference between using the t-test and MI in similarity calculation. Further, we found that using a combination of precision and recall weighted towards recall performs substantially better than using the harmonic mean (which is equivalent to Jaccard's measure). In our experiments, the development-set similarity using the harmonic mean in the additive MI-based CRM was 0.312 for high-frequency nouns and 0.153 for low-frequency nouns, and the development-set similarity using the harmonic mean in the additive t-test based CRM was 0.294 for high-frequency nouns and 0.129 for low-frequency nouns.

5.2 Pseudo-Disambiguation Task

Pseudo-disambiguation tasks have become a standard evaluation technique (Gale, Church, and Yarowsky 1992; Schütze 1992; Pereira, Tishby, and Lee 1993; Schütze 1998; Lee 1999; Dagan, Lee, and Pereira 1999; Golding and Roth 1999; Rooth et al. 1999; Even-Zohar and Roth 2000; Lee 2001; Clark and Weir 2002) and, in the current setting, we may use a noun's neighbors to decide which of two co-occurrences is the most likely. Although pseudo-disambiguation is an artificial task, it has relevance in at least two application areas. First, by replacing occurrences of a particular word in a test suite with

a pair of words from which a technique must choose, we recreate a simplified version of the word sense disambiguation task; that is, we choose between a fixed number of homonyms based on local context. The second is in language modeling where we wish to estimate the probabilities of co-occurrences of events but, due to the sparse data problem, it is often the case that a possible co-occurrence has not been seen in the training data.

5.2.1 Experimental Set-up. A typical approach to performing pseudo-disambiguation is as follows. A large set of noun-verb direct-object pairs is extracted from a corpus, of which a portion is used as test data and another portion is used as training data. The training data can be used to construct a language model and/or determine the distributionally nearest neighbors of each noun. Noun-verb pairs (n, v_1) in the test data are replaced with noun-verb-verb triples (n, v_1, v_2) and the task is to decide which of the two verbs is the most likely to take the noun as its direct object. Performance is usually measured as error rate. We will now discuss the details of our own experimental set-up.

As already discussed (Section 3), 80% of the noun-verb direct-object data extracted from the BNC for each of 2,000 nouns was used to compute the similarity between nouns and is also used as the language model in the pseudo-disambiguation task, and 20% of the data was set aside as test data, providing unseen co-occurrences for this pseudo-disambiguation task.

In order to construct the test set from the test data, we took all[10] of the test data set aside for each target noun and modified it as follows. We converted each noun-verb pair (n, v_1) in the test data into a noun-verb-verb triple (n, v_1, v_2). v_2 was randomly selected from the verbs that have the same frequency, calculated over all the training data, as v_1 plus or minus 1. If there are no other verbs within this frequency range, then the test instance is discarded. This method ensures that there is no systematic bias towards v_2 being of a higher or lower frequency than v_1. We also ensured that (n, v_2) has not been seen in the test or training data. Ten test instances[11] were then selected for each target noun in a two-step process of (1) while more than ten triples remained, discarding duplicate triples and (2) randomly selecting ten triples from those remaining after step 1. At this point, we have 10,000 test instances pertaining to high-frequency nouns and 10,000 test instances pertaining to low-frequency nouns, and there are no biases towards the higher-frequency or lower-frequency nouns within these sets. Each of these sets was split into five disjoint subsets, each containing two instances for each target noun. We use these five subsets in two ways. First, we perform five-fold cross validation. In five-fold cross validation, we compute the optimal parameter settings in four of the subsets and the error rate at this optimal parameter setting in the remaining subset. This is repeated five times with a different subset held out each time. We then compute an average optimal error rate. We cannot, however, compute an average optimal parameter setting, since this would assume a convex relationship between parameter settings and error rate. In order to study the relationship between parameter settings and error rate, we combine three of the sets to form a development set and two of the sets to form a test set. The development set is used to optimize parameters and the test set

10 Unlike Lee (1999), we do not delete instances from the test data that occur in the training data. This is discussed in detail in (Weeds 2003), but our main justification for this approach is that a single co-occurrence of (n, v_1) compared to zero co-occurrences of (n, v_2) is not necessarily sufficient evidence to conclude that the population probability of (n, v_1) is greater than that of (n, v_2).

11 Ten being less than the minimum number (14) of (possibly) indistinct co-occurrences for any target noun in the original test data.

to determine error rates at the optimal settings. In graphs showing the relationship between error rate and parameter settings, it is the error rate in this development set that is shown. In the case of the CRMs, the parameters that are optimized are β, γ, and k (the number of nearest neighbors).[12] For the existing measures, the only parameter to be optimized is k.

Having constructed the test sets, the task is to take each test instance (n, v_1, v_2) and use the nearest neighbors of noun n (as computed from the training data) to decide which of (n, v_1) and (n, v_2) was the original co-occurrence. Each of n's neighbors, m, is given a vote that is equal to the difference in frequencies of the co-occurrences (m, v_1) and (m, v_2) and that it casts to the verb with which it occurs most frequently. Thus, we distinguish between cases where a neighbor occurs with each verb approximately the same number of times and where a neighbor occurs with one verb significantly more often than the other. The votes for each verb are summed over all of the k nearest neighbors of n, and the verb with the most votes wins. Performance is measured as error rate.

$$\text{Error rate} = \frac{1}{T}\left(\#\text{ of incorrect choices } + \frac{\#\text{ of ties}}{2}\right) \tag{51}$$

where T is the number of test instances and a tie results when the neighbors cannot decide between the two alternatives.

5.2.2 Results. In this section, we present results on the pseudo-disambiguation task for all of the CRMs described in Section 2. We also compare the results with the six existing distributional similarity measures (Section 4) and the two WordNet-based measures (Section 5.1).

A baseline for these experiments is the performance obtained by a technique that backs-off to the unigram probabilities of the verbs being disambiguated. By construction of the test set, this should be approximately 0.5. The actual empirical figures are 0.553 for the high-frequency noun test set and 0.586 for the low-frequency noun test set. The deviation from 0.5 is due to the unigram probabilities of the verbs not being exactly equal and to their being calculated over a larger data set than just the training data for the 2,000 target nouns. These baseline error-rates are also different from what is observed when all 1,999 potential neighbors are considered. In this case, we obtain an error rate of 0.6885 for the high-frequency noun test set and 0.6178 for the low-frequency noun test set. These differences are due to the fact that the correct choice verb, but not the incorrect choice verb, has occurred, possibly many times, with the target noun in the training data, but a noun is not considered as a potential neighbor of itself.

The results are summarized in Table 6. The table gives the average optimal error rates for each measure, and for high- and low-frequency nouns, calculated using five-fold cross validation. For ease of comparison, the cross-validated average optimal error rates are illustrated in Figure 5. Standard deviation in the mean optimal error rate across the five folds was always less than 0.15 and thus differences greater than 0.028 are significant at the 99% level and differences greater than 0.012 are significant at the 90% level. From the results, we make the following observations.

12 We also experimented with optimizing a similarity threshold t, but found that optimizing k gave better results (Weeds 2003).

Table 6
Mean optimal error rates using five-fold cross-validation (when optimizing k, γ and β).

	Noun Frequency		Measure	Noun Frequency	
Measure	high	low		high	low
sim^{add}_{type}	0.196	0.197	sim^{dw}_{type}	0.214	0.185
sim^{add}_{tok}	0.219	0.241	sim^{dw}_{tok}	0.234	0.202
sim^{add}_{mi}	0.178	**0.169**	sim^{dw}_{mi}	0.187	0.176
sim^{add}_{wmi}	0.173	0.192	sim^{dw}_{wmi}	0.171	0.192
sim^{add}_{t}	**0.154**	0.172	sim^{dw}_{t}	0.163	0.186
sim^{add}_{z}	0.164	0.183	sim^{dw}_{z}	0.167	0.193
sim^{add}_{allr}	0.170	0.211	sim^{dw}_{allr}	0.171	0.215
sim_{dice}	0.215	0.204	sim_{jacc}	0.215	0.204
$dist_{L_1}$	0.234	0.202	$dist_{\alpha 1}$	0.230	0.192
sim_{hind}	0.201	0.18	sim_{lin}	0.193	0.181
wn_sim_{lin}	0.295	0.294	wn_dist_{JC}	0.302	0.295
baseline	0.553	0.586			

First, the best measure appears to be the additive t-test based CRM. This significantly outperforms all but one (the z-test based CRM) of the other measures for high-frequency nouns. For low-frequency nouns, slightly higher performance is obtained using the additive MI-based CRM. This difference, however, is not statistically significant. Second, all of the distributional similarity measures perform considerably better than the WordNet-based measures[13] at this task for high- and low-frequency nouns. Third, for many measures, performance over high-frequency nouns is not significantly higher (and is in some cases lower) than over low-frequency nouns. This suggests that distributional similarity can be used in language modeling even when there is relatively little corpus data over which to calculate distributional similarity.

We now consider the effects of the different parameters on performance. Since we use the development set to determine the optimal parameters, we consider performance on the development set as each parameter is varied. Table 7 shows the optimized parameter settings in the development set, error rate at these settings in the development set, and error rate at these settings in the test set. For the CRMs, we considered how the performance varies with each parameter when the other parameters are held constant at their optimum values. Figure 6 shows how performance varies with β, and Figure 7 shows how performance varies with γ for the additive and difference-weighted t-test based and MI-based CRMs. For reference, the optimal error rates for the best performing existing distributional similarity measure (sim_{lin}) is also shown as a straight line on each graph.

We do not show the variation with respect to k for any of the measures, but this was fairly similar for all measures and is as would be expected. To begin with, considering

13 However, for this task, in contrast to earlier work, wn_sim_{lin} gives slightly, although insignificantly, better performance than wn_dist_{JC}.

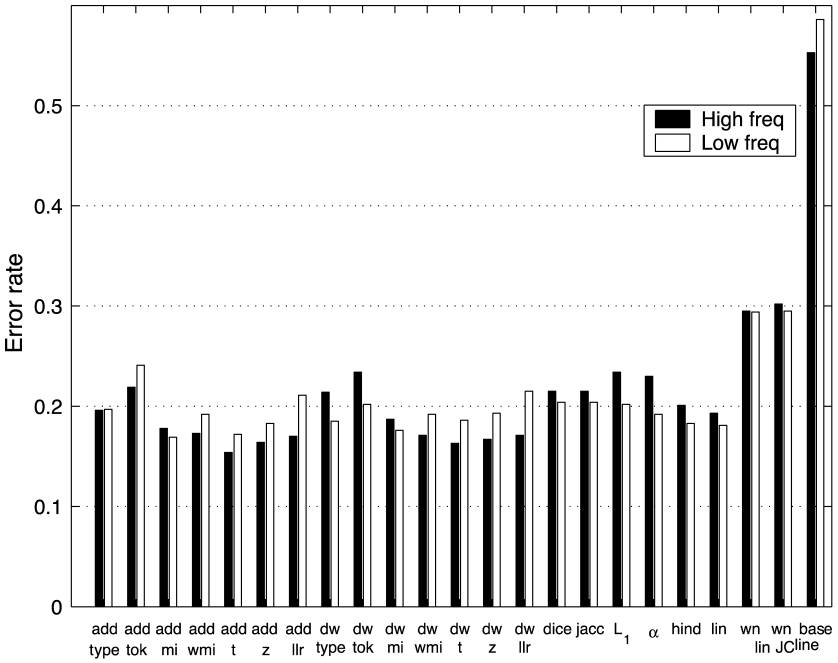

Figure 5
Bar chart illustrating cross-validated optimal error rates for each measure when k is optimised.

more neighbors increases performance, since more neighbors allow decisions to be made in a greater number of cases. However, when k increases beyond an optimal value, a greater number of these decisions will be in the wrong direction, since these words are not very similar to the target word, leading to a decrease in performance. In a small number of cases (when using the ALLR-based CRMs or the WMI-based CRMs for high frequency nouns), performance peaks at $k = 1$. This suggests that these measures may be very good at finding a few very close neighbors.

The majority of models, including the additive t-test based and additive MI-based CRMs, perform significantly better at low values of γ (0.25-0.5) and high values of β (around 0.8). This indicates that a potential neighbor with high-precision retrieval of informative features is more useful than one with high-recall retrieval. In other words, it seems that it is better to sacrifice being able to make decisions on every test instance with a small number of neighbors in favor of not having neighbors that predict incorrect verb co-occurrences. This also suggests why we saw fairly low performance by the α-skew divergence measure on this task, since it is closest to a high-recall setting in the additive t-test based model. The low values of γ indicate that a combination of precision and recall that is closer to a weighted arithmetic mean is generally better than one that is closer to an unweighted harmonic mean. However, this does not hold for the t-test based CRMs for low-frequency nouns. Here a higher value of γ is optimal, indicating that, in this case, requiring both recall and precision results in high performance.

6. Conclusions and Future Directions

Our main contribution is the development of a framework, first presented in a preliminary form in Weeds and Weir (2003b), that is based on the concept of lexical substi-

Table 7
Summary of results on pseudo-disambiguation task when optimizing β, γ and k.

	Noun Frequency									
	high					low				
	Optimal Parameters			Devel. Error	Test Error	Optimal Parameters			Devel. Error	Test Error
Measure	γ	β	k			γ	β	k		
sim^{add}_{type}	0.25	0.8	150	0.193	0.193	0.25	0.75	100	0.192	0.200
sim^{add}_{tok}	0	0.8	250	0.211	0.224	0.5	0.1	130	0.234	0.233
sim^{add}_{mi}	0.25	0.8	170	0.175	0.186	0.5	0.8	120	0.169	0.178
sim^{add}_{wmi}	0.0	1.0	1	0.175	0.169	0.75	0.0	100	0.183	0.182
sim^{add}_{t}	0.25	0.8	190	0.153	0.155	0.5	0.7	110	0.165	0.176
sim^{add}_{z}	0.25	0.7	40	0.165	0.163	0.5	1.0	250	0.174	0.188
sim^{add}_{allr}	0.0	0.9	1	0.170	0.169	0.25	0.6	90	0.204	0.210
sim^{dw}_{type}	0	0.6	50	0.208	0.215	0.25	0.3	190	0.177	0.188
sim^{dw}_{tok}	n/a	n/a	60	0.227	0.234	n/a	n/a	50	0.194	0.206
sim^{dw}_{mi}	0.25	0.8	100	0.181	0.193	0.5	0.7	160	0.172	0.173
sim^{dw}_{wmi}	0.0	0.0	1	0.172	0.170	0.25	0.1	450	0.183	0.190
sim^{dw}_{t}	0.5	0.8	120	0.156	0.165	0.75	0.6	250	0.179	0.187
sim^{dw}_{z}	0.5	0.7	50	0.166	0.171	0.75	0.9	400	0.187	0.199
sim^{dw}_{allr}	0.0	0.9	1	0.171	0.169	0.5	1.0	180	0.208	0.212
sim_{lin}	n/a	n/a	50	0.190	0.199	n/a	n/a	80	0.179	0.186

tutability. Here, we cast the problem of measuring distributional similarity as one of *co-occurrence retrieval* (CR), for which we can measure precision and recall by analogy with the way they are measured in document retrieval. This CR framework has then allowed us to systematically explore various characteristics of distributional similarity measures.

First, we asked whether lexical substitutability is necessarily symmetric. To this end, we have explored the merits of symmetry and asymmetry in a similarity measure by varying the relative importance attached to precision and recall. We have seen that as the distribution of word B moves away from being identical to that of word A, its *similarity* with A can decrease along one or both of two dimensions. When B occurs in contexts that word A does not, precision is lost but B may remain a high-recall neighbor of word A. When B does not occur in contexts that A does, recall is lost but B may remain a high-precision neighbor of word A. Through our experimental work, which is more thorough than that presented in Weeds and Weir (2003b), we have shown that the kind of neighbor preferred appears to depend on the application in hand. High-precision neighbors were more useful in the language modeling task of pseudo-disambiguation and high-recall neighbors were more highly correlated with WordNet-derived neighbor sets. Thus, similarity appears to be inherently asymmetric. Further, it would seem

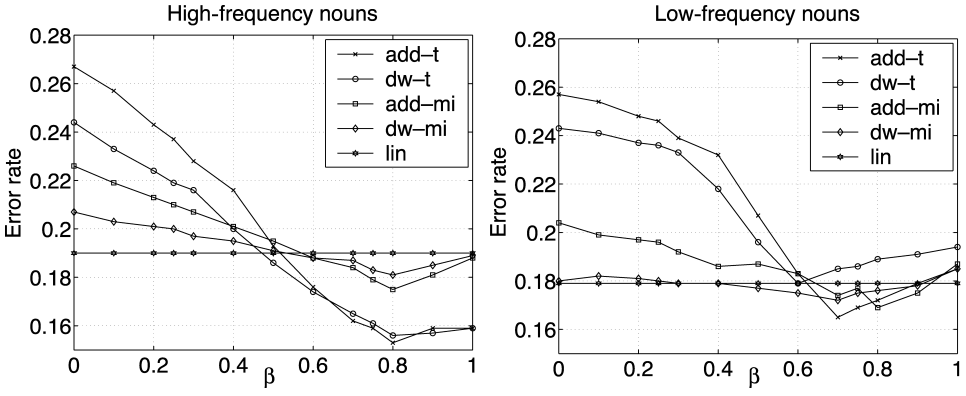

Figure 6
Performance of CRMs with respect to β (at optimal values of k and γ).

unlikely that any single, unparameterized measure of distributional similarity would be able to do better on both tasks.

Second, we asked whether all contexts are equally important in the calculation of distributional similarity. To this end, we have explored the way in which frequency information is utilized using different co-occurrence retrieval models (CRMs). Using different weight functions, we have investigated the relative importance of different co-occurrence types. In earlier work (Weeds and Weir 2003b), we saw that using MI to weight features gave improved performance on the two evaluation tasks over type-based or token-based CRMs. Here, we have seen that further gains can be made by using the t-test as a weight function. This leads to significant improvements on the pseudo-disambiguation task for all nouns and marginal improvements on the WordNet prediction task for low-frequency nouns. To some extent, this supports the findings of Curran and Moens (2002), who investigated a number of weight functions for distributional similarity and showed that the t-test performed better than a number of other weight functions including MI.

Third, we asked whether it is necessary to consider the difference in extent to which each word appears in each context. To this end, we have herein proposed difference-

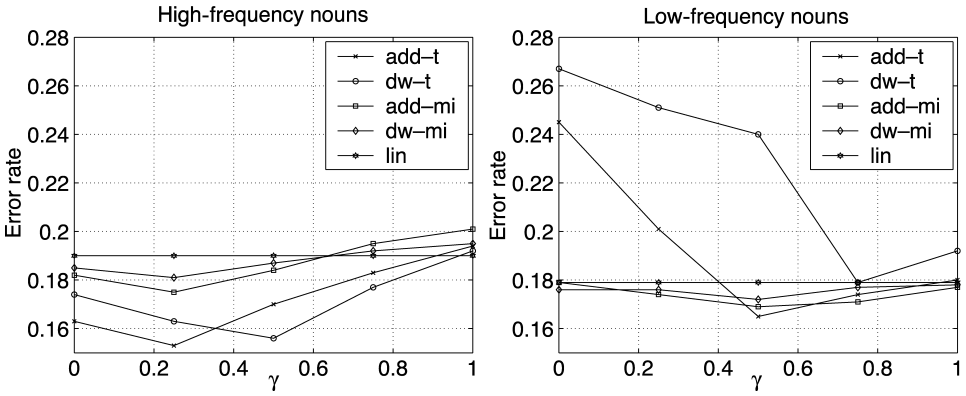

Figure 7
Performance of CRMs with respect to γ (at optimal values of k and β).

weighted versions of each model in which the similarity of two words in respect of an individual feature is defined using the same principles that we use to define the similarity of two words in respect of all their features. We have compared these difference-weighted CRMs to their additive counterparts and shown that difference-weighting does not seem to be a major factor and does not improve results when using the best-performing CRMs.

Another important contribution of this work on co-occurrence retrieval is a better understanding of existing distributional similarity measures. By comparing existing measures with the CR framework, we can analyze their CR characteristics. As discussed in Weeds and Weir (2003b), the Dice Coefficient and Jaccard's Coefficient are exactly simulated by $\gamma = 1$ in the additive type-based model and Lin's Measure is almost equivalent to the harmonic mean of precision and recall in the additive MI-based model. Here, we also show that the L_1 Norm is exactly simulated by the (unparameterized) difference-weighted token-based model, Hindle's Measure is exactly simulated by $\gamma = 0, \beta = 0$ in the additive MI-based model, and the α-skew divergence measure is most similar to high-recall settings in the additive t-test based CRM. Knowing that Lin's Measure is almost equivalent to the harmonic mean of precision and recall in the additive MI-based model explains why this measure does badly on the WordNet prediction task for low-frequency nouns. We have seen that recall is more important than precision in the WordNet prediction task, whereas the nearest neighbors of a target noun according to Lin's Measure have both high precision and high recall. Conversely, knowing that the α-skew divergence measure is most closely approximated by high-recall settings in the additive t-test based model explains why this measure performs poorly on the pseudo-disambiguation task, since we have seen that high precision is required for optimal performance on this task.

Finally, our evaluation of measures has been performed over a set of 2,000 nouns, and we have shown that the performance of distributional similarity techniques for low-frequency nouns is not significantly lower than for high-frequency nouns. This suggests that distributional techniques might be used even when there is relatively little data available. In the distributional domain, this means that we can use probability estimation techniques for rare words with greater confidence. In the semantic domain, we might be able to use distributional techniques to extend existing semantic resources to cover rare or new words or automatically generate domain-, genre-, or dialect-specific resources.

There are a number of major directions in which this work can be extended. First, although the set of CRMs defined here is more extensive than that defined in Weeds and Weir (2003b), it is still not exhaustive, and other models might be proposed. Further, it would be interesting to combine CRMs with the feature reweighting scheme of Geffet and Dagan (2004). These authors compare distributional similarity scores with human judgments of semantic entailment and show that substantial (approximately 10%) improvements over using Lin's Measure can be achieved by first calculating similarity using Lin's Measure and then recalculating similarity using a relative feature focus score, which indicates how many of a word's nearest neighbors shared that feature.

Second, there are other potential application-based tasks that could be used to evaluate CRMs and distributional similarity methods in general. In particular, we see potential for the use of distributional similarity methods in prepositional phrase attachment ambiguity resolution. This task has been previously tackled using semantic classes to predict what is ultimately distributional information. Accordingly, we believe that it should be possible to do better using the CR framework.

Finally, in order to be able to truly rival manually generated thesauri, distributional techniques need to be able to distinguish between different semantic relations such as synonymy, antonymy, and hyponymy. These are important linguistic distinctions, particularly in the semantic domain, since we are unlikely, say, to want to replace a word with its antonym. Weeds, Weir, and McCarthy (2004) give preliminary results on the the use of precision and recall to distinguish between hypernyms and hyponyms in sets of distributionally related words.

Acknowledgments

This research was supported by an Engineering and Physical Sciences Research Council (EPSRC) studentship to the first author. The authors would like to thank John Carroll, Mirella Lapata, Adam Kilgarriff, Bill Keller, Steve Clark, James Curran, Darren Pearce, Diana McCarthy, and Mark McLauchlan for helpful discussions and insightful comments throughout the course of the research. We would also like to thank the anonymous reviewers of this paper for their comments and suggestions.

References

Bernard, John R. L., editor. 1990. *The Macquarie Encyclopedic Thesaurus*. The Macquarie Library, Sydney, Australia.

Briscoe, Edward and John Carroll. 1995. Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels. In *Proceedings of the 4th ACL/SIGDAT International Workshop on Parsing Technologies*, pages 48–58, Cambridge, MA.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Budanitsky, Alexander. 1999. *Lexical Semantic Relatedness and its Application in Natural Language Processing*. Ph.D. thesis, University of Toronto, Ontario.

Budanitsky, Alexander and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL-01 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA.

Caraballo, Sharon. 1999. Automatic construction of a hypernym-labelled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 120–126, College Park, MA.

Carroll, John and Edward Briscoe. 1996. Apportioning development effort in a probabilistic LR parsing system through evaluation. In *Proceedings of the ACL/SIGDAT Conference on Empirical Methods in Natural Language Processing (EMNLP96)*, pages 92–100, Santa Cruz, CA.

Church, Kenneth W., William Gale, Patrick Hanks, Donald Hindle, and Rosamund Moon. 1994. Lexical substitutability. In B. T. S. Atkins and A. Zampolli, editors. *Computational Approaches to the Lexicon*. Oxford University Press, Oxford, pages 153–177.

Church, Kenneth W. and Patrick Hanks. 1989. Word association norms, mutual information and lexicography. In *Proceedings of the 27th Annual Conference of the Association for Computational Linguistics (ACL-89)*, pages 76–82, Vancouver, British Columbia.

Clark, Stephen and David Weir. 2000. A class-based probabilistic approach to structural disambiguation. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-00)*, pages 194–200, Saarbrucken, Germany.

Clark, Stephen and David Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Cover, T. M. and J. A. Thomas. 1991. *Elements of Information Theory*. Wiley, New York.

Curran, James R. and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-SIGLEX Workshop on Unsupervised Lexical Acquisition*, pages 59–67, Philadelphia, PA.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1999. Similarity-based models of word co-occurrence probabilities. *Machine Learning Journal*, 34(1–3):43–69.

Dagan, Ido, S. Marcus, and S. Markovitch. 1993. Contextual word similarity and estimation from sparse data. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 56–63, Columbus, OH.

Even-Zohar, Yair and Dan Roth. 2000. A classification approach to word prediction.

In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-00)*, pages 124–131, Pittsburg, PA.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Firth, John Rupert. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, pages 1–32, Philological Society, Oxford. Reprinted in Palmer, F. (ed.), 1968 *Selected Papers of J. R. Firth*, Longman, Harlow.

Fontenelle, Thierry, Walter Bruls, Luc Thomas, Tom Vanallemeersch, and Jacques Jansen. 1994. DECIDE, MLAP-Project 93-19, deliverable D-1a: a survey of collocation extraction tools. Technical report, University of Liege, Belgium.

Frakes, W. B. and R. Baeza-Yates, editors. 1992. *Information Retrieval, Data Structures and Algorithms*. Prentice Hall, New York.

Fung, Pascale and Kathleen McKeown. 1997. A technical word- and term-translation aid using noisy parallel corpora across language groups. *Machine Translation*, 12(1–2):53–87.

Gale, William, Kenneth W. Church, and David Yarowsky. 1992. Work on statistical methods for word sense disambiguation. In *Working notes of the AAAI symposium on probabilistic approaches to natural language*, pages 54–60, Menlo Park, CA.

Geffet, Maayan and Ido Dagan. 2004. Feature vector quality and distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 247–253, Geneva, Switzerland.

Golding, Andrew R. and Dan Roth. 1999. A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1–3):182–190.

Grefenstette, Gregory. 1994. Corpus-derived first-, second- and third-order word affinities. In *Proceedings of Euralex*, pages 279–290, Amsterdam, Holland.

Harris, Zelig S. 1968. *Mathematical Structures of Language*. John Wiley, New York.

Hindle, Donald. 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL-1990)*, pages 268–275, Pittsburgh, PA.

Hindle, Donald and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan.

Kaufman, Leonard and Peter J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, New York.

Kilgarriff, Adam. 2003. Thesauruses for natural language processing. In *Proceedings of the Joint Conference on Natural Language Processing and Knowledge Engineering*, pages 5–13, Beijing, China.

Kilgarriff, Adam and Colin Yallop. 2000. What's in a thesaurus. In *Second Conference on Language Resources and Evaluation (LREC-00)*, pages 1371–1379, Athens.

Kullback, S. and R.A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-1999)*, pages 23–32, College Park, MA.

Lee, Lillian. 2001. On the effectiveness of the skew divergence for statistical language analysis. *Artificial Intelligence and Statistics*, pages 65–72.

Li, Hang. 2002. Word clustering and disambiguation based on co-occurrence data. *Natural Language Engineering*, 8(1):25–42.

Li, Hang and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL-97)*, pages 64–71, Madrid, Spain.

Lin, Dekang. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL '98)*, pages 768–774, Montreal, Quebec.

Lin, Dekang. 1998b. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, Madison, WI.

Lin, Dekang, Shaojun Zhao, Lijuan Qin, and Ming Zhou. 2003. Identifying synonyms

among distributionally similar words. In *In Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1492–1493.
Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.
McCarthy, Diana, Rob Koeling, and Julie Weeds. 2004. Ranking WordNet senses. Technical Report 569, Department of Informatics, University of Sussex, Brighton.
McCarthy, Diana, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 280–287, Barcelona, Spain.
Miller, G., M. Chodorow, S. Landes, C. Leacock, and R. Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, NJ.
Pantel, Patrick and Dekang Lin. 2000. Word-for-word glossing of contextually similar words. In *Proceedings of the Conference on Applied Natural Language Processing / 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL-00)*, pages 78–85, Seattle, WA.
Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 241–257, Mexico City.
Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of similar words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 183–190, Columbus, OH.
Pustejovsky, James. 1995. *The generative lexicon*. MIT Press, Cambridge, MA.
Resnik, Philip. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.
Roget, Peter. 1911. *Thesaurus of English Words and Phrases*. Longmans, Green and Co., London.
Rooth, Mats, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pages 104–111, College Park, MA.
Salton, Gerald and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.
Schütze, Hinrich. 1992. Dimensions of meaning. In *Proceedings of Conference on Supercomputing*, pages 787–796, Minneapolis, MN.
Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.
van Rijsbergen, C. J. 1979. *Information Retrieval, second edition*. Butterworths, London.
Weeds, Julie. 2003. *Measures and Applications of Lexical Distributional Similarity*. Ph.D. thesis, University of Sussex, Brighton.
Weeds, Julie and David Weir. 2003a. Finding and evaluating sets of nearest neighbours. In *Proceedings of the 2nd International Conference on Corpus Linguistics*, pages 879–888, Lancaster, UK.
Weeds, Julie and David Weir. 2003b. A general framework for distributional similarity. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-03)*, pages 81–88, Sapporo, Japan.
Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*, pages 1015–1021, Geneva, Switzerland.
Xu, Jinxi and Bruce Croft. 1996. Query expansion using local and global disambiguation. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-96)*, pages 4–11, Zurich, Switzerland.