# Dependency-based Construction of Semantic Space Models

Sebastian Padó*
Saarland University

Mirella Lapata**
University of Edinburgh

*Traditionally, vector-based semantic space models use word co-occurrence counts from large corpora to represent lexical meaning. In this article we present a novel framework for constructing semantic spaces that take syntactic relations into account. We introduce a formalization for this class of models which allows linguistic knowledge to guide the construction process. We evaluate our framework on a range of tasks relevant for cognitive science and natural language processing: semantic priming, synonymy detection and word sense disambiguation. In all cases, our framework obtains results that are comparable or superior to the state of the art.*

## 1. Introduction

Vector space models of word co-occurrence have proved a useful framework for representing lexical meaning in a variety of natural language processing (NLP) tasks such as word sense discrimination (Schütze 1998) and ranking (McCarthy et al. 2004), text segmentation (Choi, Wiemer-Hastings, and Moore 2001), contextual spelling correction (Jones and Martin 1997), automatic thesaurus extraction (Grefenstette 1994; Lin 1998a), and notably information retrieval (Salton, Wang, and Yang 1975). These models have also been popular in cognitive science and figure prominently in several studies simulating human behavior. Examples include similarity judgments (McDonald 2000), semantic priming (Lund and Burgess 1996; Landauer and Dumais 1997; Lowe and McDonald 2000; McDonald and Brew 2004) and text comprehension (Landauer and Dumais 1997; Foltz, Kintsch, and Landauer 1998).

The popularity of vector-based models in both fields lies in their ability to represent word meaning simply by using distributional statistics. The central assumption here is that the context surrounding a given word provides important information about its meaning (Harris 1968). The semantic properties of words are captured in a multi-dimensional space by vectors that are constructed from large bodies of text by observing the distributional patterns of co-occurrence with their neighboring words. Co-occurrence information is typically collected in a frequency matrix, where each row corresponds to a unique word, commonly referred to as "target word", and each column represents a given linguistic context. The semantic similarity between any two words can then be quantified directly using a distance measure such as cosine or Euclidean distance.

* Computational Linguistics, P.O. Box 15 11 50, 66041 Saarbrücken, Germany. E-mail: pado@coli.uni-sb.de
** School of Informatics, University of Edinburgh, EH8 9LW, Edinburgh, UK. E-mail: mlap@inf.ed.ac.uk

Submission received: 20 December 2004; Revised submission received: 26 September 2006; Accepted for publication: 23 November 2006

© 2005 Association for Computational Linguistics

Contexts are defined as a small number of words surrounding the target word (Lund and Burgess 1996; Lowe and McDonald 2000) or as entire paragraphs, even documents (Salton, Wang, and Yang 1975; Landauer and Dumais 1997). Latent Semantic Analysis (LSA, Landauer and Dumais (1997)) is an example a *document-based* vector space model that is commonly used in information retrieval and cognitive science. Each target word $t$ is represented by a $k$ element vector of paragraphs $p_{1...k}$ and the value of each vector element is a function of the number of times $t$ occurs in $p_i$. In contrast, the Hyperspace Analogue to Language model (HAL, Lund and Burgess (1996)) creates a *word-based* semantic space: each target word $t$ is represented by a $k$ element vector, whose dimensions correspond to context words $c_{1...k}$. The value of each vector element is a function of the number of times each $c_i$ occurs within a window of size $n$ before or after $t$ in a large corpus.

In their simplest incarnation, semantic space models treat context as a set of unordered words, without even taking parts of speech into account (e.g., *to drink* and *a drink* are represented by a single vector). In fact, with the exception of function words (e.g., *the*, *down*), which are often removed, it is often assumed that all context words within a certain distance from the target word are semantically relevant. Since no linguistic knowledge is taken into account, the construction of semantic space models is straightforward and language-independent – all that is needed is a segmented corpus of written or spoken text.

However, the assumption that contextual information contributes indiscriminately to a word's meaning is clearly a simplification. There is ample evidence demonstrating that syntactic relations across and within sentences are crucial for sentence and discourse processing (Fodor 1995; Miltsakaki 2003; Neville et al. 1991; West and Stanovich 1986) and modulate cognitive behavior in sentence priming tasks (Morris 1994). Furthermore, much research in lexical semantics hypothesizes that the behavior of words, particularly with respect to the expression and interpretation of their arguments, is to a large extent determined by their meaning (Talmy 1985; Jackendoff 1983; Goldberg 1995; Levin 1993; Pinker 1989; Green 1974; Gropen et al. 1989; Fillmore 1965).

It is therefore not surprising that there have been efforts to enrich vector-based models with morpho-syntactic information. Extensions range from part of speech tagging (Widdows 2003; Kanejiya, Kumar, and Prasad 2003) to shallow syntactic analysis (Grefenstette 1994; Curran and Moens 2002; Lee 1999) and full-blown parsing (Lin 1998a). In these semantic space models, contexts are defined over words bearing a syntactic relationship to the target words of interest. This makes semantic spaces more flexible, different types of contexts can be selected, words do not have to co-occur within a small, fixed word window, and word order or argument structure differences can be naturally mirrored in the semantic space.

This article proposes a general framework for semantic space models which conceptualizes context in terms of syntactic relations. We introduce an algorithm for constructing semantic space models from texts annotated with syntactic information (specifically dependency relations) and illustrate how different model *classes* can be derived from this linguistically rich representation. Our guiding hypothesis is that syntactic structure in general and argument structure in particular is a close reflection of lexical meaning (Levin 1993). We thus model meaning by quantifying the degree to which words are attested in similar syntactic environments. The expressive power of our framework stems from three novel parameters which guide model construction. The first parameter determines which types of syntactic structures contribute towards the representation of lexical meaning. The second parameter allows us to weigh the relative importance of different syntactic relations. Finally, the third parameter determines how the semantic

space is actually represented, for instance as co-occurrences of words with other words, words with parts of speech, or words with argument relations (e.g., subject, object).

We evaluate our framework on tasks relevant for cognitive science and NLP. We start by simulating semantic priming, a phenomenon that has received much attention in computational psycholinguistics and is typically modeled using word-based semantic spaces (Landauer and Dumais 1997; McDonald and Brew 2004). We next consider the problem of recognizing synonyms by selecting an appropriate synonym for a target word from a set of semantically related candidate words. Specifically, we evaluate the performance of our model on synonym questions from the *Test of English as a Foreign Language* (TOEFL). These are routinely used as a testbed for assessing how well vector-based models capture lexical knowledge (Landauer and Dumais 1997; Turney 2001; Sahlgren 2006). Our final experiment concentrates on unsupervised word sense disambiguation (WSD), thereby exploring the potential of the proposed framework for NLP applications requiring large scale semantic processing. We automatically infer predominant senses in untagged text by incorporating our syntax-based semantic spaces into the modeling paradigm proposed by McCarthy et al. (2004). In all cases, we show that our framework consistently outperforms word-based models yielding results that are comparable or superior to state of the art.

Our contributions are threefold: a novel framework for semantic spaces that incorporates syntactic information in the form of dependency relations and generalizes previous syntax-based vector-based models; an application of this framework to a wide range of tasks relevant to cognitive modeling and NLP; and an empirical comparison of our dependency-based models against state-of-the-art word-based models.

In Section 2, we give a brief overview of existing word-based and syntax-based models. In Section 3, we present our modeling framework and relate it to previous work. Section 4 discusses the parameter settings for our experiments. Section 5 details our priming experiment, Section 6 presents our study on the TOEFL synonymy task, and Section 7 describes our sense ranking experiment. Discussion of our results and future work concludes the article (Section 8).

## 2. Overview of Semantic Space Models

### 2.1 Word-based and Syntax-based Models

To facilitate comparisons with our framework, we begin with a brief overview of existing semantic space models. We describe traditional word-based co-occurrence models as exemplified in Lowe (2001), Lowe and McDonald (2000), McDonald (2000), and Levy and Bullinaria (2001) as well as syntax-based models as presented in Grefenstette (1994) and Lin (1998a).

Lowe (2001) defines a semantic space model as a quadruple $\langle B, A, S, V \rangle$. $B$ is the set $b_{1\ldots D}$ of *basis elements*, the dimensions of the space. $B$ can be a set of words (Lund and Burgess 1996) or lemmas (McDonald 2000), words with their parts of speech (Widdows 2003) or words with a syntactic relation such as subject or object (Lin 1998a). Usually, the dimensionality of the matrix is restricted to a relatively small number. A popular choice are the $k$ most frequent words (minus the stop words) in a corpus, typically 100–2,000 (McDonald 2000; Levy and Bullinaria 2001). $A$ is a *lexical association function* applied to the co-occurrence frequency of target word $t$ with basis element $b$ so that each word is represented by a vector $\vec{v} = \langle A(f(t, b_1)), A(f(t, b_2)), \ldots, A(f(t, b_n)) \rangle$. If $A$ is the identity function, the raw frequencies are used. Functions such as mutual information or the log-likelihood ratio are often applied to factor out co-occurrences due to chance.

| | lorry | might | carry | sweet | apples |
|---|---|---|---|---|---|
| lorry | 0 | 1 | 1 | 0 | 0 |
| carry | 1 | 1 | 0 | 1 | 1 |
| sweet | 0 | 1 | 0 | 1 | 1 |
| fruit | 0 | 0 | 0 | 0 | 0 |

**Figure 1**
Word-based semantic space (symmetric window size 2)

$S$ is a similarity measure that maps pairs of vectors onto a continuous-valued scale of contextual similarity. $V$ is an optional transformation that reduces the dimensionality of the semantic space. Singular value decomposition (SVD; Berry, Dumais, and O'Brien (1994); Golub and Loan (1989)) is commonly used for this purpose. SVD can be thought of as a means of inferring latent structure in distributional data, while making sparse matrices more informative. For the rest of this article, we will ignore $V$ and other statistical transformations and concentrate primarily on ways of inducing structure from grammatical and syntactic information.

To illustrate this definition, we construct a word-based semantic space for the target words $T = \{lorry, carry, sweet, fruit\}$, using as our corpus the following sentence: *A lorry might carry sweet apples*. For a word-based space, we might use the basis elements $B = \{lorry, might, carry, sweet, apples\}$, a symmetric window of size 2, and identity as the association function $A$. Each target word $t_i \in T$ will then be represented by a five-dimensional row vector, and the value of each vector element will record the number of times each basis element $b_i \in B$ occurs within a window of two words to the left and two words to the right of the target word $t_i$. The co-occurrence matrix that we obtain according to these specifications is shown in Figure 1. A variety of distance measures can be used to compute the similarity $S$ between two target words (see Lee (1999) for an overview), the cosine being the most popular:

$$sim_{cos}(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1)$$

Syntax-based semantic space models (Grefenstette 1994; Lin 1998a) go beyond mere co-occurrence by capturing syntactic relationships between words such as subject-verb or modifier-noun, irrespectively of whether they are physically adjacent or not. The basis elements are generally assumed to be tuples $(r, w)$ where $w$ is a word occurring in relation type $r$ with a target word $t$. The relations typically reflect argument structure (e.g., subject, object, indirect object) or modification (e.g., adjective-noun, noun-noun) and can be obtained via shallow syntactic processing (Grefenstette 1994; Lee 1999; Curran and Moens 2002) or full parsing (Lin 1998a; Curran and Moens 2002; Curran 2004). The basis elements $(r, w)$ are treated as a single unit and are often called attributes (Grefenstette 1994; Curran and Moens 2002) or features (Lin 1998a).

Figure 2 shows a syntax-based semantic space in the manner of Grefenstette (1994), using the basis elements (*subj,lorry*), (*aux,might*), (*mod,sweet*), and (*obj,apples*). The binary association function $A$ records whether the target word possesses the feature

| | (subj,lorry) | (aux,might) | (mod,sweet) | (obj,apples) |
|---|---|---|---|---|
| lorry | | | | |
| carry | x | x | | x |
| sweet | | | | |
| fruit | | | | |

**Figure 2**
Grefenstette's (1994) semantic space

| | (subj,lorry) | (aux,might) | (mod,sweet) | (obj,apples) |
|---|---|---|---|---|
| lorry | 0 | 0 | 0 | 0 |
| carry | 1 | 1 | 0 | 1 |
| sweet | 0 | 0 | 0 | 0 |
| fruit | 0 | 0 | 0 | 0 |

**Figure 3**
Lin's (1988a) semantic space

(denoted by x in Figure 2) or not. Since the cells of the matrix do not contain numerical values, a similarity measure that is appropriate for categorical values must be chosen. Grefenstette (1994) uses a weighted version of Jaccard's coefficient, a measure of association commonly employed in information retrieval (Salton and McGill 1983). Assuming $Attr(t)$ is the set of basis elements co-occurring with $t$, Jaccard's coefficient is defined as:

$$sim_{Jacc}(t_1, t_2) = \frac{Attr(t_1) \cap Attr(t_2)}{Attr(t_1) \cup Attr(t_2)} \tag{2}$$

Lin (1998a) constructs a semantic space similar to Grefenstette (1994) except that the matrix cells represent the number of times a target word $t$ co-occurs with basis element $(r, w)$, as shown in Figure 3. He proposes an information theoretic similarity measure based on the distribution of target words and basis elements:

$$sim_{lin}(t_1, t_2) = \frac{\sum_{(r,w) \in T(t_1) \cap T(t_2)} I(t_1, r, w) + I(t_2, r, w)}{\sum_{(r,w) \in T(t_1)} I(t_1, r, w) + \sum_{(r,w) \in T(t_2)} I(t_2, r, w)} \tag{3}$$

where $I(t, r, w)$ is the mutual information between $t$ and $r, w$ and $T(t)$ is the set of basis elements $(r, w)$ such that $I(t, r, w)$ is positive and:

$$I(t, r, w) = \log \frac{P(t, r, w)P(r)}{P(w, r)P(t, r)} = \log \frac{P(w|r, t)}{P(w|r)} \tag{4}$$

### 2.2 Discussion

Since syntax-based models capture more linguistic structure than word-based models, they should at least in theory provide more informative representations of word mean-

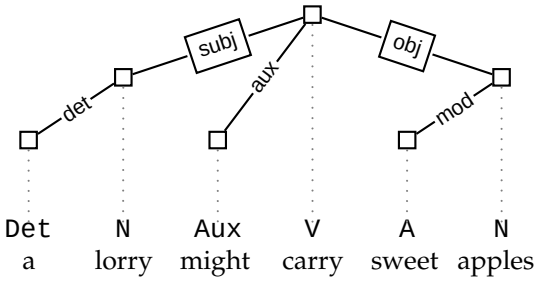
ing. Unfortunately, comparisons between the two types of models have been few and far between in the literature. Furthermore, the potential of syntax-based models has not been fully realized since most previous approaches limit themselves to a specific model class (Grefenstette 1994; Lin 1998a; Curran and Moens 2002; Lin and Pantel 2001). This section discusses these issues in more detail and sketches how we plan to address them.

*Modeling of syntactic context.* All existing syntax-based semantic space models we are aware of incorporate syntactic information in a rather limited fashion. For example, the construction of the space is either based on all relations (Grefenstette 1994; Lin 1998a) or a fixed subset (Lee 1999), but there is no quantitative distinction between different relations. Even in cases where many relations are used (Lin 1998a; Lin and Pantel 2001), only direct relations are taken into account, ignoring potentially important co-occurrence patterns between, for instance, the subject and the object of a verb, or between a verb and its non-local argument (e.g., in control structures).

*Comparison between model classes.* Syntax-based vector space models have been used in NLP for a variety of lexicon acquisition tasks ranging from thesaurus extraction (Grefenstette 1994; Lin 1998a) to paraphrase identification (Lin and Pantel 2001) and collocation discovery (Lin 1999; Bannard, Baldwin, and Lascarides 2003; McCarthy, Keller, and Carroll 2003). Comparisons between word-based and syntax-based models on the same task are rare, and the effect of syntactic knowledge has not been rigorously investigated or quantified. The few studies on this topic reveal an inconclusive picture. On the one hand, Grefenstette (1994) compared the performance of the two classes of models on the task of automatic thesaurus extraction and found that a syntactically enhanced model gave significantly better results over a simple word co-occurrence model. A replication of Grefenstette's (1994) study with a more sophisticated parser (Curran and Moens 2002) revealed that additional syntactic information yields further improvements. On the other hand, attempts to generate more meaningful indexing terms for information retrieval (IR) using syntactic analysis (Salton and Smith 1989; Strzalkowski 1999; Henderson et al. 2002) have been largely unsuccessful. Experimental results show minimal differences in retrieval effectiveness at a substantially greater processing cost (see Voorhees (1999) for details).

*Impact on cognitive modeling.* Despite their widespread use in NLP, syntax-based semantic spaces have attracted little attention in cognitive science and computational psycholinguistics. Wiemer-Hastings and Zipitria (2001) construct a semantic space similar to LSA, but enhanced with part-of-speech tags with the aim of modeling human raters in an intelligent tutoring context. Their results however show that the *tagged LSA* space yields worse performance than a word-based model. Kanejiya, Kumar, and Prasad (2003) attempt to capture syntactic context in a shallow manner by enhancing target words with the parts-of-speech of their immediately preceding words. They argue that this representation can provide useful information for the upcoming target words, as is often the case in language modeling and left-to-right parsing. They employ a document-based semantic space which they submitt to SVD and subsequently compare against an LSA model that contains no syntactic information, again in the context of an intelligent tutoring system. Their results indicate that the syntactically enhanced model has better coverage than the LSA model (i.e., it is able to evaluate more student answers), although it displays a lower correlation with human raters than raw LSA.

In this article, we argue the case for investigating dependency-based semantic space models in more depth. We provide a general definition these models which incorporates




```
a        [Det,det,N]   lorry
lorry    [N,subj,V]    carry
might    [Aux,aux,V]   carry
apples   [N,obj,V]     carry
sweet    [A,mod,N]     apples
```

**Figure 4**
A dependency analysis of the sentence *A lorry might carry sweet apples* as parse tree (left) and set of head-relation-modifier triples (right).

a wider range of syntactic relations than previously considered and subsumes existing syntax-based and word-based models. In order to demonstrate the scope of our framework, we evaluate our models on tasks popular in both cognitive science and NLP. Furthermore, in all cases we report comparisons against state of the art word-based models and show that the additional processing cost incurred by syntax-based models is worth-while.

## 3. A General Framework for Semantic Space Models

Once we move away from words as the basic context unit, the issue of representation of syntactic information becomes pertinent. An ideal syntactic formalism should abstract over surface word order, mirror semantic relationships as closely as possible, and incorporate word-based information in addition to syntactic analysis. It should be also applicable to different languages. These requirements point towards *dependency grammar*, which can be considered as an intermediate layer between surface syntax and semantics. More formally, dependency relations are asymmetric binary relationships between a head and a modifier (Tesnière 1959). The structure of a sentence is analyzed as a directed graph whose nodes correspond to words. The graph's edges correspond to dependency relationships and each edge is labeled with a specific relationship type (e.g., subject, object).

The dependency analysis for the sentence *A lorry might carry sweet apples* is given in Figure 4. On the left side, the sentence is represented as a graph. The sentence head is the main verb *carry* which is modified by its subject *lorry*, its object *apples* and the auxiliary *might*. The subject and object are modified respectively by a determiner (*a*) and an adjective (*sweet*). On the right side of Figure 4, an adjacency matrix notation is used. Edges in the graph are represented as triples of a dependent word (e.g., *lorry*), a dependency label (e.g. `N:subj:V`), and a head word (e.g., *carry*). The dependency label consists of the part of speech of the modifier (capitalized, e.g., `N`) , the dependency relation itself (in lower case, e.g., `subj`), and the part of speech of the head (also capitalized, e.g., `V`).

It is combinations of dependencies like the ones in Figure 4 that will form the context over which the semantic space will be constructed. We base our discussion and experiments on the broad-coverage dependency parser MINIPAR, version 0.5 (Lin

**Table 1**
Summary of notation

| | |
|---|---|
| $b \in B$ | Basis element |
| $t \in T$ | Target word type |
| $W(t)$ | Set of tokens of target type $t$ |
| $M[t][b] \in \mathbb{R}$ | Cell of semantic space matrix for target word $t$ and basis element $b$ |
| $\pi$ | Dependency path (in a given dependency tree) |
| $\Pi$ | Set of all undirected paths |
| $\Pi_s$ | Set of all undirected paths in sentence $s$ |
| $\Pi_t$ | Set of all undirected paths in a sentence anchored at word $t$ |
| $start(\pi), end(\pi)$ | First and last node of an undirected path |
| $Cat$ | Set of POS categories (for given parser) |
| $R$ | Set of dependency relations (for given parser) |
| $l : \Pi \rightarrow (Cat \times R \times Cat)^*$ | Edge (sequence) labeling function |
| $cont : T \rightarrow 2^{\Pi}$ | Local context selection function (subset of paths) |
| $\mu : \Pi \rightarrow B$ | Basis element mapping function |
| $v : \Pi \rightarrow \mathbb{R}$ | Path value function |
| $A : \mathbb{R}^4 \rightarrow \mathbb{R}$ | Lexical association function |

1998a, 2001). However, there is nothing inherent in our formalization that restricts us to this particular parser. Any other parser with broadly similar dependency output (e.g., Briscoe and Carroll (2002)) could serve our purposes.

In the remainder of this section, we first give a non-technical description of our algorithm for the construction of semantic spaces. Then, we proceed to discuss each construction step (context selection, basis mapping, and quantification of co-occurrences) in more detail. Finally, we show how our framework subsumes existing models. Table 1 lists the notation we use in the rest of the article.

### 3.1 The construction algorithm

Our algorithm for creating semantic space models is summarized in Figure 5. Central in the construction process is the notion of *paths*, namely sequences of dependency edges extracted from the dependency parse of a sentence (we define paths formally in Section 3.2). Consider again the graph in Figure 4. Besides individual edges (i.e., paths of length 1), it contains several longer paths, such as the path between *lorry* and *sweet* ($\langle lorry, carry, apples, sweet \rangle$), the path between *a* and *carry* ($\langle a, lorry, carry \rangle$), the path between *lorry* and *carry* ($\langle lorry, carry \rangle$), etc. The usage of paths allows us to represent direct and indirect relationships between words and gives rise to three novel parameters:

1. The **context selection function** $cont(t)$ determines which paths in the graph contribute towards the representation of target word $t$. For example we may choose to consider only paths of length 1, or paths with length $\geq 3$. The function is effectively a syntax-based generalization of the traditional "window size" parameter.
2. The **path value function** $v$ assigns weights to paths, thus allowing linguistic knowledge to influence the construction of the space. For

1: $\forall$ basis element $b$: $\forall$ target $t$: initialize matrix cell $M[t][b]$ with 0
2: **for** every target word $t$ **do**
3: **for** every token $w$ in the set $W(t)$ **do**
4: Compute local context $cont(w)$
5: **for** every path $\pi$ in the set of paths $cont(w)$ **do**
6: Identify relevant basis element $b$ by computing basis mapping function $b = \mu(\pi)$
7: Increment $M[t][b]$ by path value $v(\pi)$
8: **end for**
9: **end for**
10: **end for**
11: Apply lexical association function $A$ to each count in $M$

**Figure 5**
Algorithm for construction of semantic space

instance, it can be used to discount longer paths, or give more weight to paths containing subjects and objects as opposed to determiners or modifiers.

3. The **basis mapping function** $\mu$ creates the dimensions of the semantic space. Although paths themselves could serve as dimensions, the resulting co-occurrence matrix would be overly sparse (this is especially true for lexicalized paths whose number can become unwieldy when parsing a large corpus). For this reason, the basis elements forming the dimensions of the space are defined *independently* from the path construction. The basis mapping function maps paths onto basis elements by collapsing paths deemed functionally equivalent. For instance, we may consider paths carrying the same dependency relations as equivalent, or paths ending in the same word. We thus disassociate the definition of context entities (paths) from the dimensions of the final space (basis elements).

As discussed in Section 2, the main difference among variants of semantic space models lies in the specification of basis elements $B$. By treating the dependency paths as distinct from the basis elements, we obtain a general framework for vector-based models which can be parametrized for different tasks and allows for the construction of spaces with basis elements consisting of words, syntactic entities, or combinations of both. This flexibility, in conjunction with the context selection and path value functions, allows our model to subsume both traditional word-based and syntax-based models (see Section 3.6 for more discussion).

### 3.2 Step 1: Building the context

The first step in constructing a semantic space from a large collection of dependency relations is to define an appropriate *syntactic context* for the target words of interest. We define contexts as *anchored* paths, i.e., paths in a dependency graph that start at a particular target word $t$. Our assumption is that the set of paths anchored at $t$ is a superset of the paths that can contribute relevant distributional information about $t$.

**Definition 1.** The *dependency parse* $p$ of a sentence $s$ is a directed graph $p_s = (V_s, E_s)$, where $E_s \subseteq V_s \times V_s$. The nodes $v \in V_s$ are labeled with individual words $w_i$. For simplicity, we use nodes and their labels interchangeably, and the set of nodes corresponds to the words of the sentence: $V_s = \{w_1, \ldots, w_n\}$. Each edge $e \in E_s$ bears a label $l : E_s \rightarrow Cat \times R \times Cat$ where $Cat$ belongs to a set of POS tags and $R$ to a set of dependency relations. We assume that this set is finite and parser-specific[1]. We write edge labels in square brackets. `[Det,det,N]` and `[N,subj,V]` are examples for labels provided by MINIPAR (see Figure 4, right hand side).

We are now ready to define paths in our dependency graph, save one important issue: should we confine ourselves to directed paths or perhaps disregard the direction of the edges? In a dependency graph, directed paths can only capture the relationship between a head and its (potentially transitive) dependents (e.g., *carry* and *sweet* in Figure 4). This excludes informative contexts representing for instance the relationship between the subject and the object of a predicate (e.g., *lorry* and *apples* in Figure 4). Our intuition is therefore that directed paths would limit the context too severely. In the following, we assume *undirected* paths:

**Definition 2.** An (undirected) *path* $\pi$ is an ordered tuple of nodes $\langle v_0, \ldots, v_n \rangle \in V_s^*$ for some sentence $s$ which meets the following two constraints:

$$\forall i : (v_{i-1}, v_i) \in E_s \vee (v_i, v_{i-1}) \in E_s \quad \text{(connectedness)}$$
$$\forall i \forall j : i \neq j \Rightarrow v_i \neq v_j \quad \text{(cycle-freeness)}$$

In the rest of the article, we use the term path as a shorthand for undirected path.

**Definition 3.** A path $\pi$ is *anchored* at a word $t$ iff $start(\pi) = t$. We write $\Pi_t \subseteq \Pi_s$ for the set of all paths anchored at $t$ in sentence $s$.

As an example, the set of paths anchored at *lorry* in Figure 4 is:

$$\{\langle lorry, carry \rangle, \langle lorry, a \rangle, \quad \text{(two paths of length 1)}$$
$$\langle lorry, carry, apples \rangle, \langle lorry, carry, might \rangle, \quad \text{(two paths of length 2)}$$
$$\langle lorry, carry, apples, sweet \rangle\} \quad \text{(one path of length 3)}$$

**Definition 4.** The *context selection function* $cont : W \rightarrow 2^{\Pi_t}$ assigns to a word $t$ a subset of the paths anchored at $t$. We call this subset the *syntactic context* of $t$.

The context selection function allows direct control over the type of linguistic information represented in the semantic space. In traditional vector-based models, the context selection function does not take any syntactic information into account: all paths $\pi$ are selected for which the absolute difference (*abs*) between the positions (*pos*) of the anchor

1 For the sake of simplicity, we use $R$ without a subscript to denote the set of dependency relations provided by MINIPAR. We utilize subscripts to distinguish between general sets (e.g., $E$ for the set of all conceivable edges) and sentence-specific sets (e.g., $E_s$ for the set of edges in the parse tree of sentence $s$).

$start(\pi)$ and the end word $end(\pi)$ does not exceed the window size $k$:

$$cont(t) = \{\pi \in \Pi_t \mid \text{abs}(pos(start(\pi)) - pos(end(\pi))) \leq k\} \tag{5}$$

The dependency-based models proposed by Grefenstette (1994) and Lin (1998a) consider minimal syntactic contexts in the form of individual dependency relations, i.e., dependency paths of length 1:

$$cont(t) = \{\pi \in \Pi_t \mid \; ||\pi|| = 1\} \tag{6}$$

The context selection function as defined above permits the elimination of paths from the semantic space on the basis of linguistic or other information. For example, it can be argued that subjects and objects convey more semantic information than determiners or auxiliaries. We can thus limit our context to the set of all anchored paths consisting exclusively of subject or object dependencies:

$$cont(t) = \{\pi \in \Pi_t \mid l(\pi) \in \{[V, subj, N], [V, obj, N]\}^*\} \tag{7}$$

When this context specification function is applied to the dependency graph in Figure 4, only the edges showed in boxes are retained. The context of *lorry* is thus reduced to two paths: $\langle lorry, carry\rangle$ (length 1) and $\langle lorry, carry, apples\rangle$ (length 2). The paths $\langle lorry, a\rangle$, $\langle lorry, carry, might\rangle$, and $\langle lorry, carry, apples, sweet\rangle$ are omitted since their label sequences (such as `[N,det,Det]` for $\langle lorry, a\rangle$) are disallowed by (7).

### 3.3 Step 2: Basis mapping

The second step in the construction of our semantic space model is to specify its dimensions, the basis elements following Lowe's (2001) terminology.

**Definition 5.** The *basis mapping* function $\mu : \Pi \rightarrow B$ maps paths onto basis elements.

By dissociating dependency paths and basis elements in this way, we decouple the observed syntactic context from its representation in the final semantic space. The basis mapping allows us to exploit underlying relationships among different paths: two paths which are (in some sense) equivalent can be mapped onto the same basis element. The function effectively introduces a partitioning of paths into equivalence classes "labeled" by basis elements, thus offering more flexibility in defining the basis elements of the semantic space.

Traditional co-occurrence models use a *word-based basis mapping*. This means that all paths ending at word $w$ are mapped onto the basis element $w$, resulting in a semantic space with context words as basis elements (recall that all paths in the local context start at the target word):

$$\mu(\pi) = end(\pi) \tag{8}$$

A word-based mapping is also possible when paths are defined over dependency graphs. As an example consider the paths anchored at *lorry* in Figure 4. Using (8), these

paths are mapped to the following basis elements:

$$\begin{aligned} \langle lorry, carry \rangle &\quad carry \\ \langle lorry, a \rangle &\quad a \\ \langle lorry, carry, apples \rangle &\quad apples \\ \langle lorry, carry, might \rangle &\quad might \\ \langle lorry, carry, apples, sweet \rangle &\quad sweet \end{aligned}$$

A different mapping is used in Grefenstette (1994) and Lin (1998a) who consider only paths of length 1. In their case, paths are mapped onto pairs representing a dependency relation $r$ and the end word $w$ (see the discussion in Section 2):

$$\mu(\pi) = (r, end(\pi)) \text{ where } ||\pi|| = 1 \wedge \langle r \rangle = l(\pi) \tag{9}$$

Any plausible and computationally feasible function can be used as basis mapping. However, in this article we restrict ourselves to models which use a word-based basis mapping. The resulting spaces are similar to traditional word-based spaces – both use sets of context words – which allows for direct comparisons between our models and word-based alternatives. Crucially, our models differ from traditional models in the more general treatment of (syntactic) context: only paths in the syntactic context, and not surface co-occurrences, contribute towards counts in the matrix. The context selection function supports inference over classes of basis elements (which in previous models would have been considered distinct) as well as fine-grained control over the types of relationships that enter into the space construction.

### 3.4 Step 3: Quantifying syntactic co-occurrence

The last step in the construction of the dependency-based semantic models is to specify the relative importance (i.e., value) of different paths:

**Definition 6.** The *path value function* $v$ assigns a real number to a path: $v : \Pi \rightarrow \mathbb{R}$.

Traditional models do not exploit this possibility, thus giving equal weight to all paths:

$$v_{plain}(\pi) = 1 \tag{10}$$

The path value function provides additional flexibility for incorporating linguistic information into our framework. Even if two paths are mapped onto the same basis element (by the basis mapping), the path value function can weigh their respective contributions differently. For instance, it could discount longer paths which express indirect relationships between words. An example of such a *length-based path value function* is given in (11). It assigns a value of 1 to paths of length 1 and fractions to longer paths:

$$v_{length}(\pi) = \frac{1}{||\pi||} \tag{11}$$

A more linguistically-informed path value function can be defined by taking into account the obliqueness hierarchy of grammatical relations (Keenan and Comrie 1977). According to this hierarchy subjects are more salient than objects, which in turn are more salient than obliques (e.g., prepositional phrases). And obliques are more salient than genitives. We thus define a linear relation-based weighting scheme that ranks paths according to their most salient grammatical function, without considering their length:

$$v_{gram-rel}(\pi) = \begin{cases} 5 & \text{if } subj \in l(\pi) \\ 4 & \text{if } obj \in l(\pi) \\ 3 & \text{if } obl \in l(\pi) \\ 2 & \text{if } gen \in l(\pi) \\ 1 & \text{else} \end{cases} \tag{12}$$

The path value function assigns a numerical value to each path forming the syntactic context of a token $t$. We can next define the *local co-occurrence frequency* between $t$ and a basis element $b$ as the sum of the path values $v(\pi)$ for all paths $\pi \in cont(t)$ which are mapped onto $b$. Since our semantic space construction algorithm operates over word *types*, we sum the local co-occurrence frequencies for all instances of a target word type $t$ (written as $W(t)$) to obtain its *global co-occurrence frequency*. The latter is a measure of the co-occurrence of $t$ and $b$ over the entire corpus:

**Definition 7.** The global co-occurrence frequency of a basis element $b$ and a target $t$ is function $f : B \times T \rightarrow \mathbb{R}$ defined by

$$f(b,t) = \sum_{w \in W(t)} \sum_{\pi \in cont(w) \wedge \mu(\pi)=b} v(\pi)$$

The global co-occurrence frequency $f(b,t)$ could be used directly as the matrix value $M[b][t]$. However, as Lowe (2001) notes, raw counts are likely to give misleading results. This is due to the non-uniform distribution of words in corpora which will introduce a *frequency bias* so that words with similar frequency will be judged more similar than they actually are. It is therefore advisable to use a lexical association function $A$ to factor out chance co-occurrences explicitly.

Our definition allows an arbitrary choice of lexical association function (see Manning and Schütze (1999) for an overview). In our experiments, we follow Lowe and McDonald (2000) in using the well-known *log-likelihood ratio* $G^2$ (Dunning 1993). We can visualize the computation using a two-by-two contingency table whose four cells correspond to four events (Kilgarriff 2001):

| | t | ¬ t |
|---|---|---|
| b | k | l |
| ¬ b | m | n |

The top left cell records the frequency $k$ with which $t$ and $b$ co-occur (i.e., $k$ corresponds to raw frequency counts). The top right cell $l$ records how many times $b$ is attested with any word other than $t$, the bottom left cell $m$ represents the frequency of any word other than $b$ with $t$, and the bottom right cell $n$ records the frequency of pairs involving neither $b$ nor $t$. The function $G^2 : \mathbb{R}^4 \rightarrow \mathbb{R}$ is defined as:

$$\begin{aligned} G^2(k,l,m,n) = 2(&k\log k + l\log l + m\log m + n\log n \\ &- (k+l)\log(k+l) - (k+m)\log(k+m) \\ &- (l+n)\log(l+n) - (m+n)\log(m+n) \\ &+ (k+l+m+n)\log(k+l+m+n)) \end{aligned} \tag{13}$$

A naive implementation of the log-likelihood ratio would keep track of all four events for each pair $(t, b)$; this strategy would require updating the entire matrix for each path and would render the construction of the space prohibitively expensive. This can be avoided by computing only $k = f(t, b)$, the global co-occurrence frequency, and using the marginal frequencies of paths and targets to estimate $l$, $m$ and $n$ as follows:

$$l = \sum_t f(t,b) - k \qquad m = \sum_b f(t,b) - k \qquad n = \sum_b \sum_t f(t,b) - (k+l+m) \tag{14}$$

For example, $l$ can be computed as the total value of all paths in the corpus which are mapped onto $b$ minus the value of those paths which are anchored at $t$.

### 3.5 Definition of semantic space

Our extended framework of semantic space models can now be formally specified by extending Lowe's (2001) definition from Section 2:

**Definition 8.** A semantic space is a tuple $\langle B, T, M, S, A, cont, \mu, v\rangle$. $B$ is the set of basis elements, $T$ the set of target words, and $M$ is the matrix $M = B \times T$. We write $M[t_j][b_i] \in \mathbb{R}$ for the matrix cell $(i, j)$. $A : \mathbb{R}^4 \to \mathbb{R}$ is the lexical association function, and $S : T \times T \to \mathbb{R}$ the similarity measure. Our additional parameters are the content selection function $cont : T \to 2^{\Pi}$, the basis mapping function $\mu : \Pi \to B$, and the path value function $v : \Pi \to \mathbb{R}$.

Note that the set of target words $T$ can contain either word *types* or word *tokens*. In the preceding definitions, we have assumed that co-occurrence counts are constructed over word types, however the framework can be also used to represent word tokens. In this case, each set of target tokens contains exactly one word ($W(t) = \{t\}$), and the outer summation step in Definition 7 trivially does not apply. We work with type-based spaces in the rest of this article. The use of tokens may be appropriate for other applications such as word sense discrimination (Schütze 1998).

We can now construct a semantic space that illustrates our framework. Consider again the sentence *A lorry might carry sweet apples*. According to Definition 8, in order to construct vectors for the target words $T = \{lorry, might, carry, sweet, fruit\}$, we must provide a context selection function, a basis mapping function and a path value function. The space resulting from a context selection function which considers exclusively subject and object dependencies (see (7)), a word-based basis mapping function (see (8)), and a length-based path value function (see (11)), is shown in Figure 6.

| | lorry | might | carry | sweet | apples |
|---|---|---|---|---|---|
| lorry | 0 | 0 | 1 | 0 | 0.5 |
| might | 0 | 0 | 0 | 0 | 0 |
| carry | 1 | 0 | 0 | 0 | 1 |
| sweet | 0 | 0 | 0 | 0 | 0 |
| fruit | 0 | 0 | 0 | 0 | 0 |

**Figure 6**
A dependency-based semantic space using context selection function (7), basis mapping function (8) and path value function (11)

### 3.6 Discussion

We have proposed a general framework for semantic space models which operates on dependency relations and allows linguistic knowledge to inform the construction of the semantic space. The framework is highly flexible: depending on the context selection and basis mapping functions, semantic spaces can be constructed over words, words and parts of speech, syntactic relations, or combinations of words and syntactic relations. This flexibility unavoidably increases the parameter space of our models, since there is a potentially large number of context selection or path value functions for which semantic spaces can be constructed.

At the same time, this allows us to subsume existing semantic space models in our framework, and facilitates comparisons across different kinds of spaces (compare Figures 1, 3, and 6). Our space is sparser than the word-based space in Figure 1, due to the choice of a more selective context specification function (see (5) and (7)). However, this is expected since our main motivation is to distinguish between informative and uninformative syntactico-semantic relations. Using a minimal context selection function results in a space that contains indisputably valid semantic relations, excluding potentially noisy relations like the one between *might* and *sweet*. By adding richer linguistic information to the context selection function, the space can be expanded in a principled manner. In comparison with previous syntax-based models, which only use direct dependency relations (see (6)), our dependency-based space additionally represents indirect semantic relations (e.g., between *lorry* and *apples*).

A smaller parameter space could have resulted from collapsing the context selection and path value functions into one parameter, for example by defining context selection directly as a function from (anchored) paths to their path values, and thus assigning a value of zero to all paths $\pi \notin cont(t)$. However, we refrained from doing this for two reasons, a methodological and a technical one. On the methodological side, we believe that it makes sense to keep the two concepts of context selection and context weighting distinct. The separation allows us to experiment with different path value functions while keeping the set of paths resulting from context selection constant. On the technical side, the two functions are easier to specify declaratively when kept separately. Also, a separate context selection function can be used to efficiently isolate relevant context paths without having to compute the values for all anchored paths.

The context selection function operates over a subset of dependency paths that are anchored, cycle-free and connected. These three preconditions on paths are meant to reflect linguistic properties of reasonable syntactic contexts while at the same time they guarantee the efficient construction of the semantic space. Anchoredness ensures that all paths are semantically connected to the target; this also means that the search space

can be limited to paths starting at the target word. Cycle-freeness and connectedness exclude linguistically meaningless paths such as paths of infinite length (cycles) or paths consisting of several unconnected fragments. These properties guarantee that context paths can be created incrementally, and that construction terminates.

### 3.7 Runtime and Implementation

Our implementation uses path templates to encode the context selection function (see Appendix A for more details). The runtime of the semantic space construction algorithm presented in Section 3 is $O(max_g \cdot |cont| \cdot t)$ where $max_g$ is the maximal degree of a node in the grammar, $|cont|$ the number of path templates used for context selection, and $t$ the number of target tokens in the corpus. This assumes that $\mu(\pi)$ and $v(\pi)$ can be computed in constant time, which is warranted in practice since most linguistically interesting paths will be of limited length (in our study, all paths have a length of at most four). The linear runtime in the size of the corpus provides a theoretical guarantee that the method is applicable to large corpora such as the British National Corpus (BNC).

A Java implementation of the framework presented in this article is available under the GPL from `http://www.coli.uni-saarland.de/~pado/dv/dv.html`. The system can create dependency spaces from the output of MINIPAR (Lin 1998b, 2001). We also provide an interface for integrating other parsers. The distribution includes a set of prespecified parameter settings, namely the word-based basis mapping function, and the path value and context selection functions used in our experiments.

## 4. Experimental Setup

In this section, we describe the corpus and parser chosen for our experiments. We also discuss our parameter and model choice procedure, and introduce the baseline word-based model which we use for comparison with our approach. Our experiments are next presented in Sections 5–7.

### 4.1 Corpus and Parser

All our experiments were conducted on the British National Corpus (BNC), a 100 million word collection of samples of written and spoken English (Burnard 1995). The corpus represents a wide range of British English including samples from newspapers, magazines, books (both academic and fiction), letters, essays as well as spontaneous conversations, business or government meetings, radio shows, and phone-ins. The BNC has been used extensively in building vector space models for many tasks relevant for cognitive science (Patel, Bullinaria, and Levy 1998; McDonald 2000; McDonald and Brew 2004) and NLP (McCarthy et al. 2004; Weeds 2003; Widdows 2003).

In order to construct dependency spaces, the BNC was parsed with MINIPAR, version 0.5 (Lin 1998b, 2001), a wide-coverage dependency parser. MINIPAR employs a manually constructed grammar and a lexicon derived from WordNet with the addition of proper names (130,000 entries in total). Lexicon entries contain part-of-speech and subcategorization information. The grammar is represented as a network of 35 nodes (i.e., grammatical categories) and 59 edges (i.e., types of dependency relationships). MINIPAR uses a distributed chart parsing algorithm. Grammar rules are implemented as constraints associated with the nodes and edges. When evaluated on the SUSANNE corpus (Sampson 1995), the parser achieved a precision of 89% and a recall of 79% in identifying labeled dependencies (Lin 1998b).

### 4.2 Model Selection

The construction of semantic space models involves a large number of parameters: the dimensions of the space, the size and type of the employed context, the choice of similarity function. A number of studies (Patel, Bullinaria, and Levy 1998; Levy and Bullinaria 2001; McDonald 2000) have explored the parameter space for word-based models in detail, using evaluation benchmarks such as human similarity judgments or synonymy choice tests. The motivation behind such studies is to identify parameters or parameter classes that yield consistently good performance across tasks. To avoid overfitting, exploration of the parameter space is typically performed on a development data set different from the test data (McDonald 2000).

The benchmark dataset collected by Rubenstein and Goodenough (1965) is routinely used in NLP and cognitive science for development purposes, e.g., for evaluating automatic measures of semantic similarity (Budanitsky and Hirst 2001; Resnik 1995; Banerjee and Pedersen 2003) or for exploring the parameter space of vector space models (McDonald 2000). It consists of 65 noun-pairs ranging from highly synonymous (*gem-jewel*) to semantically unrelated (*noon-string*). For each pair, a similarity judgment (on a scale of 0 to 4) was elicited from human subjects. The average rating for each pair represents an estimate of the perceived similarity of the two words. Correlation analysis is often used to examine the degree of linear relationship between the human ratings and the corresponding automatically derived similarity values.

Following previous work, we explored the parameter space of our dependency models on the Rubenstein and Goodenough (1965) dataset. The best performing model was then used in all our subsequent experiments. We expect a dependency model optimized on the semantic similarity task to perform well across other related lexical tasks, which incorporate semantic similarity either directly or indirectly. This is true for all tasks reported in this article, namely priming (Experiment 1), inferring whether two words are synonyms (Experiment 2), and acquiring predominant word senses (Experiment 3). Some performance gains could be expected, if parameter optimization took place separately for each task. However, such a strategy would unavoidably lead to overfitting, especially since our datasets are generally small (see Experiments 1 and 2).

We next detail how parameters were instantiated in our dependency models with an emphasis on the influence of the context selection and path value functions.

*Parameters.* Dependency contexts were defined over a set of 14 dependency relations each of which occurred more than 500,000 times in the BNC and which in total accounted for about 76 million of the 88 million dependency relations found in the corpus. These relations are: `amod` (adjective modifier), `comp1` (first complement), `conj` (coordination), `fc` (finite complement), `gen` (genitive noun modifier), `i` (the relationship between a main clause and a complement clause), `lex-mod` (lexical modifier), `mod` (modifier), `nn` (noun-noun modifier), `obj` (object of a verb), `pcomp-n` (nominal complement of prepositions), `rel` (relative clause), `s` (surface subject), and `subj` (subject of a verb). From these, we constructed three context selection functions (fully described in appendix A), which we implemented as parser-specific templates (one template per non-lexical dependency path):

- `minimum` contexts contain paths of length 1 (27 templates; in Figure 4 *sweet* and *carry* are the `minimum` context for *apples*). This definition of syntactic context considers only direct relations and corresponds to local verbal predicate-argument structure.

- `medium` contexts add to `minimum` contexts dependency paths which model the internal structure of noun phrases (length $\leq 3$; 59 templates). In particular, the `medium` context covers phenomena such as coordination, genitive constructions, noun compounds, and different kinds of modification.
- `maximum` contexts combine all templates defined over the 14 dependency relations described above into a rich context representation (length $\leq 4$; 123 templates).

The context specification functions were combined with the three path value functions introduced in Section 3:

- *plain* ($v_{plain}$, see (10))) assigns the same value (namely 1) to every path. It is the simplest path value function and assumes that all paths are equally important.
- *length* ($v_{length}$, see (11)) implements a length-based weighting scheme: it assigns each path a value inversely proportional to its length, thus giving more weight to shorter paths corresponding to more direct relationships.
- *gram-rel* ($v_{oblique}$, see (12)) uses the obliqueness hierarchy (Keenan and Comrie 1977) to rank paths according to the salience of their grammatical relations. Specifically, each path is assigned the value of its most salient grammatical relation (subjects are more salient than objects, which are more salient than other noun phrases).

The combination of the three context selection and three path value functions yields nine model instantiations[2]. To facilitate comparisons with traditional semantic space models, we used a word-based basis mapping function (see (8)) and the log-likelihood score (see (13)) as our lexical association function. We also created semantic spaces with different dimensions, using the 500, 1,000, and 2,000 most frequent basis elements obtained from the BNC. Finally, we experimented with a variety of similarity measures: cosine, Euclidean distance, $L_1$ norm, Jaccard's coefficient, Kullback-Leibler divergence, skew divergence, and Lin's (1998a) measure[3].

*Results.* The effects of different parameters on modeling semantic similarity (using Rubenstein and Goodenough's (1965) dataset) are illustrated in Tables 2 and 3. We report the Pearson Product Moment Correlation ("Pearson's $r$") between human ratings of similarity and vector-based similarity. Rubenstein and Goodenough report an inter-subject correlation of $r = 0.85$ on the rating task. The latter can be considered an upper bound for what can be expected from computational models. For the sake of brevity, we only report results with 2,000 basis elements, since we found that models with fewer dimensions (e.g., 500 and 1,000) generally obtained worse performance. Lin's (1998a) similarity measure uniformly outperformed all other measures by a large margin. For

2 Since the `minimum` context selection only considers paths of length 1, the combinations `minimum`-*plain* and `minimum`-*length* are identical.

3 The original specification of Lin's distance measure (Equation (3)) assumes relation-word pairs as basis elements. Since we work with a word-based basis mapping, we use a simplified version, where $I(t, r, w)$ reduces to $I(t, w) = \log \frac{P(t,w)}{P(t)P(w)}$.

**Table 2**
Correlations (Pearson's *r*) between elicited similarity and dependency models using the cosine distance, 2,000 basis elements and the log-likelihood association function

| Context \ Path | *plain* | *length* | *gram-rel* |
|---|---|---|---|
| `minimum` | 0.45 | 0.45 | 0.43 |
| `medium` | 0.45 | 0.45 | 0.44 |
| `maximum` | 0.47 | 0.46 | 0.45 |

**Table 3**
Correlations (Pearson's *r*) between elicited similarity and dependency models using Lin's (1998a) similarity measure, 2,000 basis elements and the log-likelihood association function

| Context \ Path | *plain* | *length* | *gram-rel* |
|---|---|---|---|
| `minimum` | 0.58 | 0.58 | 0.58 |
| `medium` | 0.60 | 0.62 | 0.59 |
| `maximum` | 0.56 | 0.59 | 0.55 |

comparison, we also give the results we obtained with the cosine similarity measure (see Table 2).

As can be seen, the *gram-rel* path value function performs generally worse than *length* or *plain*. We suspect that this function is, at least in its present form, too selective, giving a low weight to a large number of possibly informative paths without subjects or objects. A similar result is reported in Henderson et al. (2002), who find that using the obliqueness hierarchy to isolate important index terms in an information retrieval task degrades performance. The use of the less fine-grained *length* path value function delivers better results for the `medium` and `maximum` context configurations (see Table 3). Finally, we observe that the `medium` context yields the best overall performance. Within the currently explored parameter space, `medium` appears to strike the best balance: it includes some dependency paths beyond length one (corresponding to informative indirect relations), but also avoids very long and infrequent contexts which could potentially lead to overly sparse representations. In sum, the best dependency-based model uses the `medium` content selection and *length* path value functions, 2,000 basis elements, and Lin's (1998a) similarity measure. This model will be used for our subsequent experiments without additional parameter tuning. We will refer to this model as the *optimal dependency-based model*.

### 4.3 Baseline Model

Our experiments will compare the optimal dependency model just described against a state-of-the art word-based vector space model commonly used in the literature. The latter employs a "bag of words" definition of context (see (5)), uses words as basis elements and assumes that all words are given equal weight. In order to allow a fair comparison, we trained the word-based model on the same corpus as the dependency-based model (the complete BNC) and selected parameters that have been considered "optimal" in the literature (Patel, Bullinaria, and Levy 1998; McDonald 2000; Lowe and McDonald 2000). Specifically, we built a word-based model with a symmetric

10 word window as context and the most frequent 500 content words from the BNC as dimensions.[4] We used log-likelihood as our lexical association function and the cosine similarity measure[5] as distance measure.

## 5. Experiment 1: Single-word Priming

A large number of modeling studies in psycholinguistics have focused on simulating semantic priming phenomena (Lowe and McDonald 2000; McDonald 2000; Lund and Burgess 1996; McDonald and Brew 2004). The semantic priming paradigm provides a natural test bed for semantic space models, as it concentrates on the semantic similarity or dissimilarity between words, and it is precisely this type of lexical relations that vector-based models should capture. If dependency-based models indeed represent more linguistic knowledge, they should be able to model semantic priming better than traditional word-based models.

In this experiment, we focus on Hodgson's (1991) single-word lexical priming study. In single-word semantic priming, the transient presentation of a *prime word* like *tiger* directly facilitates pronunciation or lexical decision on a *target word* like *lion*: responses are usually faster and more accurate when the prime is semantically related to the target than when it is unrelated. Hodgson (1991) set out to investigate which types of lexical relations induce priming. He collected a set of 144 word pairs exemplifying six different lexical relations: (a) synonymy (words with the same meaning, e.g., *value* and *worth*), (b) superordination and subordination (one word is an instance of the kind expressed by the other word, e.g., *pain* and *sensation*), (c) category coordination (words which express two instances of a common superordinate concept, e.g., *truck* and *train*), (d) antonymy (words with opposite meaning, e.g., *friend* and *enemy*), (e) conceptual association (the first word subjects produce in free association given the other word, e.g., *leash* and *dog*), and (f) phrasal association (words which co-occur in phrases, e.g., *private* and *property*). The pairs covered the most prevalent parts of speech (adjectives, verbs, and nouns), they were selected to be unambiguous examples of the relation type they instantiate and were matched for frequency. Hogdson found equivalent priming effects (i.e., reduced reading times) for all six types of lexical relation, indicating that priming was not restricted to particular types of prime-target relation.

The priming effects reported in Hodgson (1991) have recently been modeled by McDonald and Brew (2004) using an incremental vector-based model of contextual facilitation. Their ICE model (short for Incremental Construction of Semantic Expectations) simulates the difference in effort between processing a target word preceded by a related prime and processing the same target preceded by an unrelated prime. This is achieved by quantifying the ability of the distributional characteristics of the prime word to predict the distributional properties of the target. The prime word is represented by a vector of probabilities which reflects the likely location in semantic space of the upcoming word. When the target word is observed, the representation is updated using a Bayesian inference mechanism to reflect the newly arrived information. McDonald and Brew (2004) use a traditional semantic space that takes only word co-occurrences into account and is defined over the 500 most frequent words of the spoken portion of

4 Increasing the dimensions of the space to 1,000 and 2,000 degraded performance. Smaller context windows did not yield performance gains either.

5 We repeated all experiments for the word-based model with Lin's (1998a) distance measure, obtaining consistently worse results.

the BNC. They measure distance in semantic space using relative entropy (also known as Kullback-Leibler divergence) and successfully model the data by predicting that its value should be lower for related prime-target pairs than for unrelated prime-target pairs.

### 5.1 Method

In this experiment we follow McDonald and Brew's (2004) methodology in simulating semantic priming. However, since our primary focus is on the representation of the semantic space, we do not adopt their incremental model of semantic processing. We simply model reading time for prime-target pairs by distance in the semantic space, without making explicit predictions about upcoming words.

From the 143 prime-target pairs listed in Hodgson (1991) (one synonymy pair is missing in the original dataset), seven pairs containing at least one low-frequency word (less than 100 occurrences in the BNC) were removed to avoid creating vectors with unreliable counts.[6] We constructed a dependency-based model with the parameters that yielded best performance on our development set (see Section 4.2) and a baseline word-based model (see Section 4.3). Each prime-target pair was represented by two vectors (one corresponding to the prime and one corresponding to the target).

These prime-target pairs form the items in this experiment. The independent variables (i.e., the variables directly manipulated by Hodgson (1991) in his original experiment) are (1) the type of Lexical Relation (antonyms, synonyms, conceptual associates, phrasal associates, category coordinates, superordinate-subordinates), and (2) the Prime (related, unrelated). The dependent variable (i.e., the quantity being measured) is the distance between the vector space representations of the prime and the target. The priming effect is simulated by comparing the distances between Related and Unrelated prime-target pairs. Since the original materials do not provide Unrelated primes, we emulated the unrelated pairs as described in McDonald and Brew (2004), by using the average distance of a target to all other primes of the same relation.

We test two hypotheses: first, that our dependency-based model can simulate semantic priming. Failure to do so would indicate that our model is deficient since it cannot capture basic semantic relatedness, a notion underlying many tasks in cognitive science and NLP. Second, we predict that the dependency-based model will be better at simulating priming than a traditional word-based one.

### 5.2 Results

We carried out a two-way Analysis of Variance (ANOVA) on the simulated priming data generated by the optimal dependency-based and the baseline word-based model. The factors were the two independent variables introduced above, namely Lexical Relation (six levels) and Prime (two levels). A reliable Prime effect was observed for the dependency-based model ($F(1,129) = 182.46$, $MSE = 0.93$, $p < 0.01$): the distance between a target and its Related prime was significantly smaller than between a target and an Unrelated prime. We also observed a reliable Prime effect for the traditional word-based model that did not use any syntactic information ($F(1,129) = 106.69$,

6 Low frequency words are deemed to produce *high variance* vectors because the co-occurrence counts needed to determine $M[t][b]$ will be unreliable (see McDonald (2000) for further evidence). Variance can be decreased by providing more data or by smoothing; however, we leave this to future work.

**Table 4**
Mean distance values for Related and Unrelated prime-target pairs; Prime Effect size (= Related − Unrelated) for the dependency model and ICE.

| Lexical Relation | $N$ | Related | Unrelated | Effect (dependency) | Effect (ICE) |
|---|---|---|---|---|---|
| Synonymy | 23 | 0.267 | 0.102 | 0.165** | 0.063 |
| Superordination | 21 | 0.227 | 0.121 | 0.106** | 0.067 |
| Category Coordination | 23 | 0.256 | 0.119 | 0.137** | 0.074 |
| Antonymy | 24 | 0.292 | 0.127 | 0.165** | 0.097 |
| Conceptual Association | 23 | 0.204 | 0.121 | 0.083** | 0.086 |
| Phrasal Association | 22 | 0.146 | 0.103 | 0.043** | 0.058 |
| **$p < 0.01$ (2-tailed) | | | | | |

$MSE = 2.92$, $p < 0.01$). There was no main effect of Lexical Relation for either model ($F(5, 129) < 1$).

The fact that the analysis of variance has produced a significant $F$ for the two models only indicates that there are differences between the Related and Unrelated prime-target means that cannot be attributed to error. Ideally, we would like to compare the two models, for example, by quantifying the magnitude of the Prime effect. Eta-squared ($\eta^2$) is a statistic[7] often used to measure the strength of an experimental effect (Howell 2002). It is analogous to $r^2$ in correlation analysis and represents how much of the overall variability in the dependent variable (in our case distance in semantic space) can be explained or accounted for by the independent variable (i.e., Prime). The use of $\eta^2$ allowed us to perform comparisons between models (the higher the $\eta^2$, the better the model). The Prime effect size was greater for the dependency model which obtained an $\eta^2$ of 0.332 compared to the word-based model whose $\eta^2$ was 0.284. In other words, the dependency model accounted for 33.2% of the variance, whereas the word-based model accounted for 28.4%.

To establish whether the priming effect observed by the dependency model holds across all relations, we next conducted separate ANOVAS for each type of Lexical Relation. The ANOVAS revealed reliable priming effects for all six relations. Table 4 shows the mean distance values for each relation in the Related and Unrelated condition and the Prime Effect size for the dependency model. The latter was estimated as the difference in distance values between related and unrelated prime-target pairs (asterisks indicate whether the difference is statistically significant, according to a two-tailed paired t-test). For comparison, we also report the Prime Effect size that McDonald and Brew (2004) obtained in their simulation.

To summarize, our results indicate that a semantic space model defined over dependency relations simulates direct priming across a wide range of lexical relations. Furthermore, our model obtained a priming effect that is not only reliable but also greater in magnitude than the one obtained by a traditional word-based model. Although we used a less sophisticated model than McDonald and Brew (2004), without an update procedure and an explicit computation of expectations, we obtained priming effects across all relations. In fact, we consider the two models complementary. McDonald and

7 Eta-squared is defined as $\eta^2 = \frac{SS_{effect}}{SS_{total}}$ where $SS_{effect}$ is the variance (sum of squares) created by one particular effect (Prime in our case) and $SS_{total}$ is the variance of all observations together.

Brew's model could straightforwardly incorporate syntax-based semantic spaces like the ones defined in this article.

We next examine synonymy, a single lexical relation, in more detail and assess whether the proposed dependency model can reliably distinguish synonyms from non-synonyms. This capability may be exploited to automatically generate corpus-based thesauri (Grefenstette 1994; Lin 1998a; Curran and Moens 2002) or used in applications that utilize semantic similarity. Examples include contextual spelling correction (Jones and Martin 1997), summarization (Erkan and Radev 2004; Barzilay 2003) and question answering (Lin and Pantel 2001).

## 6. Experiment 2: Detecting Synonymy

The *Test of English as a Foreign Language* (TOEFL) is commonly used as a benchmark for comparing the merits of different similarity models . The test is designed to assess non-native speakers' knowledge of English. It consists of multiple-choice questions, each involving a target word embedded in a sentence and four potential synonyms. The task is to identify the real synonym. An example is shown below where *crossroads* is the real synonym for *intersection*.

You will find the office at the main **intersection**.
(a) place (b) crossroads (c) roundabout (d) building

Landauer and Dumais (1997) were the first to propose the TOEFL items as a test for lexical semantic similarity. Their LSA model achieved an accuracy of 64.4% on 80 items, a performance comparable to the average score attained by non-native speakers taking the test. Sahlgren (2006) uses Random Indexing, a method comparable to LSA, to represent the meaning of words and reports a 75.0% accuracy on the same TOEFL items. It should be noted that both Landauer and Dumais (1997) and Sahlgren (2006) report results on seen data, i.e., parameters are optimized on the entire dataset until performance has peaked.

Rather than assuming that similar words tend to occur in similar contexts, Turney (2001) and Higgins (2004) propose models that capitalize on the collocational nature of semantically related words. Two words are considered similar if they tend to occur *near each other*. Turney (2001) uses pointwise mutual information (PMI) to measure the similarity between a target word and each of its candidate synonyms. Co-occurrence frequencies are retrieved from the web using an information retrieval (IR) engine:

$$Similarity_{\text{PMI-IR}}(w_1, w_2) = \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \approx \frac{\text{hits}(w_1 \text{ NEAR } w_2)}{\text{hits}(w_1)\text{hits}(w_2)} \tag{15}$$

where $P(w_1, w2)$ is estimated by the number of hits (i.e., number of documents) returned by the IR engine (Turney (2001) used Altavista) when submitting a query with the NEAR operator[8]. The PMI-IR model obtained an accuracy of 72.5% on the TOEFL dataset.

8 The NEAR operator constrains the search to documents that contain $w_1$ and $w_2$ within ten words of one another, in either order.

Higgins (2004) proposes a modification to (15): he dispenses with the NEAR operator by concentrating on word pairs that are strictly adjacent:

$$Similarity_{\text{LC-IR}}(w_1, w_2) = \frac{\min(\text{hits}(w_1, w_2), \text{hits}(w_2, w_1))}{\text{hits}(w_1)\text{hits}(w_2)} \quad (16)$$

Note that (16) takes the minimum number of hits for the two possible orders $w1, w_2$ and $w_2, w_1$ in an attempt to rule out the effects of collocations and part-of-speech ambiguities. The LC-IR (local-context information retrieval) model outperformed PMI-IR, achieving an accuracy of 81.3% on the TOEFL items.

### 6.1 Method

For this experiment, we used the TOEFL benchmark dataset[9] (80 items). We compared our optimal dependency-based model against the baseline word-based model. We would also like to compare the vector-based models against Turney's (2001) and Higgins' (2004) collocational models. Ideally, such a comparison should take place on the same corpus. Unfortunately, downloading and parsing a snapshot of the whole web is outside the scope of the present article. Instead, we assessed the performance of these models on the BNC, using a search engine which simulated Altavista. Specifically, we indexed the BNC using Glimpse (Manber and Wu 1994), a fast and flexible indexing and query system[10]. Glimpse supports approximate and exact matching, Boolean queries, wild cards, regular expressions, and many other options.

For the PMI-IR model, we estimated hits($w_1$ NEAR $w_2$) by retrieving and counting the number of documents containing $w_1$ and $w_2$ or $w_2$ and $w_1$ in the same sentence. The target $w_1$ and its candidate synonym $w_2$ did not have to be adjacent, but the number of the intervening words was bounded by the length of the sentence. The frequencies hits($w_1$) and hits($w_2$) were estimated similarly by counting the number of documents in which $w_1$ and $w_2$ occurred. Ties were resolved by randomly selecting one of the candidate synonyms. The BNC proved too small a corpus for the LC-IR model which relies on $w_1$ and $w_2$ occurring in directly adjacent positions. This is not a problem when frequencies are obtained from web-scale corpora, but in our case most queries retrieved no documents at all (96.6% of hits($w_1, w_2$) and 95% of hits($w_2, w_1$) were zero). We thus report only the performance of the PMI-IR model on the BNC.

The models performed a decision task similar to TOEFL test takers: they had to decide which one of the four alternatives was synonymous with the target word. For the vector-based models, we computed the distance between the vector representing the candidate word and each of the candidate synonyms, and selected the candidate with the smallest distance. Analogously, the candidate with the largest PMI-IR value was chosen for Turney's (2001) model. Accuracy was measured as the percentage of right decisions the model made. We also report the accuracy of a naive baseline model which guesses synonyms at random.

In this experiment, we aim to show that the superior performance of the dependency model carries over to a different task and dataset. We are further interested to see whether linguistic information (represented in our case by dependency paths) makes up for the vast amounts of data required by the collocational models. We therefore

9 The items were kindly provided to us by Thomas Landauer.
10 The software can be downloaded from `http://webglimpse.net/download.php`.

**Table 5**
Comparison of different models on the TOEFL synonymy task ($^{\dagger}$: significantly better than random guessing, $^{*}$: significantly better than word-based vector model)

| Model | Corpus | Accuracy (%) |
|---|---|---|
| Random Baseline | — | 25.0 |
| Word-based Space | BNC | $61.3^{\dagger}$ |
| Dependency Space | BNC | $73.0^{\dagger *}$ |
| PMI-IR | BNC | $61.3^{\dagger}$ |
| PMI-IR | Web | $72.5^{\dagger *}$ |
| LC-IR | Web | $81.3^{\dagger *}$ |

compare directly previously proposed web-based similarity models with BNC-based vector space models.

### 6.2 Results

Our results[11] are summarized in Table 5. We used a $\chi^2$ test to determine whether the differences in accuracy are statistically significant. Not surprisingly, all models are significantly better than random guessing ($p < 0.01$). The dependency model significantly outperforms the word-based model and PMI-IR when the latter uses BNC frequencies ($p < 0.05$). PMI-IR performs comparably to our model when using web frequencies. The web-based LC-IR numerically outperforms the dependency model, however the difference is not statistically significant on the TOEFL dataset ($p < 1$). Expectedly, web-based PMI-IR and LC-IR are significantly better than the word-based vector model and the BNC-based PMI-IR ($p < 0.05$).

Our results show that the dependency-based model retains its advantage over the word-based model on the synonymy detection task. On the BNC, it also outperforms the collocation-based PMI-IR. Our interpretation is that the conceptually simpler collocation models suffer from data sparseness, while the dependency model can profit from the additional distributional information it incorporates. It is a matter of future work to examine whether dependency models can carry over their advantage to larger corpora.

Our following experiment applies the dependency space introduced in this article to word sense disambiguation (WSD), a task which has received much attention in NLP and is ultimately important for document understanding.

## 7. Experiment 3: Sense ranking

The ability to identify the intended reading of a polysemous word (the *word sense*) in context is crucial for accomplishing many NLP tasks. Examples include lexicon acquisition, discourse parsing, or metonymy resolution. Applications such as question answering or machine translation could also benefit from large scale word sense disambiguation (WSD).

Given the importance of WSD for basic NLP tasks and multilingual applications, a variety of approaches have been proposed for disambiguating word senses. To date,

11 We omit LSA (Landauer and Dumais 1997) and Random indexing (Sahlgren 2006) from our comparison, since these models were not evaluated on unseen data.

most accurate WSD systems are supervised and rely on the availability of training data (see Yarowsky and Florian (2002), Mihalcea and Edmonds (2004) and the references therein). Although supervised methods typically achieve better performance than their unsupervised alternatives, their applicability is limited to those words for which sense labeled data exists, and their accuracy is strongly correlated with the amount of labeled data available. Furthermore, if the distribution of senses is skewed, as is often the case, the simple heuristic of choosing the most common or predominant sense in the training data (henceforth "the first sense heuristic") delivers results competitive with supervised approaches based on local context (Hoste et al. 2002).

Obtaining the first sense heuristic via annotation is obviously costly and time consuming. More importantly, one would expect that a word's first sense varies across domains and text genres (the word *court* in legal documents will most likely mean "tribunal" rather than "yard"). Therefore, manual annotation must be redone for most new languages, domains, and sense inventories. McCarthy et al. (2004) show that the annotation bottleneck can be avoided by inferring the first sense heuristic automatically from raw text. They argue that, even though the first sense heuristic is not a WSD method in itself, it can be usefully combined with context-based disambiguation methods in order to alleviate the data requirements for WSD. Their method builds on the observation that a word's distributionally similar neighbors often provide cues about its senses. In their model, sense ranking is equivalent to quantifying the degree of similarity between each neighbor and each sense description of a polysemous word. The sense most similar to the neighbors is the first sense.

McCarthy et al.'s (2004) approach crucially relies on the quality of the set of neighbors to acquire more or less accurate first senses. In this experiment, we examine whether the dependency-based models discussed in this article can be used for the sense ranking task, thereby assessing their potential for practical NLP tasks. The aims of our experiment are twofold: (1) to investigate whether our dependency-based framework can be used to acquire distributionally similar words that differ in quality from those obtained with word-based models and (2) to observe their impact on WSD. We first describe McCarthy et al.'s (2004) sense ranking model, which forms the basis of our experiments, and then detail our methodology and results.

### 7.1 The sense ranking model

Let $w$ be a word, $N(w) = \{n_1, n_2, \ldots, n_k\}$ the set of the $k$ most similar words to $w$, and $S(w) = \{ws_1, ws_2, \ldots ws_n\}$ the set of senses for $w$. McCarthy et al.'s (2004) model assigns each sense $ws_i$ a "predominant sense score" $PS(ws_i)$ as follows:

$$PS(ws_i) = \sum_{n_j \in N(w)} sim_{distr}(w, n_j) \times \frac{sim_{sem}(ws_i, n_j)}{\sum_{ws_{i'} \in S(w)} sim_{sem}(ws_{i'}, n_j)} \tag{17}$$

where

$$sim_{sem}(ws_i, n_j) = \max_{ws_x \in S(n_j)} sim_{WN}(ws_i, ws_x) \tag{18}$$

The predominant sense of $w$ is simply the one with the largest $PS(ws_i)$, i.e., the sense that is maximally similar to its neighbors $n_j \in N(w)$ according to (17) and (18).

This sense ranking model has four free parameters: (1) the semantic space over which distributionally similar words are acquired, (2) the measure of distributional similarity ($sim_{distr}$), (3) the number of neighbors taken into account ($k$), and (4) the measure of sense similarity ($sim_{WN}$). The $PS$ score combines distributional similarity and sense similarity, taking into account both lexical knowledge gathered from corpora and the organization and structure of the lexical resource that provides the sense inventory. A large number of sense similarity measures have been developed for WordNet and WordNet-like taxonomies. These vary from simple edge-counting (Rada, Mili, and Bicknell 1989) to attempts to factor in peculiarities of the network structure by considering link direction (Hirst and St-Onge 1998), relative depth (Leacock and Chodorow 1998), and density (Agirre and Rigau 1996). A number of hybrid approaches have also been proposed that combine WordNet with corpus statistics (Resnik 1995; Jiang and Conrath 1997).

McCarthy et al. (2004) use their ranking model to automatically infer the first senses of all nouns attested in SemCor, a subset of the Brown corpus containing 23,346 lemmas annotated with senses according to WordNet 1.6. They acquire distributionally similar words from a large collection of dependency relations obtained from the written part of the BNC (90 million words) using Briscoe and Carroll's (2002) parser. Their model considers solely dependency paths of length one (see context selection function (5)), and is restricted to a small set of dependency relations (verb-subject, verb-object, noun-noun, and adjective-noun). They employ a basis mapping function that maps paths to $(r, w)$ tuples (see (9)) and Lin's information-theoretic similarity measure (see (3)). They obtained a type-level accuracy of 54% (a random baseline achieved 32%) at recovering the most prevalent sense (using 50 neighbors and either Lesk's (1986) or Jiang and Conrath's (1997) measures). They also used a token disambiguator that always defaults to the automatically acquired first sense and obtained a token-level disambiguation accuracy of 48% for Lesk (50 neighbors) and 46% for Jiang and Conrath (50 neighbors). Their baseline for this task was 24%.

### 7.2 Method

We replicated McCarthy et al.'s (2004) study using our optimal dependency-based model (`medium` context selection, *length* path value functions, 2,000 basis elements, Lin's (1998a) similarity measure, and the log-likelihood association function) and the baseline word-based model. We used equation (17) to find the first sense for all polysemous nouns in SemCor (according to WordNet 1.6). Following McCarthy et al., we only considered polysemous nouns attested in SemCor with a frequency $> 2$, and in our parsed version of the BNC with a frequency $\geq 10$. The total number of nouns after applying the frequency cutoffs was 2,750[12] and the average sense ambiguity was 4.55 (the most ambiguous word had 30 senses, and least ambiguous 2). For each one of the 2,750 nouns, we generated the set of its distributionally similar neighbors from the set of the nouns in the intersection between the BNC and WordNet (15,656 in total).

We did not experiment in detail with WordNet-based similarity measures or with the number of distributionally similar neighbors required for the computation of the prevalence score. McCarthy et al. (2004) undertook a thorough comparison and ob-

12 McCarthy et al. (2004) use 2,595 nouns. The slight variation is due to the different parsers employed in the two studies. Recall that we obtain dependency relations using MINIPAR (Lin 1998b), whereas McCarthy et al. employ Briscoe and Carroll's (2002) parser.

tained best results with 50 neighbors using Lesk's (1986) and Jiang and Conrath's (1997) measures. They argue that the latter measure is more efficient for large scale WSD and use it exclusively in all subsequent work (McCarthy et al. 2004; Koeling, McCarthy, and Carroll 2005). We thus adopted the parameters that McCarthy et al. found to be optimal, namely 50 neighbors and Jiang and Conrath's similarity measure, which we briefly describe below.

Jiang and Conrath's (1997) measure estimates the similarity between two word senses by combining taxonomic information with corpus data. It is based on the notion of information content (IC) of a WordNet synset $s$. IC is defined as the negative log-likelihood of $s$, the probability of encountering $s$ in a given corpus:

$$IC(s) = -\log p(s) \tag{19}$$

Jiang and Conrath (1997) define a distance measure that combines IC with edge counting by taking into account local density, node depth and link type. They introduce two parameters, $\alpha$ and $\beta$, that control the influence of node depth and density respectively. Setting $\alpha$ to zero and $\beta$ to one, their measure simplifies to:

$$D_{jcn}(s_1, s_2) = \log p(s_1) + \log p(s_2) - 2 \times \log p(\text{lso}(s_1, s_2)) \tag{20}$$

where $\text{lso}(s_1, s_2)$ is the lowest super-ordinate (most specific common subsumer) of synsets (that is, senses) $s_1$ and $s_2$. We used the WordNet Similarity Package (Pedersen, Patwardhan, and Michelizzi 2004) which provides an implementation of Jiang and Conrath's (1997) measure (version 0.06).[13] We re-estimated the IC counts from the BNC, since those provided with the package are derived from the manually annotated SemCor and would positively bias our results.

We replicated McCarthy et al.'s (2004) procedure for evaluating the acquired predominant sense against the manually annotated SemCor. We use the following notation to describe our evaluation measures: $W$ is the set of all word types ($|W| = 2,570$) and $W_{ps}$ is the set of word types with a predominant sense, i.e., with a sense that is more frequent than the second sense in SemCor ($|W_{ps}| = 2,338$). $S(w)$ is the set of WordNet senses for word type $w$, and $T(w)$ the set of all tokens of $w$. Finally, we use $ps_{sc}(w)$ and $ps_r(w)$ to refer to the predominant sense of word $w$ according to SemCor and the sense ranking model, respectively, and $sense_{sc}(t)$ to denote the sense annotated in SemCor for a particular token $t$.

We first evaluate our models performance on the *sense ranking* task ($Acc_{sr}$), i.e., on identifying the predominant sense for a word type, if one exists:

$$Acc_{sr} = \frac{|\{w \in W_{ps} \mid ps_{sc}(w) = ps_r(w)\}|}{|W_{ps}|} \tag{21}$$

A baseline for the sense ranking task can be easily defined by selecting a sense at random for each word type from its sense inventory and assuming that this is the first sense:

$$Random_{sr} = \frac{1}{|W_{ps}|} \sum_{w \in W_{ps}} \frac{1}{|S(w)|} \tag{22}$$

13 The package is publicly available from `http://www.d.umn.edu/~tpederse/similarity.html`.

**Table 6**
Results on sense ranking and WSD tasks, using 50 neighbors and the Jiang and Conrath (1995) distance measure ($^{\dagger}$: significantly better than random baseline, $^{*}$: significantly better than word-based model, $^{\$}$: significantly better than McCarthy et al.)

| Models | $Acc_{sr}$ | $Acc_{wsd}$ |
|---|---|---|
| Random Baseline | 31.0 | 25.4 |
| Word-based Space | $49.3^{\dagger}$ | $49.9^{\dagger\$}$ |
| Dependency Space | $54.3^{\dagger *}$ | $54.3^{\dagger *\$}$ |
| McCarthy et al. | $54.0^{\dagger *}$ | $46.0^{\dagger}$ |
| Upper Bound | — | 67.0 |

Like McCarthy et al. (2004), we also assessed the *word sense disambiguation* potential ($Acc_{wsd}$) of the automatically acquired first senses for each word token. We assigned the predominant sense (according to the ranking model) to every noun token, without taking its context into account, and measured the ratio of tokens for which the first sense given by the ranking model is identical to the SemCor gold standard sense:

$$Acc_{wsd} = \frac{\sum_{w \in W} |\{t \in T(w) \mid ps_r(w) = sense_{sc}(t)\}|}{\sum_{w \in W} |T(w)|} \tag{23}$$

A baseline disambiguator can be defined by assigning a random sense to each token:

$$Random_{wsd} = \frac{1}{\sum_{w \in W} |T(w)|} \sum_{w \in W} |T(w)| \frac{1}{|S(w)|} \tag{24}$$

### 7.3 Results

Table 6 shows the results for the optimal dependency-based model, the random baseline, the baseline word-based model, and McCarthy et al.'s (2004) state of the art model. As an upper bound, we report WSD accuracy when defaulting to the first (i.e., most frequent) sense provided by SemCor. All models use 50 nearest neighbors and Jiang and Conrath's (1997) WordNet-based semantic similarity measure. As far as distributional similarity is concerned, our dependency model employs Lin's (1998a) measure and so do McCarthy et al., whereas the traditional word co-occurrence model uses cosine. Our model differs from McCarthy et al. in the context selection, path value and basis mapping functions (see the discussion below). We used a $\chi^2$ test to determine if the differences in performance are statistically significant. Note that we have a slightly different set of nouns from McCarthy et al. (2004); this is due to the use of a different parser and a larger corpus. We work on the assumption that this difference is negligible. We use a set of diacritics to denote statistical significance, explanations for which are provided in Table 6.

We first consider the predominant sense acquisition task ($Acc_{sr}$). Table 6 shows that all models significantly outperform the random baseline ($p < 0.01$). Furthermore, both the dependency-based model and McCarthy et al. (2004) significantly outperform the word-based model. The two dependency models yield comparable performances ($p < 1$). For the WSD task, we also observe that all models significantly outperform

**Table 7**
Sense ranking and WSD accuracy for the dependency-based model as word frequency and average sense ambiguity are varied (FBand: frequency band, AvgAmbig: average WordNet sense ambiguity within frequency band, Types: number of noun types within frequency band)

| FBand | AvgAmbig | Types | $acc_{sr}$ | $acc_{wsd}$ |
|---|---|---|---|---|
| <50 | 3.29 | 174 | 0.53 | 0.46 |
| 50-200 | 3.60 | 489 | 0.54 | 0.49 |
| 200-1,000 | 4.29 | 1,014 | 0.57 | 0.54 |
| **1,000-5,000** | **5.65** | **583** | **0.51** | **0.57** |
| **5,000+** | **8.32** | **78** | **0.50** | **0.51** |

the random baseline ($p < 0.01$). Our dependency model significantly outperforms the word-based model and McCarthy et al. ($p < 0.01$). The word-based model performs significantly better than McCarthy et al. ($p < 0.01$). All models expectedly perform worse than the upper bound ($p < 0.01$).

An interesting observation is that our dependency model outperforms McCarthy et al. (2004) by a large margin (8.3%) on the WSD task, while the two models yield comparable performances on sense ranking. Also, the word-based model performs significantly better than McCarthy et al. on WSD, while it is significantly worse than McCarthy et al. in sense ranking. This indicates that the words for which each model delivers the first sense correctly are different. Indeed, inspection of the first sense assignments reveals that McCarthy et al. and our dependency model have only 35.7% nouns in common for which they predict the first sense correctly. McCarthy et al. has 34.8% nouns in common with the word-based model which in turn has 40.3% nouns in common with our dependency model.

To follow up on this observation, we investigated how ambiguity and word frequency influence the performance of our ranking model. In theory, an automatically acquired sense ranker should have a good accuracy on *all ambiguous* words in order to do well on WSD. However, in practice the sense ranker's performance depends crucially on its ability to correctly predict the first sense for *highly frequent* and *highly ambiguous* words. An additional complicating factor is the sense distribution of the words in question. For words whose sense distributions are not particularly skewed, getting the first sense wrong will not be entirely detrimental as long as the WSD method misclassifies as predominant relatively frequent senses.

Take, for example, the word *corner* which is attested 61 times in Semcor and has 11 senses according to WordNet 1.6. Among these, sense 1 is found seventeen times, sense 2 fifteen, sense 3 ten0, and sense 4 nine (all other senses have considerably smaller frequencies). Now suppose that the sense ranking method wrongly identifies sense 2 as the predominant sense for *corner*. Using this sense, our WSD system will correctly disambiguate 24.6% of the instances of *corner* in Semcor, despite the fact that it will not receive any credit for identifying the first sense. Note that the right first sense would yield only slightly better accuracy (i.e., 27.4%).

We grouped all ambiguous noun tokens in SemCor into five frequency bands (frequencies were estimated from the BNC as it constitutes a larger sample of English than Semcor). Table 7 illustrates our models' sense ranking and WSD accuracy according to these bands; we also list the average sense ambiguity and number of word types for each band. As can be seen, our dependency model obtains consistently good performance on both tasks, even in the high ambiguity bands (Bands 1,000–5,000 and 5,000+, highlighted

in Table 7). The obtained accuracies are well above the baseline of choosing a sense at random (for example, an average ambiguity of 8.3 in the 5000+ band corresponds to a random baseline of 12% in the sense ranking task). This is not entirely surprising; frequent words are represented by more reliable vectors. As a result, the acquired neighbors are of higher quality, which counteracts the increased ambiguity.

The results in Table 7 furthermore reveal that WSD performance exceeds sense ranking accuracy in high-frequency bands (most notably in Band 1,000-5,000), which seems counterintuitive. This effect can be explained by taking into account the observed sense frequencies and the types of errors introduced by our model in these bands. The distribution of senses in the high-frequency bands tends to be less skewed, at least according to Semcor (82% of nouns in Band 1,000-5,000 and 65% in Band 5,000+ have a first sense with frequency $<$50). Our model's mistakes are often "near misses", i.e., the first and second sense ranks are flipped. Specifically, near misses are observed for 25% of the noun types in Band 1,000-5,000, and 15% in Band 5,000+. Now, for nouns with non-skewed sense distributions, disambiguating with the second sense will boost WSD accuracy even though this is not the case for sense ranking (see the discussion above).

Our results show that semantic space models defined according to the framework presented in this article can be successfully used for the automatic acquisition of first senses from raw text. We obtained results similar to McCarthy et al. (2004) on the sense ranking task and demonstrated that our model performs significantly better on WSD. Furthermore, it outperformed a word-based semantic space on both tasks. Our model differs from McCarthy et al. in three important ways: (a) following our terminology, they use a semantic space with the `minimum` context selection (paths of length one) and *plain* path value (no path weighting) functions, whereas our model employs the `medium` content selection and *length* path value functions; (b) their space is constructed over a limited set of dependency paths, namely subject, object and adjective/noun modification relations, whereas our model uses a wider range of relations including information about tense (for example, whether a complement is finite or not), relativisation, etc. (see Section 4.2 for details); and (c) their basis mapping function maps paths to tuples whereas we employ a word-based function and restrict the dimensions of the space to the 2,000 most frequent elements (McCarthy et al. do not employ any cutoffs). Furthermore, they used a slightly smaller corpus (only the written part of the BNC, amounting to 90% of the total corpus) and a different parser (Briscoe and Carroll 2002).

Although replicating our study with Briscoe and Carroll's parser (2002) is outside of the scope of this article, we should note that the two parsers yield comparable performances and employ a similar inventory of dependency relations (see Curran (2004) for more discussion). We thus suspect that differences in performance cannot be uniquely attributed to parser performance. We can, however, assess whether the difference is due to corpus size by examining its effect on the performance of our model. If it is indeed sensitive to corpus size, we would expect a relatively large drop in performance when our semantic space is built on smaller corpora. We randomized the order of sentences in the BNC and constructed semantic spaces on data sets progressively increasing in size: the first space was constructed from 5% of the BNC, the next from 10% and so on. We tested each model on the SemCor data (see Section 7.2). Figure 7 shows the resulting learning curves. When the dependency model is constructed on 5% of the BNC, it delivers a WSD accuracy of 51% which eventually increases to 54.3% when the entire corpus is used. This result indicates that the model performs well when trained on a small corpus and that its good performance cannot be attributed solely to corpus size. However, it also suggests that a large increase in corpus size is necessary to obtain substantial improvements with the present sense ranking strategy, which

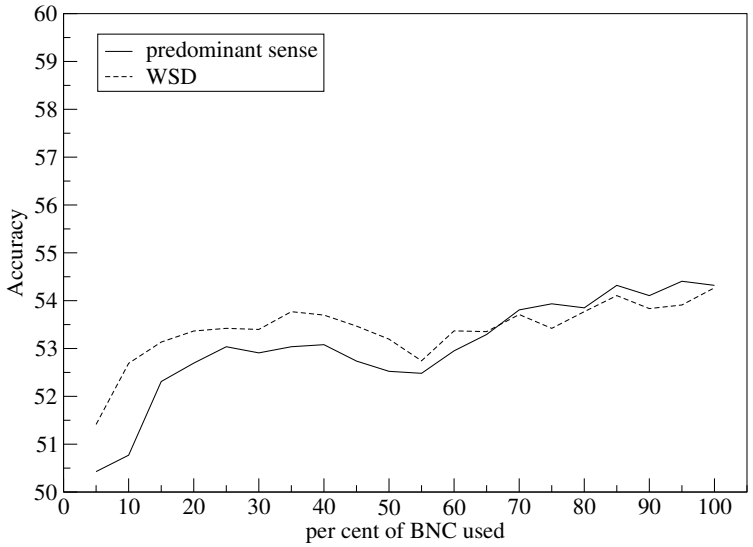



**Figure 7**
Learning curve for the dependency-based model on a randomized version for the BNC: accuracy of predominant sense acquisition (solid) and WSD (dashed) with varying corpus size

uses distributional similarity as a corrective for taxonomy-based similarity: accuracy increases by approximately 4% when our corpus size increases by a factor of 20.

We believe that the differences in performance between the two models are largely due to differences in the basis mapping function. Since McCarthy et al. (2004) use all available basis elements, their semantic space grows linearly with vocabulary (i.e., corpus) size. Each target word is represented by a set of "features" – relation-word pairs with a non-zero occurrence frequency – which may vary widely between target words. In contrast, our model defines a modest number of basis elements (2,000) which are shared between all target words. The resulting representation is a vector space which is less sparse and the resulting neighbors capture more succinctly the semantic properties of words. Additional evidence comes from the performance of the word-based model, which also uses a word basis mapping function and a fixed number of dimensions (500 words). Although this model does not incorporate syntactic information in any way, it manages to outperform McCarthy et al. on the WSD task. In sum, we attribute the superior performance of the vector-based model to two key factors: low dimensionality (as seen by the comparison to McCarthy et al.) and the incorporation of linguistic knowledge (as seen by the comparison to the word-based model).

## 8. General Discussion

In this article, we presented a general framework for the construction of semantic space models. The framework operates on paths of dependency relations, allowing linguistic knowledge to guide the construction of semantic spaces. It extends previous work on traditional word-based semantic space models as well as syntax-based models by providing a principled way for defining the context and the dimensions of the semantic space. More specifically, we isolated three important parameters of space construction: the context selection function, the basis mapping function and the path value function.

In combination, these three functions determine which paths (e.g., local or distant), dimensions (e.g., words, parts of speech or word-relation tuples), and dependency relations (e.g., subjects, objects) contribute towards the construction of a semantic space.

We evaluated our framework on tasks relevant for NLP and cognitive science and compared it against state of the art models. Experiment 1 revealed that semantic space models defined over dependency relations adequately simulate semantic priming. Experiments 2 and 3 examined the usefulness of our framework for NLP: we used our model to detect synonymy relations and to automatically acquire prevalent senses for polysemous words. In all cases, syntactically enriched models outperformed traditional word-based models that did not take account of syntax.

Our strategy in the present study was to define a small number of generic parameterizations, evaluate the resulting models on a development set, and select a broadly optimal model for testing on unseen data.Therefore, our models were not specifically tuned for the tasks at hand and we have only explored a relatively small subset of the parameter space. Our examination of different parameter combinations in Section 4.2 revealed that `medium` syntactic contents yield consistently better performance when combined with a path value function that penalizes longer paths (*length*). An important avenue for future work concerns defining more fine-grained path value functions. Our results show that a path value function inspired by the obliqueness hierarchy delivers worse results than the linguistically naive *length* function. Alternatively, we could define a function that combines *gram-rel* with *length*, or more generally learn a weighting scheme for paths by optimizing some objective function.

Our experiments concentrated on spaces that used solely a basis mapping function that maps dependency paths to words. It should also be interesting to experiment with different types of basis mapping functions. For example, we could experiment with more coarse-grained functions based on parts-of-speech or more fine-grained ones such as the relation-word pairs used by McCarthy et al. (2004). We would also like to observe the impact of singular value decomposition (SVD) on our semantic spaces along the lines of Kanejiya et al.'s (2003) cognitive modeling work. They use SVD to reduce the dimensionality of a semantic space that uses (word, part-of-speech) pairs as basis elements, obtaining better coverage compared with an LSA space constructed over word co-occurrences. Further studies must examine the effect of parser quality on the obtained co-occurrences, and the influence of the chosen similarity measure.

We have just scratched the surface of the possibilities for the framework discussed in this article. The potential applications are many and varied both for cognitive science and NLP. Our syntactically enriched models retain the simplicity of word co-occurrence models while allowing for the role of syntactic structure to influence the representation of the semantic space. The resulting vectors have a higher degree of linguistic plausibility – it is not mere lexical association that accounts for the meaning of words but rather their lexical and syntactic dependencies. Arguably, this property holds great promise for languages less configurational than English. A prediction that we intend to test in the future is that syntax-based semantic space models should be able to represent meaning more adequately than traditional word-based models for languages that allow constituent scrambling (e.g., German) or have free word order (e.g., Czech).

It remains to be seen whether our models can capture the wide range of data that traditional and LSA-based models have accounted for. Possible future experiments include mediated priming (Lowe and McDonald 2000) and multiple priming (McDonald and Brew 2004), intelligent tutoring (Kanejiya, Kumar, and Prasad 2003), and coherence rating (Foltz, Kintsch, and Landauer 1998). A number of NLP tasks could also benefit from the framework presented in this article. Examples include word sense

discrimination (Schütze 1998; Lin 1998a) automatic thesaurus construction (Grefenstette 1994; Curran and Moens 2002), automatic clustering, lexicon acquisition and in general similarity-based approaches to NLP.

**Acknowledgments**

We are grateful to Diana McCarthy for providing us with the results of her system on our data. We are also grateful to four anonymous reviewers for *Computational Linguistics* whose feedback helped to substantially improve the present article. We also thank Colin Bannard, Gemma Boleda, Amit Dubey, Katrin Erk, Frank Keller, Ulrike Padó, and Caroline Sporleder for useful comments and suggestions. A preliminary version of this work was published in the proceedings of ACL 2003; we thank the anonymous reviewers of that paper for their comments.

**References**

Agirre, Eneko and German Rigau. 1996. Word sense disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 16–22, Copenhagen, Denmark.

Banerjee, Satanjeev and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco, Mexico.

Bannard, Colin, Timothy Baldwin, and Alex Lascarides. 2003. A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 65–72, Sapporo, Japan.

Barzilay, Regina. 2003. *Information Fusion for Multi-Document Summarization: Praphrasing and Generation*. Ph.D. thesis, Columbia University, New York.

Berry, Michael W., Susan T. Dumais, and Gavin W. O'Brien. 1994. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595.

Briscoe, Ted and John Carroll. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 1499–1504, Las Palmas, Canary Islands.

Budanitsky, Alexander and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of ACL Workshop on WordNet and Other Lexical Resources*, pages 29–34, Pittsburgh, PA.

Burnard, Lou, 1995. *Users Guide for the British National Corpus*. British National Corpus Consortium, Oxford University Computing Service.

Choi, Freddy, Peter Wiemer-Hastings, and Johanna Moore. 2001. Latent Semantic Analysis for text segmentation. In *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing*, pages 109–117, Seattle, WA.

Curran, James R. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Curran, James R. and Marc Moens. 2002. Scaling context space. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 231–238, Philadelphia, PA.

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

Erkan, Günes and Dragomir R. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fillmore, Charles. 1965. *Indirect Object Constructions and the Ordering of Transformations*. Mouton, The Hague.

Fodor, Janet Dean. 1995. Comprehending sentence structure. In Lila R. Gleitman and Mark Liberman, editors, *Invitation to Cognitive Science*, volume 1. MIT Press, Cambridge, MA, pages 209–246.

Foltz, Peter W., Walter Kintsch, and Thomas K. Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Process*, 15:285–307.

Goldberg, Adele. 1995. *Constructions*. Chicago University Press, Chicago.

Golub, Gene H. and Charles F. Van Loan. 1989. *Matrix Computations*. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, 3rd edition.

Green, Georgia. 1974. *Semantics and Syntactic Regularity*. Indiana University Press, Bloomington.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer

Academic Publishers.
Gropen, Jess, Steven Pinker, Michelle Hollander, Richard Goldberg, and Ronald Wilson. 1989. The learnability and acquisition of the dative alternation. *Language*, 65(2):203–257.
Harris, Zellig. 1968. *Mathematical Structures of Language*. Wiley, New York.
Henderson, James, Paola Merlo, Ivan Petroff, and Gerold Schneider. 2002. Using syntactic analysis to increase efficiency in visualizing text collections. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 335–341, Taipei, Taiwan.
Higgins, Derrick. 2004. Which statistics reflect semantics? Rethinking synonymy and word similarity. In *Proceedings of the International Conference on Linguistic Evidence*, pages 265–284, Tübingen, Germany.
Hirst, Graeme and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA, pages 305–332.
Hodgson, James M. 1991. Informational constraints on pre-lexical priming. *Language and Cognitive Processes*, 6:169–205.
Hoste, Véronique, Iris Hendrickx, Walter Daelemans, and Antal van den Bosch. 2002. Parameter optimization for machine-learning of word sense disambiguation. *Language Engineering*, 8(4):311–325.
Howell, David C. 2002. *Statistical Methods for Psychology*. Duxbury, Pacific Grove, CA, 5th edition.
Jackendoff, Ray. 1983. *Semantic and Cognition*. The MIT Press, Cambridge, MA.
Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th International Conference on Research in Computational Linguistics*, pages 19–33, Taipei, Taiwan.
Jones, Michael P. and James H. Martin. 1997. Contextual spelling correction using Latent Semantic Analysis. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 166–173, Washington, DC.
Kanejiya, Dharmendra, Arun Kumar, and Surendra Prasad. 2003. Automatic evaluation of students' answers using syntactically enhanced LSA. In *Proceedings of the HLT-NAACL Workshop on Building Educational Applications Using Natural Language Processing*, pages 53–60, Edmonton, Canada.
Keenan, Edward and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8:62–100.
Kilgarriff, Adam. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.
Koeling, Rob, Diana McCarthy, and John Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the joint Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 419–426, Vancouver, Canada.
Landauer, Thomas and Susan T. Dumais. 1997. A solution to Platos problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.
Leacock, Claudia and Martin Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In Christiane Fellbaum, editor, *WordNet: A Lexical Reference System and its Application*. 1998, Cambridge, MA, pages 265–283.
Lee, Lillian. 1999. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, College Park, MA.
Lesk, Michael. 1986. Automatic sense disambiguation: How to tell a pine cone from an ice cream cone. In *Proceedings of the 1986 Special Interest Group in Documentation*, pages 24–26, New York. Association for Computing Machinery.
Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.
Levy, Joseph P. and John A. Bullinaria. 2001. Learning lexical properties from word usage patterns. In Robert French, editor, *Neural Network Models of Evolution, Learning and Development*. Springer, pages 273–282.
Lin, Dekang. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the joint Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics*, pages 768–774, Montréal, Canada.
Lin, Dekang. 1998b. Dependency-based evaluation of MINIPAR. In *Proceedings of the LREC Workshop on the Evaluation of*

*Parsing Systems*, pages 234–241, Granada, Spain.
Lin, Dekang. 1999. Automatic identification of non-compositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 317–324, College Park, MA.
Lin, Dekang. 2001. LaTaT: Language and text analysis tools. In *Proceedings of the 1st Human Language Technology Conference*, pages 222–227, San Francisco, CA.
Lin, Dekang and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):342–360.
Lowe, Will. 2001. Towards a theory of semantic space. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 576–581, Edinburgh, UK.
Lowe, Will and Scott McDonald. 2000. The direct route: Mediated priming in semantic space. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 675–680, Philadelphia, PA.
Lund, Kevin and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28:203—208.
Manber, Udi and Sun Wu. 1994. GLIMPSE: a tool to search through entire file systems. In *Proceedings of the USENIX Winter 1994 Technical Conference*, pages 23–32, San Francisco, CA.
Manning, Chris and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.
McCarthy, Diana, Bill Keller, and John Carroll. 2003. Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 73–80, Sapporo, Japan.
McCarthy, Diana, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, pages 280–287, Barcelona, Spain.
McDonald, Scott. 2000. *Environmental Determinants of Lexical Processing Effort*. Ph.D. thesis, University of Edinburgh.
McDonald, Scott and Chris Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Barcelona, Spain.
Mihalcea, Rada and Phil Edmonds, editors. 2004. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.
Miltsakaki, Eleni. 2003. *The Syntax-Discourse Interface: Effects of the Main-Subordinate Distinction on Attention Structure*. Ph.D. thesis, University of Pennsylvania.
Morris, Robin K. 1994. Lexical and message-level sentence context effects on fixation times in reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, (20):92–103.
Neville, Helen, Janet L. Nichol, Andrew Barss, Kenneth I. Forster, and Merrill F. Garrett. 1991. Syntactically based sentence prosessing classes: evidence form event-related brain potentials. *Journal of Congitive Neuroscience*, 3:151–165.
Patel, Malti, John A. Bullinaria, and Joseph P. Levy. 1998. Extracting semantic representations from large text corpora. In John A. Bullinaria, David W. Glasspool, and George Houghton, editors, *Proceedings of the 4th Neural Computation and Psychology Workshop: Connectionist Representations*, pages 199–212, London.
Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity – measuring the relatedness of concepts. In *Proceedings of the joint Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 38–41, Boston, MA. Demonstration system.
Pinker, Steven. 1989. *Learnability and Cognition: The Acquisition of Argument Structure*. The MIT Press, Cambridge, MA.
Rada, Roy, Hafedh Mili, and Ellen Bicknell. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30.
Resnik, Philip. 1995. Using information content to evaluate semantic similarity. In *Proceedings of 14th International Joint Conference on Artificial Intelligence*, pages 448–453, Montréal, Canada.
Rubenstein, Herbert and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.
Sahlgren, Magnus. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Stockholm

University.
Salton, G, A Wang, and C Yang. 1975. A vector-space model for information retrieval. *Journal of the American Society for Information Science*, 18:613–620.
Salton, Gerard and Michael J. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.
Salton, Gerard and Maria Smith. 1989. On the application of syntactic methodologies in automatic text indexing. In *Proceedings of the 12th ACM SIGIR Conference*, pages 137–150, Cambridge, MA.
Sampson, Geoffrey R. 1995. *English for the Computer*. Oxford University Press.
Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.
Strzalkowski, Tomek, editor. 1999. *Natural Language Information Retrieval*. Kluwer Academic Publishers, Dordrecht.
Talmy, L. 1985. Lexicalisation patterns: Semantic structure in lexical forms. In T. Shopen, editor, *Language Typology and Syntactic Description III: Grammatical Categories and the Lexicon*. Cambrige University Press, Cambridge, pages 57–149.
Tesnière, Lucien. 1959. *Elements de syntaxe structurale*. Klincksieck, Paris.
Turney, Peter D. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, pages 491–502, Freiburg, Germany.
Voorhees, Ellen M. 1999. Natural language processing and information retrieval. In *2nd School on Information Extraction (SCIE99)*, pages 32–48.
Weeds, Julie. 2003. *Measures and Applications of Lexical Distributional Similarity*. Ph.D. thesis, University of Sussex.
West, R.F. and K.E. Stanovich. 1986. Robust effects of syntactic structure on visual word processing. *Journal of Memory and Cognition*, 14:104–112.
Widdows, Dominic. 2003. Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In *Proceedings of the joint Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 197–204, Edmonton, Canada.
Wiemer-Hastings, Peter and Iraide Zipitria. 2001. Rules for syntax, vectors for semantics. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 1140–1145, Edinburgh, UK.
Yarowsky, David and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 9(4):293–310.

## Appendix A. Context Selection Functions

In what follows we present the context selection functions we used in our experiments. These are encoded as non-lexicalized path templates and are distributed as part of the software package that implements our dependency-based semantic space framework (see Section 3.7 for details). Each context selection function *cont* is represented by a set of path templates, $Temp(cont)$. Each path template directly corresponds to a path label sequence. Path templates are denoted by a comma-separated sequence of one or more edge labels; each edge label is a colon-separated triple `POS1:relation:POS2` (see Definition 1). The semantics of a set of path templates $Temp(c)$ is as follows: for a target word $t$ and a context selection function $c$, the context of $t$ consists of all paths $Pi_t$ (i.e., all paths anchored at $t$) so that there is a path template $temp \in Temp(c)$ which matches the label sequence $l(\pi_t)$.

**Minimum:**

```
A:amod:V
A:mod:A
A:mod:A
A:mod:N
A:mod:Prep
A:mod:V
A:subj:N
N:conj:N
N:gen:N
N:mod:A
N:mod:Prep
N:nn:N
N:obj:V
N:pcomp-n:Prep
N:subj:A
N:subj:N
N:subj:V
(null):lex-mod:V
Prep:mod:A
Prep:mod:N
Prep:mod:V
Prep:pcomp-n:N
V:amod:A
V:lex-mod:(null)
V:mod:A
V:mod:Prep
V:obj:N
V:subj:N
```

**Medium contains all `minimum` templates and:**

```
A:mod:N,N:lex-mod:(null)
A:mod:N,N:nn:N
A:subj:N,N:lex-mod:(null)
A:subj:N,N:nn:N
N:conj:N,N:lex-mod:(null)
N:conj:N,N:nn:N
N:gen:N,N:lex-mod:(null)
N:gen:N,N:nn:N
N:nn:N,N:conj:N
N:nn:N,N:conj:N,N:nn:N
N:nn:N,N:gen:N
N:nn:N,N:gen:N,N:nn:N
N:nn:N,N:mod:A
N:nn:N,N:mod:Pred
N:nn:N,N:obj:V
N:nn:N,N:subj:A
N:nn:N,N:subj:V
(null):lex-mod:N,N:conj:N
(null):lex-mod:N,N:conj:N,
  N:lex-mod:(null)
(null):lex-mod:N,N:gen:N
(null):lex-mod:N,N:gen:N,
  N:lex-mod:(null)
(null):lex-mod:N,N:mod:A
(null):lex-mod:N,N:mod:Pred
(null):lex-mod:N,N:obj:V
(null):lex-mod:N,N:subj:A
(null):lex-mod:N,N:subj:V
Prep:mod:N,N:lex-mod:(null)
Prep:mod:N,N:nn:N
V:obj:N,N:lex-mod:(null)
V:obj:N,N:nn:N
V:subj:N,N:lex-mod:(null)
V:subj:N,N:nn:N
```

**Maximum contains all `medium` templates and:**

```
A:mod:A,A:mod:N,N:lex-mod:(null)
A:mod:A,A:mod:N,N:nn:N
A:mod:Prep,Prep:pcomp-n:N,
  N:lex-mod:(null)
N:mod:Prep,Prep:pcomp-n:N,
  N:lex-mod:(null)
N:mod:Prep,Prep:pcomp-n:N,
  N:nn:N
N:nn:N,N:mod:A,A:mod:A
```

```
N:nn:N,N:mod:Prep,Prep:pcomp-n:N
N:nn:N,N:mod:Prep,Prep:pcomp-n:N,
  N:nn:N
N:nn:N,N:obj:V,V:subj:N
N:nn:N,N:obj:V,V:subj:N,N:nn:N
N:nn:N,N:pcomp-n:Prep
N:nn:N,N:pcomp-n:Prep,Prep:mod:N
N:nn:N,N:pcomp-n:Prep,Prep:mod:N,
  N:nn:N
N:nn:N,N:subj:V,V:obj:N
N:nn:N,N:subj:V,V:obj:N,N:nn:N
N:nn:N,V:s:C,C:fc:V
N:obj:V,V:subj:N,N:lex-mod:(null)
N:obj:V,V:subj:N,N:nn:N
N:pcomp-n:Prep,Prep:mod:N,
  N:lex-mod:(null)
N:pcomp-n:Prep,Prep:mod:N,N:nn:N
N:subj:V,V:obj:N,N:lex-mod:(null)
N:subj:V,V:obj:N,N:nn:N
(null):lex-mod:N,N:mod:A,A:mod:A
(null):lex-mod:N,N:mod:Prep,
  Prep:pcomp-n:N
(null):lex-mod:N,N:mod:Prep,
  Prep:pcomp-n:N,N:lex-mod:(null)
(null):lex-mod:N,N:obj:V,V:subj:N
(null):lex-mod:N,N:obj:V,
  V:subj:N,N:lex-mod:(null)
(null):lex-mod:N,N:pcomp-n:Pred,
  Prep:mod:A
(null):lex-mod:N,N:pcomp-n:Prep
(null):lex-mod:N,N:pcomp-n:Prep,
  Prep:mod:N
(null):lex-mod:N,N:pcomp-n:Prep,
  Prep:mod:N,N:lex-mod:(null)
(null):lex-mod:N,N:pcomp-n:Prep,
  Prep:mod:V
(null):lex-mod:N,N:rel:C,C:i:V
(null):lex-mod:N,N:subj:V,V:obj:N
(null):lex-mod:N,N:subj:V,V:obj:N,
  N:lex-mod:(null)
(null):lex-mod:N,V:s:C,C:fc:V
Prep:pcomp-n:N,N:lex-mod:(null)
Prep:pcomp-n:N,N:nn:N
V:fc:C,C:s:N,N:lex-mod:(null)
V:fc:C,C:s:N,N:nn:N
V:i:C,C:rel:N,N:lex-mod:(null)
V:mod:Prep,Prep:pcomp-n:N,
  N:lex-mod:(null)
```