1

1

Bootstrapping Distributional Feature Vector Quality

Maayan Zhitomirsky-Geffet**
Bar-Ilan University

Ido Dagan†
Bar-Ilan University

This article presents a novel bootstrapping approach for improving the quality of feature vector weighting in distributional word similarity. The method was motivated by attempts to utilize distributional similarity for identifying the concrete semantic relationship of lexical entailment. Our analysis revealed that a major reason for the rather loose semantic similarity obtained by distributional similarity methods is insufficient quality of the word feature vectors, caused by deficient feature weighting. This observation led to the definition of a bootstrapping scheme which yields improved feature weights, and hence higher quality feature vectors. The underlying idea of our approach is that features which are common to similar words are also most characteristic for their meanings, and thus should be promoted. This idea is realized via a bootstrapping step applied to an initial standard approximation of the similarity space. The superior performance of the bootstrapping method was assessed in two different experiments, one based on direct human gold standard annotation and the other based on an automatically created disambiguation dataset. These results are further supported by applying a novel quantitative measurement of the quality of feature weighting functions. Improved feature weighting also allows massive feature reduction, which indicates that the most characteristic features for a word are indeed concentrated at the top ranks of its vector. Finally, experiments with three prominent similarity measures and two feature weighting functions showed that the bootstrapping scheme is robust and is independent of the original functions over which it is applied.

1. Introduction

1.1 Motivation

Distributional word similarity has long been an active research area (Hindle 1990; Ruge 1992; Grefenstette 1994; Lee 1997; Lin 1998; Dagan, Lee, and Pereira 1999; Weeds and Weir 2005). This paradigm is inspired by Harris' distributional hypothesis (Harris 1968), which states that semantically similar words tend to appear in similar contexts. In a computational realization, each word is characterized by a weighted feature vector, where features typically correspond to other words that co-occur with the characterized word in the context. Distributional similarity measures quantify the degree of similarity between a pair of such feature vectors. It is then assumed that two words that occur

** Department of Information Science, Bar- Ilan University, Ramat-Gan, Israel. E-mail:zhitomim@mail.biu.ac.il
† Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel. E-mail:dagan@cs.biu.ac.il

Submission received: 6 December 2006; Revised submission received: 9 July 2008; Accepted for publication: 21 November 2008

© 2005 Association for Computational Linguistics

within similar contexts, as measured by similarity of their context vectors, are indeed semantically similar.

The distributional word similarity measures were often applied for two types of inferences. The first type is making similarity-based generalizations for smoothing word co-occurrence probabilities, in applications such as language modeling and disambiguation. For example, assume that we need to estimate the likelihood of the verb-object co-occurrence pair *visit–country*, while it did not appear in our sample corpus. Yet, co-occurrences of the verb *visit* with words that are distributionally similar to *country*, such as *state*, *city*, and *region* do appear in the corpus. Consequently, we may infer that *visit–country* is also a plausible expression, using some mathematical scheme of similarity-based generalization (Essen and Steinbiss 1992; Dagan, Marcus, and Markovitch 1995; Karov and Edelman 1996; Ng 1997; Ng and Lee 1996; Dagan, Lee, and Pereira 1999; Lee 1999; Weeds and Weir 2005). The rationale behind this inference is that if two words are distributionally similar then the occurrence of one word in some contexts indicates that the other word is also likely to occur in such contexts.

A second type of semantic inference, which primarily motivated our own research, is meaning-preserving lexical substitution. Many NLP applications, such as Question Answering, Information Retrieval, Information Extraction, and (multi-document) summarization need to recognize that one word can be substituted by another one in a given context while preserving, or entailing the original meaning. Naturally, recognizing such substitutable lexical entailments is a prominent component within the textual entailment recognition paradigm, which models semantic inference as an application independent task (Dagan, Glickman, and Magnini 2006). Accordingly, several textual entailment systems did utilize the output of distributional similarity measures to model entailing lexical substitutions (Jijkoun and de Rijke 2005; Ferrandez et al. 2006; Vanderwende, Menezes, and Snow 2006; Nicholson, Stokes, and Baldwin 2006; Adams 2006). In some of these papers the distributional information typically complements manual lexical resources in textual entailment systems, most notably WordNet (Fellbaum 1998).

Lexical substitution typically requires that the meaning of one word would entail the meaning of the other. For instance, in question answering, the word *company* in a question can be substituted in an answer text by *firm*, *automaker*, or*subsidiary*, whose meanings entail the meaning of *company*. However, as it turns out, traditional distributional similarity measures do not capture well such lexical substitution relationships, but rather capture a somewhat broader (and looser) notion of semantic similarity. For example, quite distant co-hyponyms such as *party* and *company* also come out as distributionally similar to *country*, due to a partial overlap of their semantic properties. Clearly, the meanings of these words do not entail each other.

Motivated by these observations, our long term goal is to investigate whether the distributional similarity scheme may be improved to yield tighter semantic similarities, and eventually better approximation of lexical entailments. This article presents one component of this research plan, which focuses on improving the underlying semantic quality of distributional word feature vectors. The paper describes the methodology, definitions, and analysis of our investigation and the resulting bootstrapping scheme for feature weighting which yielded improved empirical performance.

1.2 Main Contributions and Outline

As a starting point for our investigation, an operational definition was needed for evaluating the correctness of candidate pairs of similar words. Following the lexical substitution motivation, in Section 3 we formulate the **substitutable lexical entailment**

relation (or **lexical entailment**, for brevity), refining earlier definitions in (Geffet and Dagan 2004, 2005). Generally speaking, this relation holds for a pair of words if a possible meaning of one word entails a meaning of the other, and the entailing word can substitute the entailed one in some typical contexts. Lexical entailment overlaps partly with traditional lexical semantic relationships, while capturing more generally the lexical substitution needs of applications. Empirically, high inter-annotator agreement was obtained when judging the output of distributional similarity measures for lexical entailment.

Next, we analyzed the typical behavior of existing word similarity measures relative to the lexical entailment criterion. Choosing the commonly used measure of Lin (1998) as a representative case, the analysis shows that quite noisy feature vectors are a major cause for generating rather "loose" semantic similarities. On the other hand, one may expect that features which seem to be most characteristic for a word's meaning should receive the highest feature weights. Yet, this does not seem to be the case for common feature weighting functions, such as Point-wise Mutual Information (Church and Patrick 1990; Hindle 1990).

Following the above observations, we developed a bootstrapping formula that improves the original feature weights (Section 4), leading to better feature vectors and better similarity predictions. The general idea is to promote the weights of features that are common for semantically similar words, since these features are likely to be most characteristic for the word's meaning. This idea is implemented by a bootstrapping scheme, where the initial (and cruder) similarity measure provides initial approximation for semantic word similarity. The bootstrapping method yields a high concentration of semantically characteristic features amongst the top-ranked features of the vector, which also allows aggressive feature reduction.

The bootstrapping scheme was evaluated in two experimental settings, which correspond to the two types of applications for distributional similarity. First, it achieved significant improvements in predicting lexical entailment as assessed by human judgments, when applied over several base similarity measures (Section 5). Additional analysis relative to the lexical entailment dataset revealed cleaner and more characteristic feature vectors for the bootstrapping method. To obtain quantitative analysis of this behavior, we defined a measure called **average common-feature rank ratio**. This measure captures the idea that a prominent feature for a word is expected to be prominent also for semantically similar words, while being less prominent for unrelated words. To the best of our knowledge this is the first proposed measure for direct analysis of the quality of feature weighting functions, without the need to employ them within some vector similarity measure.

As a second evaluation, we applied the bootstrapping scheme for similarity-based prediction of co-occurrence likelihood within a typical pseudo-word sense disambiguation experiment, obtaining substantial error reductions (Section 7). Section 8 concludes this article, suggesting the relevance of our analysis and bootstrapping scheme for the general use of distributional feature vectors.[1]

1 A preliminary version of the bootstrapping method was presented in (Geffet and Dagan 2004). That paper presented initial results for the bootstrapping scheme, when applied only over Lin's measure and tested by the manually judged dataset of lexical entailment. The current paper extends our initial results in many respects. It refines the definition of lexical entailment; utilizes a revised test set of larger scope and higher quality, annotated by three assessors; extends the experiments to two additional similarity measures; provides comparative qualitative and quantitative analysis of the bootstrapped vectors, while employing our proposed average common-feature rank ratio; and presents an additional evaluation based on a pseudo-WSD task.

2. Background: Distributional Similarity Models

This section reviews the components of the distributional similarity approach and specifies the measures and functions that were utilized by our work.

The Distributional Hypothesis assumes that semantically similar words appear in similar contexts, suggesting that semantic similarity can be detected by comparing contexts of words. This is the underlying principle of the vector-based distributional similarity model, which is comprised of two phases. First, context features for each word are constructed and assigned weights; then, the weighted feature vectors of pairs of words are compared by a vector similarity measure. The following two subsections review typical methods for each phase.

2.1 Features and Weighting Functions

In the typical computational setting word contexts are represented by feature vectors. A feature represents another word (or term) w' with which w co-occurs, and possibly specifies also the syntactic relationship between the two words, as in (Grefenstette 1994; Lin 1998; Weeds and Weir 2005). Thus, a word (or term) w is represented by a feature vector, where each entry in the vector corresponds to a feature f. Pado and Lapata (2007) demonstrate that using syntactic dependency-based features help distinguishing among classes of lexical relations, which seems to be more difficult when using "bag of words" features that are based on co-occurrence in a text window.

A syntactic-based feature f for a word w is defined as a triple:

$$\langle fw, syn\_rel, f\_role \rangle$$

where *fw* is a context word (or term) that co-occurs with w under the syntactic dependency relation *syn_rel*. The feature role (*f_role*) corresponds to the role of the feature word *fw* in the syntactic dependency, being either the head (denoted h) or the modifier (denoted m) of the relation. For example, given the word *company* the feature $\langle$*earnings, gen, h*$\rangle$ corresponds to the genitive relationship *company's earnings*, and $\langle$*investor, pcomp_of, m*$\rangle$ corresponds to the prepositional complement relationship *the company of the investor*.[2] Throughout this article we use syntactic dependency relationships generated by the Minipar dependency parser (Lin 1993). Table 1 lists common Minipar dependency relations involving nouns. Minipar also identifies multi-word expressions, which is advantageous for detecting distributional similarity for such terms. For example, Curran (2004) reports that multi-word expressions make up between 14–25% of the synonyms in a gold-standard thesaurus.

Thus, in our representation the corpus is first transformed to a set S of dependency relationship instances of the form $\langle w,f \rangle$, where each pair corresponds to a single co-occurrence of w and f in the corpus. f is termed as a feature of w. Then, a word w is represented by a feature vector, where each entry in the vector corresponds to one feature f. The value of the entry is determined by a feature weighting function *weight*(w,f), which quantifies the degree of statistical association between w and f in the set S. For example, some feature weighting functions are based on the logarithm of the word–feature co-occurrence frequency (Ruge 1992), or on the conditional probability

2 Following a common practice, we consider the relationship between a head noun (*company* in the example above) and the nominal complement of a modifying prepositional phrase (*investor*) as a single direct dependency relationship. The preposition itself is encoded in the dependency relation name, having a distinct relation for each preposition.

Table 1
Common grammatical relations of Minipar involving nouns.

Relation	Description
appo	apposition
comp1	first complement
det	determiner
gen	genitive marker
mod	the relationship between a word and its adjunct modifier
pnmod	post nominal modifier
pcomp	nominal complement of prepositions
post	post determiner
vrel	passive verb modifier of nouns
obj	object of verbs
obj2	second object of ditransitive verbs
subj	subject of verbs
s	surface subject

of the feature given the word (Pereira, Tishby, and Lee 1993; Lee 1999; Dagan, Lee, and Pereira 1999).

Probably the most widely used feature weighting function is (point-wise) Mutual Information (*MI*) (Church and Patrick 1990; Hindle 1990; Luk 1995; Lin 1998; Gauch, Wang, and Rachakonda 1999; Dagan 2000; Weeds, Weir, and McCarthy 2004; Pantel, Ravichandran, and Hovy 2004; Pantel and Ravichandran 2004; Chklovski and Pantel 2004; Baroni and Vegnaduzzo 2004), defined by:

$$weight_{MI}(w, f) = \log_2 \frac{P(w,f)}{P(w)P(f)} \quad (1)$$

We calculate the *MI* weights by the following statistics in the space of co-occurrence instances S:

$$weight_{MI}(w, f) = \log_2 \frac{count(w, f) \cdot nrels}{count(w) \cdot count(f)} \quad (2)$$

where $count(w, f)$ is the frequency of the co-occurrence pair $\langle w,f \rangle$ in S, $count(w)$ and $count(f)$ are the independent frequencies of w and f in S, and $nrels$ is the size of S. High *MI* weights are assumed to correspond to strong word–feature associations.

Curran and Moens (2002) argue that, generally, informative features are statistically correlated with their corresponding headword. Thus, they suggest that any statistical test used for collocations is a good starting point for improving feature-weight functions. In their experiments the t-test-based metric yielded the best empirical performance.

However, a known weakness of *MI* and most of the other statistical weighting functions used for collocation extraction, including t-test and chi^2, is their tendency to inflate the weights for rare features (Dunning 1993). In addition, a major property of lexical collocations is their "non-substitutability", as termed in (Manning and Schutze 1999). That is, typically neither a headword nor a modifier in the collocation can be

substituted by their synonyms or other related terms. This implies that using modifiers within strong collocations as features for a head word would provide a rather small amount of common features for semantically similar words. Hence, the above functions seem less suitable for learning broader substitutability relationships, such as lexical entailment.

Yet, similarity measures that utilize *MI* weights showed good performance. In particular, a common practice is to filter out features by minimal frequency and weight thresholds. Then, a word's vector is constructed from the remaining (not filtered) features, that are strongly associated with the word. These features are denoted here as **active** features.

In the current work we use *MI* for data analysis, and for the evaluations of vector quality and word similarity performance.

2.2 Vector Similarity Measures

Once feature vectors have been constructed the similarity between two words is defined by some vector similarity measure. Similarity measures which have been used in the cited papers above include weighted Jaccard (Grefenstette 1994), Cosine (Ruge 1992), and various information theoretic measures as introduced and reviewed in (Lee 1997, 1999). In the current work we experiment with the following three popular similarity measures.

1. The basic Jaccard measure compares the number of common features with the overall number of features for a pair of words. One of the weighted generalization of this scheme to non-binary values replaces intersection with minimum weight, union with maximum weight, and set cardinality with summation. This measure is commonly referred as *weighted Jaccard* (*WJ*) (Grefenstette 1994; Dagan, Marcus, and Markovitch 1995; Dagan 2000; Gasperin and Vieira 2004), defined as follows:

$$sim_{WJ}(w,v) = \frac{\sum_{f \in F(w) \cap F(v)} \min(weight(w,f), weight(v,f))}{\sum_{f \in F(w) \cup F(v)} \max(weight(w,f), weight(v,f))} \quad (3)$$

   where $F(w)$ and $F(v)$ are the sets of active features of the two words w and v. The appealing property of this measure is that it considers the association weights rather than just the number of common features.

2. The standard *Cosine* measure (*COS*), which is popularly employed for Information Retrieval (Salton and McGill 1983) and also utilized for learning distributionally similar words (Ruge 1992; Caraballo 1999; Gauch, Wang, and Rachakonda 1999; Pantel and Ravichandran 2004), is defined as follows:

$$sim_{COS}(w,v) = \frac{\sum_f (weight(w,f) \cdot weight(v,f))}{\sqrt{\sum_f (weight(w,f))^2} \cdot \sqrt{\sum_f (weight(v,f))^2}} \quad (4)$$

   This measure computes the cosine of the angle between the two feature vectors, which normalizes the vector lengths and thus avoids inflated discrimination between vectors of significantly different lengths.

3. A popular state of the art measure has been developed by Lin (1998), motivated by Information Theory principles. This measure behaves quite

similarly to the weighted Jaccard measure (Weeds, Weir, and McCarthy 2004), and is defined as follows:

$$sim_{LIN}(w,v) = \frac{\sum_{f \in F(w) \cap F(v)} (weight_{MI}(w,f) + weight_{MI}(v,f))}{\sum_{f \in F(w)} weight_{MI}(w,f) + \sum_{f \in F(v)} weight_{MI}(v,f)} \quad (5)$$

where $F(w)$ and $F(v)$ are the active features of the two words. The weight function used originally by Lin is *MI* (Equation 1).

It is interesting to note that a relatively recent work by Weeds and Weir (2005) investigates a more generic similarity framework. Within their framework, the similarity of two nouns is viewed as the ability to predict the distribution of one of them based on that of the other. Their proposed formula combines the precision and recall of a potential "retrieval" of similar words based on the features of the target word. The precision of w's prediction of v's feature distribution indicates how many of the features of the word w co-occurred with the word v. The recall of w's prediction of v's features indicates how many of the features of v co-occurred with w. Words with both high precision and high recall can be obtained by computing their harmonic mean, *mh*, (or F-score) and a weighted arithmetic mean. However, after empirical tuning of weights for the arithmetic mean their formula practically reduces to Lin's measure, as was anticipated by their own analysis (in Section 4 of their paper).

Consequently, we choose the Lin measure (Equation 5) (henceforth denoted as *LIN*) as representative for state of the art and utilize it for data analysis and as a starting point for improvement. To further explore and evaluate our new weighting scheme, independently of a single similarity measure, we conduct evaluations also with the other two similarity measures of weighted Jaccard and Cosine.

3. Substitutable Lexical Entailment

As mentioned in the Introduction, the long term research goal which inspired our work is modeling meaning-entailing lexical substitution. Motivated by this goal, we proposed in earlier work (Geffet and Dagan 2004, 2005) a new type of lexical relationship which aims to capture such lexical substitution needs. Here we adopt that approach and formulate a refined definition for this relationship, termed **substitutable lexical entailment**. In the context of the current article, utilizing a concrete target notion of word similarity enabled us to apply direct human judgment for the "correctness" (relative to the defined notion) of candidate word pairs suggested by distributional similarity. Utilizing these judgments we could analyze the behavior of alternative distributional vector representations and, in particular, conduct error analysis for word pair candidates that were judged negatively.

The discussion in the Introduction suggested that multiple text understanding applications need to identify term pairs whose meanings are both entailing and substitutable. Such pairs seem to be most appropriate for lexical substitution in a meaning preserving scenario. To model this goal we present an operational definition for a lexical semantic relationship that integrates the two aspects of entailment and substitutability,[3] which is termed **substitutable lexical entailment** (or **lexical entailment**, for brevity).

3 The WordNet definition of the lexical entailment relation is specified only for verbs and, therefore, is not felicitous for general purposes: A verb X entails Y if X cannot be done unless Y is, or has been, done (e.g., *snore* and *sleep*).

This relationship holds for a given directional pair of terms (w, v) saying that w entails v, if the following two conditions are fulfilled:

1. *Word meaning entailment*: the meaning of a possible sense of w implies a possible sense of v;
2. *Substitutability*: w can substitute v in some naturally occurring sentence, such that the meaning of the modified sentence would entail the meaning of the original one.

To operationally assess the first condition (by annotators) we propose considering the meaning of terms by existential statements of the form "there exists an instance of the meaning of the term w in some context" (notice that unlike propositions, it is not intuitive for annotators to assign truth values to terms). For example, the word *company* would correspond to the existential statement "there exists an instance of the concept *company* in some context". Thus, if in some context "there is a *company*" (in the sense of "commercial organization") then necessarily "there is a *firm*" in that context (in the corresponding sense). Therefore, we conclude that the meaning of *company* implies the meaning of *firm*. On the other hand, "there is an *organization*" does not necessarily imply the existence of *company*, since *organization* might stand for some non-profit association, as well. Therefore, we conclude that *organization* does not entail *company*.

To assess the second condition, the annotators need to identify some natural context in which the lexical substitution would satisfy entailment between the modified sentence and the original one. Practically, in our experiments presented in Section 5 the human assessors could consult external lexical resources and the entire web to obtain all the senses of the words and possible sentences for substitution. We note that the task of identifying the common sense of two given words is quite easy since they mutually disambiguate each other, and once the common sense is known it naturally helps finding a corresponding common context. We note that this condition is important, in particular, in order to eliminate cases of anaphora and co-reference in contexts, where two words quite different in their meaning can sometimes appear in the same contexts only due to the text pragmatics in a particular situation. For example, in some situations *worker* and *demonstrator* could be used interchangeably in text, but clearly it is a discourse co-reference rather than common meaning that makes the substitution possible. Instead, we are interested in identifying word pairs in which one word's meaning provides a reference to the entailed word's meaning. This purpose is exactly captured by the existential propositions of the first criterion above.

As reported further in Section 5.1, we observed that assessing these two conditions for candidate word similarity pairs was quite intuitive for annotators, and yielded good cross-annotator agreement. Overall, substitutable lexical entailment captures directly the typical lexical substitution scenario in text understanding applications, as well as in generic textual entailment modeling. In fact, this relation partially overlaps with several traditional lexical semantic relations that are known as relevant for lexical substitution, such as synonymy, hyponymy, and some cases of meronymy. For example, we say that the meaning of *company* is lexically entailed by the meaning of *firm* (synonym) or *automaker* (hyponym), while the word *government* entails *minister* (meronym) as "The government voted for the new law" entails "A minister in the government voted for the new law".

On the other hand, lexical entailment is not just a superset of other known relations, but it is rather designed to select those sub-cases of other lexical relations that are needed

Table 2
The top-20 most similar words for *country* (and their ranks) in the similarity list of *LIN*, followed by the next four words in the similarity list that were judged as entailing at least in one direction. 12 out of 20 top similarities (60%) were judged as mutually non-entailing and are marked with '*'. The similarity data was produced as described in Section 5.

nation	1	*city	7	economy	13	*company	19
region	2	territory	8	*neighbor	14	*industry	20
state	3	area	9	*sector	15	kingdom	30
*world	4	*town	10	*member	16	place	35
*island	5	republic	11	*party	17	colony	41
*province	6	*north	12	government	18	democracy	82

for applied entailment inference. For example, lexical entailment does not cover all cases of meronyms, (e.g., *division* does not entail *company*), but only some sub-cases of part-whole relationship mentioned above. In addition, some other relations are also covered by lexical entailment, like *ocean* and *water* and *murder* and *death*, which do not seem to directly correspond to meronymy or hyponymy relations.

Notice also that while lexical entailment is a directional relation, that specifies which word of the pair entails the other, the relation may hold in both directions for a pair of words, as is the case for synonyms. More detailed motivations for the substitutable lexical entailment relation and analysis of its relationship to traditional lexical semantic relations appear in (Geffet 2006; Geffet and Dagan 2004, 2005).

4. Bootstrapping Feature Weights

To gain better understanding of distributional similarity behavior we first analyzed the output of the *LIN* measure, as a representative case for state of the art, while regarding lexical entailment as a reference evaluation criterion. We judge as correct, with respect to lexical entailment, those candidate pairs of the distributional similarity method for which entailment holds at least in one direction.

For example, the word *area* is entailed by *country*, since the existence of *country* entails the existence of *area*, and the sentence "There is no rain in subtropical countries during the summer period" entails the sentence "There is no rain in subtropical areas during the summer period"). As another example, *democracy* is a type of *country* at the political sense, thus the existence entailment holds and also the sentence "Israel is a democracy in the Middle East" entails "Israel is a country in the Middle East".

On the other hand, our analysis revealed that many candidate word similarity pairs suggested by distributional similarity measures do not correspond to "tight" semantic relationships. In particular, many word pairs suggested by the *LIN* measure do not satisfy the lexical entailment relation, as demonstrated in Table 2.

A deeper look at the corresponding word feature vectors reveals typical reasons for these lexical entailment prediction errors. Most relevant for the scope of the current article, in many cases highly ranked features in a word vector (when sorting the features by their weight) do not seem very characteristic for the word meaning. This is demonstrated in Table 3, which shows the top-10 features in the vector of *country*. As can be seen, some of the top features are either too specific (*landlocked*, *airspace*), and are thus less reliable, or too general (*destination*, *ambition*), thus not indicative and

Table 3
The top-10 ranked features for *country* produced by *MI*, the weighting function employed in *LIN* method.

Feature	$weight_{MI}$
Commercial bank, gen, *h*	8.08
Destination, pcomp_of, *m*	7.97
Airspace, pcomp_of, *h*	7.83
Landlocked, mod, *m*	7.79
Trade balance, gen, *h*	7.78
Sovereignty, pcomp_of, *h*	7.78
Ambition, nn, *h*	7.77
Bourse, gen, *h*	7.72
Politician, gen, *h*	7.54
Border, pcomp_of, *h*	7.53

may co-occur with many different types of words. On the other hand, intuitively more characteristic features of *country*, like *population* and *governor*, occur further down the sorted feature list, at positions 461 and 832. Overall, features that seem to characterize the word meaning well are scattered across the ranked feature list while many non-indicative features receive high weights. This behavior often yields high similarity scores for word pairs whose semantic similarity is rather loose while missing some much tighter similarities.

Furthermore, we observed that characteristic features for a word w, which should receive higher weights, are expected to be common for w and other words that are semantically similar to it. This observation suggests a computational scheme which would promote the weights of features that are common for semantically similar words. Of course, there is an inherent circularity in such a scheme: to determine which features should receive high weights we need to know which words are semantically similar, while computing distributional semantic similarity already requires pre-determined feature weights.

This kind of circularity can be approached by a bootstrapping scheme. We first compute initial distributional similarity values, based on an initial feature weighting function. Then, to learn more accurate feature weights for a word w, we promote features that characterize other words that are initially known to be similar to w. By the same rationale, features that do not characterize many words that are sufficiently similar to w are demoted. Even if such features happen to have a strong direct statistical association with w they would not be considered reliable, since they are not supported by additional words that have a similar meaning to that of w.

4.1 Bootstrapped feature weight definition

The bootstrapped feature weight is defined as follows. First, some standard word similarity measure sim is computed to obtain an initial approximation of the similarity space. Then, we define the **word set** of a feature f, denoted by *WS*(f), as the set of words for which f is an active feature. Recall from Section 2.2 that an active feature is a feature that is strongly associated with the word, that is, its (initial) weight is higher than an

empirically predefined threshold, θ_{weight}. The **semantic neighborhood** of w, denoted by $N(w)$, is defined as the set of all words v which are considered sufficiently similar to w, satisfying $sim(w,v) > \theta_{sim}$, where θ_{sim} is a second empirically determined threshold. The bootstrapped feature weight, denoted $weight^B$, is then defined by:

$$weight^B(w,f) = \sum_{v \in WS(f) \cap N(w)} sim(w,v) \quad (6)$$

That is, we identify all words v that are in the semantic neighborhood of w and are also characterized by f, and then sum the values of their similarities to w.

Intuitively, summing the similarity values above captures simultaneously a desired balance between feature specificity and generality, addressing the observations in the beginning of this section. Some features might characterize just a single word that is very similar to w, but then the sum of similarities will include a single element, yielding a relatively low weight. This is why the sum of similarities is used rather than an average value, which might become too high by chance when computed over just a single element (or very few elements). Relatively generic features, which occur with many words and are thus less indicative, may characterize more words within $N(w)$ but then on average the similarity values of these words with w is likely to be lower, contributing smaller values to the sum. To receive a high overall weight a reliable feature has to characterize multiple words that are highly similar to w.

We note that the bootstrapped weight is a sum of word similarity values rather than a direct function of word-feature association values, which is the more common approach. It thus does not depend on the exact statistical co-occurrence level between w and f. Instead, it depends on a more global assessment of the association between f and the semantic vicinity of w. We notice that the bootstrapped weight is determined separately relative to each individual word. This differs from measures that are global word-independent functions of the feature, such as the feature entropy used in (Grefenstette 1994) and the feature term strength relative to a predefined class as employed in (Pekar, Krkoska, and Staab 2004) for supervised word classification.

4.2 Feature reduction and similarity re-computation

Once the bootstrapped weights have been computed, their sufficient accuracy allows for aggressive feature reduction. As shown in the following section, in our experiments it sufficed to use only the top-100 features for each word in order to obtain optimal word similarity results, since the most informative features now receive the highest weights.

Finally, similarity between words is re-computed over the reduced vectors using the *sim* function with $weight^B$ replacing the original feature weights. The resulting similarity measure is further referred as sim^B.

5. Evaluation by Lexical Entailment

To test the effectiveness of the bootstrapped weighting scheme we first evaluated whether it contributes to better prediction of lexical entailment. This evaluation was based on gold standard annotation determined by human judgments of the substitutable lexical entailment relation, as defined in Section 3. The new similarity scheme, sim^B, based on the bootstrapped weights, was first computed using the standard *LIN* method as the initial similarity measure. The resulting similarity lists of sim_{LIN} (the original *LIN* method) and sim^B_{LIN} (*Bootstrapped LIN*) schemes were evaluated for a sample of nouns (Section 5.2). Then, the evaluation was extended (Section 5.3) to apply

the bootstrapping scheme over the two additional similarity measures that were presented in Section 2.2, sim_{WJ} (weighted Jaccard), and sim_{COS} (Cosine). Along with these lexical entailment evaluations we also analyzed directly the quality of the bootstrapped feature vectors, according to the average common-feature rank ratio measure, which was defined in Section 6.

5.1 Experimental Setting

Our experiments were conducted using statistics from an 18 million token subset of the Reuters RCV1 corpus (known as Reuters Corpus, Volume 1, English Language, 1996-08-20 to 1997-08-19), parsed by Lin's Minipar dependency parser (Lin 1993).

The test set of candidate word similarity pairs was constructed for a sample of 30 randomly selected nouns, whose corpus frequency exceeds 500. In our primary experiment we computed the top-40 most similar words for each noun by the sim_{LIN} and by sim^B_{LIN} measures, yielding 1,200 pairs for each method, and 2,400 pairs altogether. About 800 of these pairs were common for the two methods, therefore leaving out approximately 1,600 distinct candidate word similarity pairs. Since the lexical entailment relation is directional, each candidate pair was duplicated to create two directional pairs, yielding a test set of 3,200 pairs. Thus, for each pair of words, w and v, the two ordered pairs (w, v) and (v, w) were created to be judged separately for entailment in the specified direction (whether the first word entails the other). Consequently, a non-directional candidate similarity pair w, v is considered as a correct entailment if it was assessed as an entailing pair at least in one direction.

The assessors were only provided with a list of word pairs without any contextual information and could consult any available dictionary, WordNet and search the web. The judgment criterion follows the criterion presented in Section 3. In particular, the judges were asked to apply the two operational conditions, existence and substitutability in context, on each given pair. Prior to performing the final test of the annotation experiment, the judges were presented with an annotated set of entailing and non-entailing pairs along with the existential statements and sample sentences for substitution demonstrating how the two conditions can be applied in different cases of entailment. In addition, they had to judge a training set of several dozen pairs and then discuss their judgment decisions with each other to gain a better understanding of the two criteria.

The following example illustrates the above process. Given a non-directional pair *company, organization*, 2 directional pairs are created: *(company, organization)* and *(organization, company)*. The former pair is judged as a correct entailment: the existence of a company entails the existence of an organization, and the meaning of the sentence: "John works for a large company" entails the meaning of the sentence with substitution: "John works for a large organization". Hence, *company* lexically entails *organization*, but not vice versa (as shown in Section 3.3), therefore the second pair is judged as not entailing. Eventually, the non-directional pair *{company, organization}* is considered as a correct entailment.

Finally, the above test set of 3,200 pairs was split into three disjoint subsets that were judged by three native English speaking assessors, who also possess a Bachelor degree in English Linguistics. For each subset a different pair of assessors was assigned, each person judging the entire subset. The judges were grouped into three different pairs (i.e., JudgeI+JudgeII, JudgeII+JudgeIII and JudgeI+JudgeIII). Each pair was assigned initially to judge all the word similarities in each subset, while the third assessor was employed in cases of disagreement between the first two. The majority vote was taken as the final

decision. Hence, each assessor had to fully annotate two thirds of the data and for a third subset she only had to judge the pairs for which there was disagreement between the other two judges. This was done in order to measure the agreement achieved for different pairs of annotators.

The output pairs from both methods were mixed so the assessors could not associate a pair with the method that proposed it. We note that this evaluation methodology, in which human assessors judge the correctness of candidate pairs by some semantic substitutability criterion, is similar to common evaluation methodologies used for paraphrase acquisition (Lin and Pantel 2001; Barzilay and McKeown 2001; Szpektor et al. 2004).

Measuring human agreement level for this task, the proportions of matching decisions were 93.5% between Judge I and Judge II, 90% for Judge I and Judge III, and 91.2% for Judge II and Judge III. The corresponding Kappa values are 0.83, 0.80, and 0.80, which is regarded as "very good agreement" (Landis and Koch 1997). It is interesting to note that after some discussion most of the disagreements have been settled down, while few remaining mismatches were due to different understanding of word meanings. These findings seem to have a similar flavor to the human agreement findings reported for the Recognizing Textual Entailment challenges (Dagan, Glickman, and Magnini 2006; Bar-Haim et al. 2006), in which entailment was judged for pairs of sentences. In fact, the Kappa values obtained in our evaluation are substantially higher than reported for sentence level textual entailment, which suggests that it is easier to make entailment judgments at the lexical level than at the full sentence level.

Parameter values of the algorithms were tuned using a development set of similarity pairs generated for 10 additional nouns, distinct from the 30 nouns used for the test set. The parameters were optimized by running the algorithm systematically with various values across the parameter scales and judging a subset sample of the results. $weight_{MI} = 4$ was found as the optimal *MI* threshold for active feature weights (features included in the feature vectors), yielding a 10% precision increase of sim_{LIN} and removing over 50% of the data relative to no feature filtering. Accordingly, this value also serves as the θ_{weight} threshold in the bootstrapping scheme (Section 4). As for the θ_{sim} parameter, the best results on the development set were obtained for $\theta_{sim} = 0.04$, $\theta_{sim} = 0.02$, and $\theta_{sim} = 0.01$ when bootstrapping over the initial similarity measures *LIN*, *WJ*, and *COS*, respectively.

5.2 Evaluation results for sim^{B}_{LIN}

We measured the contribution of the improved feature vectors to the resulting precision of sim_{LIN} and sim^{B}_{LIN} in predicting lexical entailment. The results are presented in Table 4, where precision and error reduction values were computed for the top-20, 30, and 40 word similarity pairs produced by each method. It can be seen that the *Bootstrapped LIN* method outperformed the original *LIN* approach by 6–9 precision points at all Top-N levels. As expected, the precision for the shorter top-20 list is higher for both methods, thus leaving a bit less room for improvement.

Overall, the *Bootstrapped LIN* method extracted 104 (21%) more correct similarity pairs than the other measure and reduced the number of errors by almost 15%. We also computed the relative recall which shows the percentage of correct word similarities found by each method relative to the joint set of similarities that were extracted by both methods. The overall relative recall of the *Bootstrapped LIN* was quite high (94%), exceeding *LIN*'s relative recall (of 78%) by 16%. We found that the bootstrapped method

Table 4
Lexical entailment precision values for Top-N similar words by the *Bootstrapped LIN* and the original *LIN* method.

Top-N words	Correct Entailments (%)		Error Rate Reduction (%)
	sim_{LIN}	sim^B_{LIN}	
Top-20	52.0	57.9	12.3
Top-30	48.2	56.2	15.4
Top-40	41.0	49.7	14.7

covers over 90% of the correct similarities learnt by the original method, while also identifying many additional correct pairs.

It should be noted at this point that the current limited precision levels are determined not just by the quality of the feature vectors but much due to the nature of the vector comparison measure itself (i.e., the *LIN* formula, as well weighted Jaccard and Cosine as reported in Section 5.3). It was observed in other work (Geffet and Dagan 2005) that these common types of vector comparison schemes exhibit certain flaws in predicting lexical entailment. Our present work thus shows that the bootstrapping method yields a significant improvement in feature vector quality while future research is needed to investigate improved vector comparison schemes.

An additional indication of the improved vector quality is the massive feature reduction allowed by having the most characteristic features concentrated at the top ranks of the vectors. The vectors of active features of *LIN*, as constructed after standard feature filtering (Section 5.1), could be further reduced by the bootstrapped weighting to about one third of their size . As illustrated in Figure 1, changing the vector size affects significantly the similarity results. In sim^B_{LIN} the best result was obtained with top-100 features per word, while using less than 100 or more than 150 features caused a 5–10% decrease in performance. On the other hand, an attempt to cut off the lower ranked features of the *MI* weighting always resulted in a noticeable decrease in precision. These results show that for *MI* weighting many important features appear further down in the ranked vectors while for the bootstrapped weighting adding too many features adds mostly *noise*, since most characteristic features are concentrated at the top ranks. Thus, in addition to better feature weighting the bootstrapping step provides effective feature reduction which improves vector quality, and consequently the similarity results.

We note that the optimal vector size we obtained conforms to previous results, e.g., by Widdows (2003) and Curran (2004), Curran and Moens (2002), who also used reduced vectors of up to 100 features as optimal for learning hyponymy and synonymy, correspondingly. In (Widdows 2003) the known SVD method for dimension reduction of LSA-based vectors is applied, while in (Curran 2004; Curran and Moens 2002) only the strongly associated verbs (direct and indirect objects of the noun) are selected as "canonical features" that are expected to be shared by true synonyms.

Finally, we tried executing an additional bootstrapping iteration of $weight^B$ calculation over the similarity results of sim^B_{LIN}. The resulting increase in precision was much smaller, of about 2%, showing that most of the potential benefit is exploited in the first bootstrapping iteration (which is not uncommon for natural language data). On the

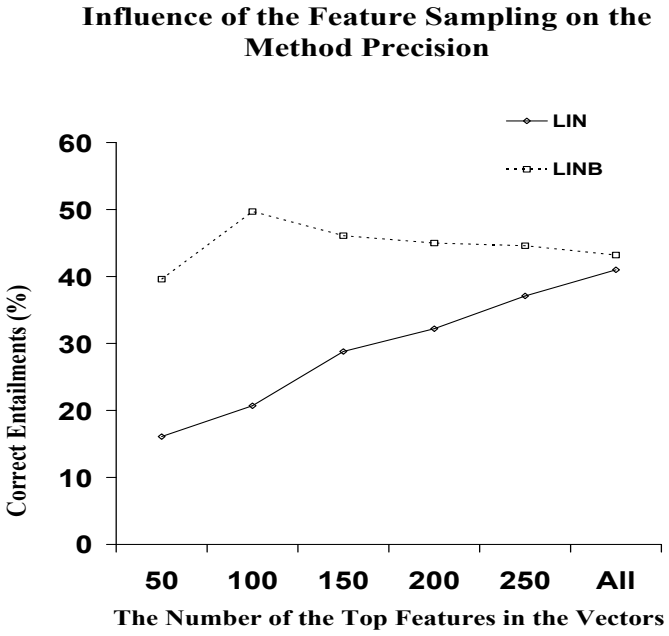

Figure 1
Percentage of correct entailments within the top-40 candidate pairs of each of the methods, *LIN* and *Bootstrapped LIN* (denoted as *LINB* in the figure), when using varying numbers of top-ranked features in the feature vector. The value of "All" corresponds to the full size of vectors and is typically in the range of 300-400 features.

other hand, computing the bootstrapping weight twice increases computational time significantly, which led us to suggest a single bootstrapping iteration as a reasonable cost effectiveness tradeoff for our data.

5.3 Evaluation for sim^B_{WJ} and sim^B_{COS}

To further validate the behavior of the bootstrapping scheme we experimented with two additional similarity measures, weighted Jaccard (sim_{WJ}) and Cosine (sim_{COS}) (described in Section 2.2). For each of the additional measures the experiment repeats the main three steps described in Section 4: initially, the basic similarity lists are calculated for each of the measures using *MI* weighting; then, the bootstrapped weighting, $weight^B$, is computed based on the initial similarities, yielding new word feature vectors; finally, the similarity values are recomputed by the same vector similarity measure using the new feature vectors.

Table 5
Comparative precision values for the top-20 similarity lists of the three selected similarity measures, with *MI* and Bootstrapped feature weighting for each.

Measure	LIN–LIN^B	WJ–WJ^B	COS–COS^B
Correct Similarities (%)	52.0–57.9	51.0–54.8	46.1–50.9

To assess the effectiveness of $weight^B$ we computed the four alternative output similarity lists, using the sim_{WJ} and sim_{COS} similarity measures each with the $weight_{MI}$ and $weight^B$ weighting functions. The four lists were judged for lexical entailment by three assessors, according to the same procedure described in Section 5.1. To make the additional manual evaluation affordable we judged the top-20 similar words in each list for each of the 30 target nouns of Section 5.1.

Table 5 summarizes the precision values achieved by *LIN*, *WJ*, and *COS* with both $weight_{MI}$ and $weight^B$. As shown in the table, bootstrapped weighting consistently contributed between 4–6 points to the accuracy of each method in the top-20 similarity list. We view the results as quite positive, considering that improving over top-20 similarities is a much more challenging task than improving over longer similarity lists, while the improvement was achieved only by modifying the feature vectors without changing the similarity measure itself (as hinted in Section 5.2). Our results are also compatible with previous findings in the literature (Dagan, Lee, and Pereira 1999; Weeds, Weir, and McCarthy 2004) that found *LIN* and *WJ* to be more accurate for similarity acquisition than *COS*. Overall, the results demonstrate that the bootstrapped weighting scheme consistently produces improved results.

An interesting behavior of the bootstrapping process is that the most prominent features for a given target word converge across the different initial similarity measures, as exemplified in Table 6. In particular, although the initial similarity lists overlap only partly,[4] the overlap of the top-30 features for our 30-word sample was ranging between 88% and 100%. This provides additional evidence that the quality of the bootstrapped weighting is quite similar for various initial similarity measures.

6. Analyzing the Bootstrapped Feature Vector Quality

In this section we provide an in-depth analysis of the bootstrapping feature weighting quality compared to the state-of-the-art *MI* weighting function.

6.1 Qualitative Observations

The problematic feature ranking noticed at the beginning of Section 4 can be revealed more objectively by examining the common features which contribute mostly to the word similarity scores. To that end, we examine the common features of the two given words and sort them by the sum of their weights in both word vectors. Table 7 shows

4 Overlap rate was about 40% between *COS* and *WJ* or *LIN*, and 70% between *WJ* and *LIN*. The overlap was computed following the procedure of Weeds, Weir, and McCarthy (2004), disregarding the order of the similar words in the lists. Interestingly, they obtained roughly resembling figures, of 28% overlap for *COS* and *WJ*, 32% overlap for *COS* and *LIN*, and 81% overlap between *LIN* and *WJ*.

Table 6
Top-30 features of *town* by bootstrapped weighting based on *LIN*, *WJ*, and *COS* as initial similarities. The three sets of words are almost identical, with relatively minor ranking differences.

LIN^B	WJ^B	COS^B
southern	southern	northern
northern	northern	southern
office	office	remote
eastern	official	eastern
remote	coastal	official
official	eastern	based
troop	northeastern	northeastern
northeastern	remote	office
people	troop	coastal
coastal	people	northwestern
attack	based	people
based	populated	attack
populated	attack	troop
northwestern	home	home
base	northwestern	south
home	south	western
south	western	city
west	west	populated
western	resident	base
neighboring	neighboring	resident
resident	house	north
plant	city	west
police	base	neighboring
held	trip	trip
locate	camp	surrounding
trip	held	police
city	north	held
site	locate	locate
camp	surrounding	house
surrounding	police	camp

the top-10 common features by this sorting for a pair of truly similar (lexically entailing) words (*country–state*), and for a pair of non-entailing words (*country–party*). For each common feature the table shows its two corresponding ranks in the feature vectors of the two words.

It can be observed in Table 7 that for both word pairs the common features are scattered across the pair of feature vectors, making it difficult to distinguish between the truly similar and the non-similar pair. We suggest, on the other hand, that the desired behavior of effective feature weighting is that the common features of truly similar words would be concentrated at the top ranks of both word vectors. In other words, if the two words are semantically similar then we expect them to share their

Table 7
LIN (*MI*) weighting: The top-10 common features for *country–state* and *country–party*, along with their corresponding ranks in each of the two feature vectors. The features are sorted by the sum of their feature weights with both words.

Country–State	Ranks		*Country–Party*	Ranks	
Broadcast, pcomp_in, *h*	24	50	Brass, nn, *h*	64	22
Goods, mod, *h*	140	16	Concluding, pcomp_of, *h*	73	20
Civil servant, gen, *h*	64	54	Representation, pcomp_of, *h*	82	27
Bloc, gen, *h*	30	77	Patriarch, pcomp_of, *h*	128	28
Nonaligned, mod, *m*	55	60	Friendly, mod, *m*	58	83
Neighboring, mod, *m*	15	165	Expel, pcomp_from, *h*	59	30
Statistic, pcomp_on, *h*	165	43	Heartland, pcomp_of, *h*	102	23
Border, pcomp_of, *h*	10	247	Surprising, pcomp_of, *h*	114	38
Northwest, mod, *h*	41	174	Issue, pcomp_between, *h*	103	51
Trip, pcomp_to, *h*	105	34	Contravention, pcomp_in, *m*	129	43

Table 8
Top-10 features of *country* by the Bootstrapped feature weighting.

Feature	$Weight^B$
Industry, gen, *h*	1.21
Airport, gen, *h*	1.16
Visit, pcomp_to, *h*	1.06
Neighboring, mod, *m*	1.04
Law, gen, *h*	1.02
Economy, gen, *h*	1.02
Population, gen, *h*	0.93
Stock market, gen, *h*	0.92
Governor, pcomp_of, *h*	0.92
Parliament, gen, *h*	0.91

most characteristic features, which are in turn expected to appear at the higher ranks of each feature vector. The common features for non- similar words are expected to be scattered all across each of the vectors. In fact, these expectations correspond exactly to the rationale behind distributional similarity measures: such measures are designed to assign higher similarity scores for vector pairs that share highly weighted features.

Comparatively, we illustrate the behavior of the *Bootstrapped LIN* method relative to the observations regarding the original *LIN* method, using the same running example. Table 8 shows the top-10 features of *country*. We observe that the list now contains features that are intuitively quite indicative and reliable, while many too specific or idiomatic features, and too general ones, were demoted (compare with Table 3). Table 9 shows that most of the top-10 common features for *country–state* are now ranked highly for both words. On the other hand, there are only two common features (amongst the top 100 features) for the incorrect pair *country–party*, both with quite low ranks (compare

Table 9
Bootstrapped weighting: top-10 common features for *country–state* and *country–party* along with their corresponding ranks in the two (sorted) feature vectors.

Country–State	Ranks		*Country–Party*	Ranks	
Neighboring, mod, *m*	3	1	Relation, pcomp_with, *h*	12	26
Industry, gen, *h*	1	11	Minister, pcomp_from, *h*	77	49
Impoverished, mod, *m*	8	8			
Governor, pcomp_of, *h*	10	9			
Population, gen, *h*	6	16			
City, gen, *h*	17	18			
Economy, gen, *h*	5	15			
Parliament, gen, *h*	10	22			
Citizen, pcomp_of, *h*	14	25			
Law, gen, *h*	4	33			

Table 10
Top-20 most similar words for *country* and their ranks in the similarity list by the *Bootstrapped LIN* measure. Note that four of the incorrect similarities from Table 2 were replaced with correct entailments resulting in 20% increase of precision (reaching 60%).

nation	1	territory	6	*province	11	zone	16
state	2	*neighbor	7	*city	12	land	17
*island	3	colony	8	*town	13	place	18
region	4	*port	9	kingdom	14	economy	19
area	5	republic	10	*district	15	*world	20

with Table 7), while the rest of the common features for this pair did not pass the top-100 cutoff.

Consequently, Table 10 demonstrates a much more accurate similarity list for *country*, where many incorrect (non-entailing) word similarities, like *party* and *company*, were demoted. Instead, additional correct similarities, like *kingdom* and *land*, were promoted (compare with Table 2). In this particular case all the remaining errors correspond to words that are related quite closely to *country*, denoting geographic concepts. Many of these errors are context dependent entailments which might be substitutable in some cases, but they violate the word meaning entailment condition (e.g., *country–neighbor*, *country–port*). Apparently, these words tend to occur in contexts that are typical for *country* in the Reuters corpus. Some errors violating the substitutability condition of lexical entailment were identified as well, such as *industry–product*. These cases are quite hard to differentiate from correct entailments, since the two words are usually closely related to each other and also share highly ranked features, since they often appear in similar characteristic contexts. It may therefore be difficult to filter out such non-substitutable similarities merely by the standard distributional similarity scheme, suggesting that additional mechanisms and data types would be required.

6.2 The average common-feature rank ratio

It should be noted at this point that the above observations regarding feature weight behavior are based on subjective intuition of how *characteristic* features are for a word meaning, which is quite difficult to assess systematically. Therefore, we next propose a quantitative measure for analyzing the quality of feature vector weights.

More formally, given a pair of feature vectors for words w and v we first define their **average common-feature rank** with respect to top-n common features, denoted $acfr_n$, as follows:

$$acfr_n(w,v) = \frac{1}{n}\sum_{f\in top-n(F(w)\cap F(v))}\frac{1}{2}[rank(w,f)+rank(v,f)] \quad (7)$$

where *rank*(*w*,*f*) is the rank of feature f in the vector of the word w when features are sorted by their weight, and *F*(*w*) is the set of features in w's vector. *top-n* is the set of top-n common features to consider, where common features are sorted by the sum of their weights in the two word vectors (the same sorting as in Table 7). In other words, $acfr_n(w,v)$ is the average rank in the two feature vectors of their top-n common features.

Using this measure, we expect that a good feature weighting function would typically yield lower values of $acfr_n$ for truly similar words (as low ranking values correspond to higher positions in the vectors) than for non-similar words. Hence, given a pre-judged test set of pairs of similar and non-similar words, we define the ratio, *acfr-ratio*, between the average $acfr_n$ of the set of all the non-similar words, denoted as *Non-Sim*, and the average $acfr_n$ of the set of all the known pairs of similar words, *Sim*, to be an objective measure for feature weighting quality, as follows:

$$acfr_n - ratio = \frac{\frac{1}{|Non-Sim|}\sum_{w,v\in Non-Sim} acfr_n(w,v)}{\frac{1}{|Sim|}\sum_{w,v\in Sim} acfr_n(w,v)} \quad (8)$$

As an illustration, the two word pairs in Table 7 yielded $acfr_{10}(country, state) = 78$ and $acfr_{10}(country, party) = 64$. Both values are quite high, showing no principal difference between the tighter lexically entailing similarity versus a pair of non-similar (or rather loosely related) words. This behavior indicates the deficiency of the *MI* feature weighting function in this case. On the other hand, the corresponding values for the above two pairs produced by the *Bootstrapped LIN* method (for the features in Table 9) are $acfr_{10}(country, state) = 12$ and $acfr_{10}(country, party) = 41$. These figures clearly reflect the desired distinction between similar and non-similar words, showing that the common features of the similar words are indeed concentrated at much higher ranks in the vectors than the common features of the non-similar words.

In recent work on distributional similarity (Curran 2004; Weeds and Weir 2005) a variety of alternative weighting functions were compared. However, the quality of these weighting functions was evaluated only through their impact on the performance of a particular word similarity measure, as we did in Section 5. Our *acfr-ratio* measure provides the first attempt to analyze the quality of weighting functions directly, relative to a pre-judged word similarity set, without reference to a concrete similarity measure.

6.3 An empirical assessment of the *acfr-ratio*

In this subsection we report an empirical comparison of the *acfr-ratio* obtained for the *MI* and *BootstrappedLIN* weighting functions. To that end, we have run the Minipar system on the full Reuters RCV1 corpus which contains 2.5GB of English News Stories, and then calculated the *MI*-weighted feature vectors. The optimized threshold on the feature weights, θ_{weight}, was set to 0.2. Further, to compute the *Bootstrapped LIN* feature weights a θ_{sim} of 0.02 was applied on the *LIN* similarity values. In this experiment we employed the full bootstrapped vectors (i.e. without applying feature reduction by the top-100 cutoff). This was done to avoid the effect of the feature vector size on the $acfr_n$ metric, which tends to naturally assign higher scores to shorter vectors.

As computing the *acfr-ratio* requires a pre-judged sample of candidate word similarity pairs, we utilized the annotated test sample of candidate pairs of word similarities described in Section 5, which contains both entailing and non-entailing pairs.

First, we computed the average common-feature rank scores ($acfr_n$) (with varying values of n) for $weight_{MI}$ and for $weight^B$ over all the pairs in the test sample. Interestingly, the mean $acfr_n$ scores for $weight^B$ are ranging within *110-264* for n=10..100, while the corresponding range for $weight_{MI}$ is by an order of magnitude higher: *780-1254*, despite the insignificant differences in vector sizes. Therefore, we conclude that the common features that are relevant to establish distributional similarity in general (regardless of entailment) are much more scattered across the vectors by *MI* weighting, while with bootstrapping they tend to appear at higher positions in the vectors. These figures reflect a desired behaviour of the bootstrapping function which concentrates most of the prominent common features for all the distributionally similar words (whether entailing or not) at the lower ranks of their vectors. In particular, this explains the ability of our method to perform a massive feature reduction as demonstrated in Section 5, and to produce more informative vectors, while demoting and eliminating much of the noise in the original vectors.

Next, we aim to measure the discriminative power of the compared methods to distinguish between entailing and non-entailing pairs. To this end we calculated the *acfr-ratio*, which captures the difference in the average common feature ranks between entailing vs. non-entialing pairs, for both the *MI*-based and bootstrapped vectors.

The obtained results are presented in Figure 2. As can be seen the *acfr-ratio* values are consistently higher for *Bootstrapped LIN* than for *MI*. That is, the bootstrapping method assigns much higher $acfr_n$ scores to entailing words than to non-entailing ones, while for *MI* the corresponding $acfr_n$ scores for entailing and non-entailing pairs are roughly equal. In particular, we notice that the largest gaps in *acfr-ratio* occur for lower numbers of top common features, whose weights are indeed the most important and influential in distributional similarity measures. Thus, the above findings suggest a direct indication of an improved quality of the bootstrapped feature vectors.

7. A Pseudo-Word Sense Disambiguation Evaluation

The lexical entailment evaluation reported above corresponds to the lexical substitution application of distributional similarity. The other type of application, as reviewed in the Introduction, is similarity-based prediction of word co-occurrence likelihood, needed for disambiguation applications. Comparative evaluations of distributional similarity methods for this type of application were commonly conducted using a pseudo-word sense disambiguation scheme, which is replicated here. In the next subsections we first describe how distributional similarity can help improving Word Sense Disambiguation

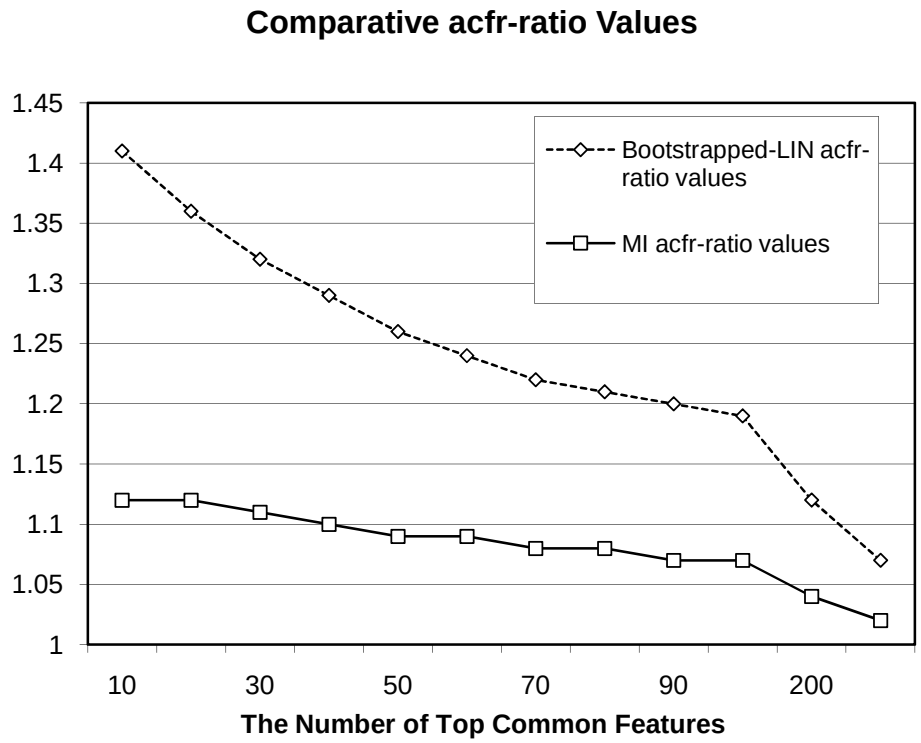

Figure 2
Comparison between the *acfr-ratio* for *MI* and *Bootstrapped LIN* methods, when using varying numbers of common top-ranked features in the words' feature vectors.

(WSD). Then we describe how the pseudo-word sense disambiguation task, which corresponds to the general WSD setting, was used to evaluate the co-occurrence likelihood predictions obtained by alternative similarity methods.

7.1 Similarity modeling for Word Sense Disambiguation

WSD methods need to identify the correct sense of an ambiguous word in a given context. For example, a test instance for the verb *save* might be presented in the context *saving private Ryan*. The disambiguation method must decide whether *save* in this particular context means *rescue*, *preserve*, *keep*, *lay aside* or some other alternative.

Sense recognition is typically based on context features collected from a sense-annotated training corpus. For example, the system might learn from the annotated training data that the word *soldier* is a typical object for the rescuing sense of *save*, as in: *They saved the soldier*. In this setting, distributional similarity is used to reduce the data sparseness problem via similarity-based generalization. The general idea is to predict the likelihood of unobserved word co-occurrences based on observed co-occurrences of distributionally similar words. For example, assume that the noun *private* did not occur as a direct object of *save* in the training data. Yet, some of the words that are distributionally similar to *private*, like *soldier* or *sergeant*, might have occurred with *save*. Thus, a WSD system may infer that the co-occurrence *save private* is more likely for the

rescuing sense of *save* because *private* is distributionally similar to *soldier*, which did co-occur with this sense of *save* in the annotated training corpus. In general terms, the WSD method estimates the co-occurrence likelihood for the target sense and a given context word based on training data for words that are distributionally similar to the context word.

This idea of similarity-based estimation of co-occurrence likelihood was applied in (Dagan, Marcus, and Markovitch 1995) to enhance WSD performance in machine translation and recently in (Gliozzo, Giuliano, and Strapparava 2005), who employed an LSA-based kernel function as a similarity-based representation for WSD. Other works employed the same idea for pseudo-word sense disambiguation, as explained in the next subsection.

7.2 The pseudo-word sense disambiguation setting

Sense disambiguation typically requires annotated training data, created with considerable human effort. Yarowsky (1992) suggested that when using WSD as a test bed for comparative algorithmic evaluation it is possible to set up a pseudo-word sense disambiguation scheme. This scheme was later adopted in several experiments, and was popular for comparative evaluations of similarity-based co-occurrence likelihood estimation (Dagan, Lee, and Pereira 1999; Lee 1999; Weeds and Weir 2005). We followed closely the same experimental scheme, as described below.

First, a list of pseudo-words is constructed by "merging" pairs of words into a single pseudo word. In our experiment each pseudo-word constitutes of a pair of randomly chosen verbs, (v, v'), where each verb represents an alternative "sense" of the pseudo-word. The two verbs are chosen to have almost identical probability of occurrence, which avoids a word frequency bias on the co-occurrence likelihood predictions.

Next, we consider occurrences of pairs of the form $\langle n,$ *(*v*,* v'*)*$\rangle$, where (v, v') is a pseudo-word and n is a noun representing the object of the pseudo-word. Such pairs are constructed from all co-occurrences of either v or v' with the object n in the corpus. For example, given the pseudo-word *(rescue, keep)* and the verb-object co-occurrence in the corpus *rescue–private* we construct the pair ⟨*private, (rescue, keep)* ⟩. Given such a test pair, the disambiguation task is to decide which of the two verbs is more likely to co-occur with the given object noun, aiming to recover the original verb from which this pair was constructed. In this example we would like to predict that *rescue* is more likely to co-occur with *private* as an object than *keep*.

In our experiment 80% of the constructed pairs were used for training, collecting the co-occurrence statistics for the original known verb in each pair (i.e., either $\langle n,\ v\rangle$ or $\langle n,\ v'\rangle$ *)*. From the remaining 20% of the pairs those occurring in the training corpus were discarded, leaving as a test set only pairs which do not appear in the training part. Thus, predicting the co-occurrence likelihood of the noun with each of the two verbs cannot rely on direct frequency estimation for the co-occurrences, but rather only on similarity-based information.

To make the similarity-based predictions we first compute the distributional similarity scores for all pairs of nouns based on the training set statistics, where the co-occurring verbs serve as the features in the distributional vectors of the nouns. Then, given a test pair $\langle$*(*v*,*v'*),* $n\rangle$ our task is to predict which of the two verbs is more likely to co-occur with n. This verb is thus predicted as being the original verb from which the pair was constructed. To this end, the noun n is substituted in turn with each of its k distributionally most similar nouns, n_i, and then both of the obtained "similar" pairs $\langle n_i, v\rangle$ and $\langle n_i, v'\rangle$ are sought in the training set.

Table 11
The comparative error rates of the pseudo- disambiguation task for the three examined similarity measures, with and without applying the bootstrapped weighting for each of them.

Measure	LIN–LIN^B	WJ–WJ^B	COS–COS^B
Error rate	0.157–0.133	0.150–0.132	0.155–0.145

Next, we would like to predict that the more likely co-occurrence amongst $\langle n, v\rangle$ and $\langle n, v'\rangle$ is the one for which more pairs of similar words were found in the training set. Several variants were used in the literature to quantify this decision procedure and we have followed the most recent one from (Weeds and Weir 2005). Each similar noun n_i is given a vote, which is equal to the difference between the frequencies of the two co-occurrences (n_i, v) and (n_i, v'), and which it casts to the verb with which it co-occurs more frequently. The votes for each of the two verbs are summed over all k similar nouns n_i and the one with most votes wins. The winning verb is considered correct, if it is indeed the original verb from which the pair was constructed, while a tie is recorded if the votes for both verbs are equal. Finally, the overall performance of the prediction method is calculated by its error rate:

$$error = \frac{1}{T}(\#of\ incorrect\ choices + \frac{\#of\ ties}{2}) \tag{9}$$

where T is the number of test instances.

In the experiment we used the 1,000 most frequent nouns in our subset of the Reuters corpus (of Section 5.1). The training and test data were created as described above, using the Minipar parser (Lin 1993) to produce verb-object co-occurrence pairs. The *k=40* most similar nouns for each test noun were computed by each of the three examined similarity measures *LIN, WJ* and *COS* (as in Section 5), with and without bootstrapping. The six similarity lists were utilized in turn for the pseudo- word sense disambiguation task, calculating the corresponding error rate.

7.3 Results

Table 11 shows the error rate improvements after applying the bootstrapped weighting for each of the three similarity measures. The largest error reduction, by over 15%, was obtained for the *LIN* method, with quite similar results for *WJ*. This result is better than the one reported in (Weeds and Weir 2005), who achieved about 6% error reduction compared to *LIN*.

This experiment shows that learning tighter semantic similarities, based on the improved bootstrapped feature vectors, correlates also with better similarity-based inference for co-occurrence likelihood prediction. Furthermore, we have seen once again that the bootstrapping scheme does not depend on a specific similarity measure, reducing the error rates for all three measures.

8. Conclusions

The primary contribution of this article is proposing a bootstrapping method that substantially improves the quality of distributional feature vectors, as needed for statistical word similarity. The main idea is that features which are common for similar words are also most characteristic for their meanings and thus should be promoted. In fact, beyond its intuitive appeal, this idea corresponds to the underlying rationale of the distributional similarity scheme: semantically similar words are expected to share exactly those context features which are most characteristic for their meaning.

The superior empirical performance of the resulting vectors was assessed in the context of the two primary applications of distributional word similarity. The first is lexical substitution, which was represented in our work by a human gold standard for the substitutable lexical entailment relation. The second is co-occurrence likelihood prediction, which was assessed by the automatically computed scores of the common pseudo-word sense disambiguation evaluation. An additional outcome of the improved feature weighting is massive feature reduction.

Experimenting with three prominent similarity measures showed that the bootstrapping scheme is robust and performs well when applied over different measures. Notably, our experiments show that the underlying assumption behind the bootstrapping scheme is valid, that is, available similarity metrics do provide a reasonable approximation of the semantic similarity space which can be then exploited via bootstrapping.

The methodology of our investigation has yielded several additional contributions:

1. Utilizing a refined definition of substitutable lexical entailment both as an end goal and as an analysis vehicle for distributional similarity. It was shown that the refined definition can be judged directly by human subjects with very good agreement. Overall, lexical entailment is suggested as a useful model for lexical substitution needs in semantic-oriented applications.
2. A thorough error analysis of state of the art distributional similarity performance was conducted. The main observation was deficient quality of the feature vectors, which reduces the eventual quality of similarity measures.
3. Inspired by the qualitative analysis, we proposed a new analytic measure for feature vector quality, namely average common-feature rank ratio (*acfr-ratio*), which is based on the common ranks of the features for pairs of words. This measure estimates the ability of a feature weighting method to distinguish between pairs of similar vs. non-similar words. To the best of our knowledge this is the first proposed measure for direct analysis of the quality of feature weighting functions, without the need to employ them within some vector similarity measure.

The ability to identify the most characteristic features of words can have additional benefits, beyond their impact on traditional word similarity measures (as evaluated in this article). A demonstration of such potential appears in (Geffet and Dagan 2005), which presents a novel feature inclusion scheme for vector comparison. That scheme utilizes our bootstrapping method to identify the most characteristic features of a word and then tests whether these particular features co-occur also with a hypothesized entailed

word. The empirical success reported in that paper provides additional evidence for the utility of the bootstrapping method.

More generally, our motivation and methodology can be extended in several directions by future work on acquiring lexical entailment or other lexical-semantic relations. One direction is to explore better vector comparison methods that will utilize the improved feature weighting, as shown in (Geffet and Dagan 2005). Another direction is to integrate distributional similarity and pattern-based acquisition approaches, which were shown to provide largely complementary information (Mirkin, Dagan, and Geffet 2006). Yet, an additional potential is to integrate automatically acquired relationships with the information found in WordNet, which seems to suffer from several serious limitations (Curran 2005), and typically overlaps to a rather limited extent with the output of automatic acquisition methods. As a parallel direction, future research should explore in detail the impact of different lexical-semantic acquisition methods on text understanding applications.

Finally, our proposed bootstrapping scheme seems to have a general appeal for improving feature vector quality in additional unsupervised settings. We thus hope that this idea will be explored further in other NLP and machine learning contexts.

References

Adams, Rod. 2006. Textual Entailment Through Extended Lexical Overlap. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 68–73, Venice, Italy.

Bar-Haim, Roy, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9, Venice, Italy.

Baroni, Marco and S. Vegnaduzzo. 2004. Identifying Subjective Adjectives through Web-based Mutual Information. In *Proceedings of KONVENS–04*, pages 17Ű-24, Vienna, Austria.

Barzilay, Regina and Kathleen McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of ACL / EACL–01*, pages 50–57, Toulouse, France.

Caraballo, Sharon A. 1999. Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text. In *Proceedings of ACL–99*, pages 120–126, Maryland, USA

Chklovski, Timothy and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of EMNLP–04*, pages 33–40, Barcelona, Spain.

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

Curran., James R. 2005. Supersense Tagging of Unknown Nouns using Semantic Similarity. In *Proceedings of ACL–2005*, pages 26–33, Ann Arbor, Michigan.

Curran, James R. 2004. *From Distributional to Semantic Similarity.* Ph.D. Thesis, School of Informatics of the University of Edinburgh, Scotland.

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

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. *Lecture Notes in Computer Science*, 3944:177–190.

Dagan, Ido. 2000. Contextual Word Similarity. In Rob Dale, Hermann Moisl, and Harold Somers, editors, *Handbook of Natural Language Processing*. Marcel Dekker Inc, Chapter 19, pages 459–476.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1999. Similarity-based models of cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, special issue on Natural Language Learning.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1995. Contextual word similarity and estimation from sparse data. *Computer, Speech and Language*, 9:123–152.

Dunning, Ted E. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1):61–74.

Essen, U. and V. Steinbiss. 1992. Co-occurrence smoothing for stochastic language modeling. In *ICASSP–92*, 1:161–164, Piscataway, New Jersey: IEEE.

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

Ferrandez, O., R.M. Terol, R. Munoz, P. Martinez-Barco, and M. Palomar. 2006. An approach based on Logic Forms and WordNet relationships to Textual Entailment performance. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 22–26, Venice, Italy.

Gasperin, Caroline and Renata Vieira. 2004. Using Word Similarity Lists for Resolving Indirect Anaphora. In *Proceedings of ACL–04 Workshop on Reference Resolution*, pages 40–46, Barcelona, Spain.

Gauch, Susan, J. Wang, and S. Mahesh Rachakonda. 1999. A Corpus Analysis Approach for Automatic Query Expansion and its Extension to Multiple Databases. *ACM Transactions on Information Systems (TOIS)*, 17(3):250–269.

Geffet, Maayan. 2006. *Refining the Distributional Similarity Scheme for Lexical Entailment.* Ph.D. Thesis. School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel.

Geffet, Maayan and Ido Dagan. 2005. The Distributional Inclusion Hypotheses and Lexical Entailment. In *Proceedings of ACL–05*, pages 107–114, Ann Arbor, Michigan.

Geffet, Maayan and Ido Dagan. 2004. Feature Vector Quality and Distributional Similarity. In *Proceedings of COLING–04*, Article number: 247, Geneva, Switzerland.

Gliozzo, Alfio, Claudio Giuliano, and Carlo Strapparava. 2005. Domain Kernels for Word Sense Disambiguation. In *Proceedings of ACL–05*, pages 403–410, Ann Arbor, Michigan.

Grefenstette, Gregory. 1994. *Exploration in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA.

Harris, Zelig S. 1968. *Mathematical structures of language*. Wiley, 1968.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL–90*, pages 268–275, Pittsburgh, PA.

Jijkoun, Valentin and Maarten de Rijke. 2005. Recognizing Textual Entailment: Is Word Similarity Enough? In Joaquin Quinonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alche-Buc, editors, *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, Lecture Notes in Computer Science 3944 Springer 2006*, pages 449–460.

Karov, Y. and S. Edelman. 1996. Learning similarity-based word sense disambiguation from sparse data. In E. Ejerhed and I. Dagan, editors, *Fourth Workshop on Very Large Corpora*, pages 42–55. Somerset, New Jersey: Association for Computational Linguistics.

Landis, J. R. and G. G. Koch. 1997. The measurements of observer agreement for categorical data. *Biometrics*, 33:159–174.

Lee, Lillian. 1999. Measures of Distributional Similarity. In *Proceedings of ACL–99*, pages 25–32, Maryland, USA.

Lee, Lillian. 1997. *Similarity-Based Approaches to Natural Language Processing*. Ph.D. Thesis. Harvard University, Cambridge, MA.

Lin, Dekang and Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343–360.

Lin, Dekang. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING / ACL–98*, pages 768–774, Montreal, Canada.

Lin, Dekang. 1993. Principle-Based Parsing without Overgeneration. In *Proceedings of ACL–93*, pages 112–120, Columbus, Ohio, USA.

Luk, Alpha K. 1995. Statistical sense disambiguation with relatively small corpora using dictionary definitions. In *Proceedings of ACL–95*, pages 181–188, Cambridge, MA USA.

Manning, Christopher D. and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, USA.

Mirkin, Shachar, Ido Dagan, and Maayan Geffet. 2006. Integrating Pattern-based and Distributional Similarity Methods for Lexical Entailment Acquisition. In *Proceedings of the COLING / ACL–06 Main Conference Poster Sessions*, pages 579–586, Sydney.

Ng, H. T. 1997. Exemplar-based word sense disambiguation: Some recent improvements. In C. Cardie and R. Weischedel, editors, *Proceedings of EMNLP– 97*, pages 208–213.

Ng, H. T. and H. B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of ACL–1996*, pages 40–47.

Nicholson, Jeremy, Nicola Stokes, and Timothy Baldwin. 2006. Detecting Entailment Using an Extended Implementation of the Basic Elements Overlap Metric. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 122–127, Venice, Italy.

Pado, Sebastian and Mirella Lapata. 2007. Dependency-based Construction of Semantic Space Models. In *Computational Linguistics*, 33(2):161–199.

Pantel, Patrick, D. Ravichandran, and E.Hovy. 2004. Towards Terascale Knowledge Acquisition. In *Proceedings of COLING–04*, Article number: 771, Geneva, Switzerland.

Pantel, Patrick and Deepak Ravichandran. 2004. Automatically Labeling Semantic Classes. In *Proceedings of HLT / NAACL–04*, pages 321–328, Boston, MA.

Pekar, Viktor, M. Krkoska, and S. Staab. 2004. Feature Weighting for Co-occurrence-based Classification of Words. In *Proceedings of COLING–04*, Article number: 799, Geneva, Switzerland.

Pereira, Fernando, Tishby Naftali, and Lee Lillian. 1993. Distributional clustering of English words. In *Proceedings of ACL–93*, pages 183–190, Colombus, Ohio, USA.

Ruge, Gerda. 1992. Experiments on linguistically-based term associations. *Information Processing & Management*, 28(3):317–332.

Salton, G. and M. J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill.

Szpektor, Idan, H. Tanev, Ido Dagan, and B. Coppola. 2004. Scaling Web-based Acquisition of Entailment Relations. In *Proceedings of EMNLP–04*, pages 41–48, Barcelona, Spain.

Vanderwende, Lucy, Arul Menezes, and Rion Snow. 2006. Microsoft Research at RTE-2: Syntactic Contributions in the Entailment Task: an implementation. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 27–32, Venice, Italy.

Weeds, Julie and David Weir. 2005. Co-occurrence Retrieval: A flexible framework for lexical distributional similarity. In *Computational Linguistics*, 31(4):439–476.

Weeds, Julie, D. Weir, and D. McCarthy. 2004. Characterizing Measures of Lexical Distributional Similarity. In *Proceedings of COLING–04*, pages 1015–1021, Switzerland, July.

Widdows, D. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of HLT/NAACL 2003*, pages 197–204, Edmonton, Canada, June.

Yarowsky, D. 1992. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of COLING–92*, pages 454–460, Nantes, France.