Text: now in 2D! A framework for lexical expansion with contextual similarity

Chris Biemann and Martin Riedl
Computer Science Department, FG Language Technology,
TU Darmstadt, Germany
{biem,riedl}@cs.tu-darmstadt.de

ABSTRACT

A new metaphor of two-dimensional text for data-driven semantic modeling of natural language is proposed, which provides an entirely new angle on the representation of text: not only syntagmatic relations are annotated in the text, but also paradigmatic relations are made explicit by generating lexical expansions. We operationalize distributional similarity in a general framework for large corpora, and describe a new method to generate similar terms in context. Our evaluation shows that distributional similarity is able to produce high-quality lexical resources in an unsupervised and knowledge-free way, and that our highly scalable similarity measure yields better scores in a WordNet-based evaluation than previous measures for very large corpora. Evaluating on a lexical substitution task, we find that our contextualization method improves over a non-contextualized baseline across all parts of speech, and we show how the metaphor can be applied successfully to part-of-speech tagging. A number of ways to extend and improve the contextualization method within our framework are discussed. As opposed to comparable approaches, our framework defines a model of lexical expansions in context that can generate the expansions as opposed to ranking a given list, and thus does not require existing lexical-semantic resources.

Keywords: distributional semantics, lexical expansion, contextual similarity, lexical substitution, computational semantics

1 INTRODUCTION

In this article, we propose the new metaphor of two-dimensional text for data-driven semantic modeling of natural language and define a framework for its implementation. Being rooted in structural linguistics and distributional similarity, this metaphor provides a new angle on how to perform automated semantic processing. Whereas technically similar approaches have been pursued in the literature before, we feel that changing the viewpoint opens up new perspectives on how to advance the automated understanding of meaning in natural language.

The key element of this metaphor is the concept of *lexical expansion*. Lexical expansion generates additional lexical items for a given chunk of text, which enrich the textual representation and may be used in NLP (Natural Language Processing) tasks and applications. Expansion is performed for all present lexical items, and taking into account the textual context. Our approach constitutes a generative unsupervised model for semantic similarity in context that can be used to generate lexical expansions for unseen text material. These expansions help to bridge the lexical gap in semantics and serve as a valuable preprocessing step for many approaches in computational semantics, like word sense disambiguation, semantic text similarity, passage scoring and text segmentation.

After giving a short history of ideas that led from linguistic structuralism to the notion of distributional similarity and providing pointers to related work, we will map out the metaphor of two-dimensional text and explain the development from distributional to contextual similarity. Section 2 is concerned with operationalizing these notions in a scalable computational framework. In Section 3, we evaluate our methodology against a lexical resource and against a lexical substitution data set and show the value of the approach both for distributional as well as for contextual similarity. Sections 4 and 5 conclude and lay out possible points of departure for further work.

1.1 *From linguistic structuralism to distributional similarity*

What happens if we 'understand' language in the sense of assigning values of meaning to its elements, e.g. when reading a text? According to de Saussure (1916, 1959), our analysis happens from two dis-

tinct viewpoints: the *syntagmatic* viewpoint is concerned with assigning values based on the linear sequence of language elements, and the *associative (also: paradigmatic)* viewpoint assigns values according to the commonalities and differences to other language elements in the reader's memory.

> We see that the co-ordinations formed outside discourse differ strikingly from those formed inside discourse. Those formed outside discourse are not supported by linearity. Their seat is in the brain; they are a part of the inner storehouse that makes up the language of each speaker. They are associative relations. [...] The syntagmatic relation is in praesentia. It is based on two or more terms that occur in an effective series. Against this, the associative relation unites terms in absentia in a potential mnemonic series. (de Saussure, 1959, p.123)

In the metaphor of two-dimensional text, we propose to represent language in two dimensions: The first dimension is given by the linear nature of language, and represents syntagmatic relations between language elements, i.e. grammatical dependencies, positional relations or others. The second dimension contains language elements that are not present in the first dimension, but stand in paradigmatic relation to the language elements present. Figure 1 exemplifies possible associations for terms, and visualizes them in a second dimension, which we aim to model explicitly within our metaphor. The first dimension represents the linear sequence of language elements and their syntagmatic relations, the second dimension models associative relations that reside in the memory of the speaker/receiver. In this way, a text expansion step is realized.

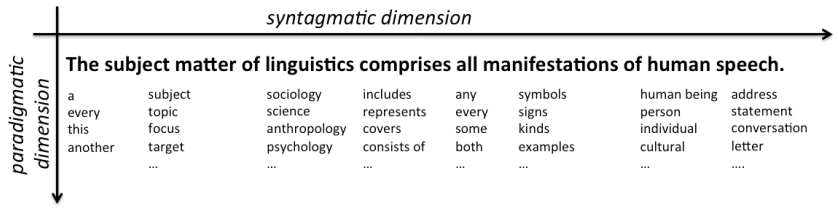

Figure 1: Exemplification of the metaphor of two-dimensional text

Please note that our metaphor specifies neither the language elements (words, terms, phrases etc.) nor the relation between the present elements and their expansions. The only constraint is that expansions in the paradigmatic relation share some commonality with their respective element. As de Saussure (1959, p.125) already states: "Mental

association creates other groups besides those based on the comparing of terms that have something in common; through its grasp of the nature of the relations that bind the terms together, the mind creates as many associative series as there are diverse relations." From an application-based perspective in Natural Language Processing, it is easy to imagine that some of such relations might prove more useful than others when operationalizing the two-dimensional text for a given task. Further note that expansions in the paradigmatic dimension need to be contextualized to the present language elements. For example, in the sentence "almost all old subject case forms disappeared in French", "subject" would be expanded differently than is shown in Figure 1.

Many decades after the foundational work of Ferdinand de Saussure, Zellig S. Harris formulated his *distributional hypothesis*:

> The distribution of an element is the total of all environments in which it occurs, i.e. the sum of all the (different) positions (or occurrences) of an element relative to the occurrence of other elements. Two utterances or features will be said to be linguistically, descriptively, or distributionally equivalent if they are identical as to their linguistic elements and the distributional relations among these elements. (Harris, 1951, pp. 15f.)

Harris (1951) used the term *environments* to denote the language elements that stand in a syntagmatic relation to the element that is characterized. Note that an environment is not a language element, but an arbitrarily complex structure. However, we will approximate the environment with a tuple consisting of language elements and the syntagmatic relation, which we will call a *context feature*.

Whereas the distributional hypothesis was defined in the context of structural linguistics and originally formulated in order to identify phonetic variants of the same phoneme, it was not operationalized for computational semantics and cognitive science until about four decades later. After departing from an absolute notion of synonymy and instead focusing on semantic similarity as a graded notion, the *strong contextual hypothesis* of Miller and Charles (1991) states that "Two words are semantically similar to the extent that their contextual representations [context features] are similar". This suggests the following approach: using large text corpora to collect context features for language elements and comparing the extent to which these lan-

guage elements share the same context features. This provides a way to compute semantic similarity without resorting to dictionary definitions or lexical resources. Miller and Charles (1991) were able to show that human judgments on semantic similarity as pioneered by Rubenstein and Goodenough (1965) correlate highly with the similarity of their context representations.

With the advent of large text corpora and reasonably precise methods to automatically assign grammatical structure to sentences, it became possible to compute term similarities for a large vocabulary (Ruge, 1992). Lin (1998) computed a *distributional thesaurus* (DT) by comparing context features defined over grammatical dependencies with an appropriate similarity measure for all reasonably frequent words in a large collection of text, and to evaluate these automatically computed word similarities against lexical resources. Entries in the DT consist of a ranked list of the globally most similar language elements (here: words) per language element of interest, which we call the *target*. While the similarities are dependent on the instantiation of the context feature as well as on the underlying text collection, they are global in the sense that the DT does not provide similarities with respect to particular occurrence of a target, but rather aggregates over all occurrences of the target and its similar elements.

We will build on the notion of the distributional thesaurus in our work, use the DT entries to populate the second dimension in the two-dimensional text representation, and move from the global notion of similarity to a contextualized version, which allows performing context-dependent text expansion for previously unseen target occurrences.

A similar review of the connection of de Saussurian linguistics and distributional similarity was presented in Sahlgren (2006). While Sahlgren motivated vector-space approaches to modeling meaning, we would like to stress that the two-dimensional text metaphor has not previously been employed as an approach to statistical semantics.

1.2 *Related work*

There has been a steady increase of interest towards incorporating distributional similarity into Natural Language Processing applications, particularly into language models. Whereas the workhorse of language modeling – the n-gram model – is a reliable and well-understood com-

ponent in NLP systems, it models only very local properties of language and has been shown to be inadequate to grasp semantic dimensions of language such as ambiguity and synonymy (Biemann *et al.*, 2012).

Since local syntax could be modeled with a simple n-gram model, a desire to model semantics in a similarly straightforward fashion (i.e. trained from a background corpus without the need for linguistic theories, rule bases or knowledge bases) sparked a large body of research on semantic modeling. This includes computational models for topicality (Deerwester *et al.*, 1990; Hofmann, 1999; Blei *et al.*, 2003), and language models that incorporate topical (as well as syntactic) information (Boyd-Graber and Blei, 2008; Tan *et al.*, 2012). In the Computational Linguistics community, the vector space model (Schütze, 1993; Turney and Pantel, 2010; Baroni and Lenci, 2010) is the prevalent metaphor for representing word meaning. Vector space operations can be represented as vector and matrix operations, which makes this easily implementable due to the availability of tools such as MATLAB and libraries such as the GNU Scientific Library.

We do not agree that "nouns are vectors, and adjectives are matrices" (Baroni and Zamparelli, 2010), although they can of course be *represented* in these or similar ways. While vector space representations are becoming increasingly successful in modeling natural language semantics, vectors are typically too sparse and too highly dimensional to be used in their canonical form, and do not (naturally) encode relations beyond undifferentiated co-occurrence. We argue that there is no need to explicitly model non-existing relations, which would be zeros in the vector representation. We posit that it is only worthwhile storing properties for words or concepts if those same properties would be explicitly represented (non-zero) in a sparse representation.

Baroni and Lenci (2010) propose to store word-link-word triples in a tensor, and to produce vector spaces of various flavors by projection. While this model is a significant step towards a more generalized representation of (structured) vector spaces, it lacks the capability to address relations of higher complexity than single relations. Since in operationalizaton, similarity computations are carried out on pairs, we pursue a slightly different route in our holing system (see Section 2.1): we refrain from storing the tensor, and directly produce pairs from the observed structures in the text. Our formulation is thus able

to produce the same behavior as the proposal of Baroni and Lenci (2010), but is more flexible and generic.

While computing semantic similarity on the basis of a background corpus produces a global model, which e.g. contains semantically similar words for different word senses of a target word, there are a number of works that aim at *contextualizing* the information held in the global model for *particular* occurrences. This is a similar task to word sense disambiguation against a lexical resource (Lesk, 1986), but without presupposing the existence of such a resource.

With his predication algorithm, Kintsch (2001) contextualizes the Latent Semantic Analysis (LSA) model (Deerwester *et al.*, 1990) for N-VP constructions by spreading activation over neighborhood graphs in the latent space. The Latent Dirichlet Allocation (LDA) model (Blei *et al.*, 2003) uses an inference step in order to adjust the topic distributions of the target occurrences. In particular, the question of operationalizing semantic compositionality in vector spaces (Mitchell and Lapata, 2008) received much attention and triggered shared evaluation tasks (Biemann and Giesbrecht, 2011; Padó and Peirsman, 2011): how can the (vector) representation of two lexical items be combined in context to yield an appropriate representation of their combination? Mixed results in favor of one or the other combination or mutual contextualization method (Mitchell and Lapata, 2008; Giesbrecht, 2009; Guevara, 2011) either indicate a dependency on the particular task, or raise questions regarding the representation itself.

Today's vector space representations suffer from two major shortcomings. First, size issues have to be handled with singular value decomposition (Golub and Kahan, 1965),[1] random indexing (Sahlgren, 2006) or other necessarily lossy dimensionality reduction techniques. Alternatively, efficient representations based on hashing functions (e.g. Goyal *et al.*, 2012) are employed to keep model estimation and computation at application time feasible. These issues arise as the word space is highly dimensional, and more structured variants (Padó and Lapata, 2007) that incorporate grammatical relations into the model lead to a further increase in the number of dimensions. Second, and more importantly, vector space models are not generative:

[1] The singular value decomposition is an algebraic factorization, which is used in LSA.

while impressive results are obtained when ranking a set of given alternatives by similarity of vector representation and context (e.g. word sense discrimination, Schütze 1998, synonyms, Rapp 2003, paraphrases, Erk and Padó 2008, word sense disambiguation, Thater *et al.* 2011), these tasks presuppose an existing list of alternatives to begin with.[2] Ideally, the alternatives should also originate from the model itself so as to avoid the manual creation of lexical resources for each language or application domain. We stress the need for a model that not only is able to rank given alternatives, but is also able to produce them.

2 OPERATIONALIZING SEMANTIC SIMILARITY

In this section, we describe how to operationalize semantic similarity. We describe a scalable and flexible computation of a Distributional Thesaurus (DT), and the contextualization of distributional similarity for specific occurrences of language elements (i.e. words). Care is taken to abstract away from particular preprocessing tasks needed for a given data set and from particular measures of similarity. Further, no assumptions regarding the size of the vocabulary nor the memory of the processors are made. For related works on the computation of distributional similarity, see Lin (1998), Gorman and Curran (2006), Lin and Dyer (2010), inter alia.

2.1 *Holing system*

To keep the framework flexible and abstract with respect to the preprocessing that identifies structure in language material (e.g. text or speech), we introduce the holing operation. Given a particular observation (structural representation) that has previously been extracted from the text (e.g. a dependency parse or an n-gram representation), the holing operation creates two distinct sets of observations: *language elements* (also referred as *terms*), and their respective *context features*. These two sets of observations form the basis for the computation of global similarities (Section 2.2) and for their contextualization (Section 2.3). Note that the holing operation is necessarily coupled to the

[2] Looping over the entire vocabulary to remove this restriction is neither computationally feasible nor plausible.

particular structural representation created by the pre-processing step, but all further steps towards contextual similarity abstract away from such pre-processing and operate on the same representation.

In the general case, an observation on the syntagmatic structure can be represented as an n-tuple containing an identifier of the observation, and the language elements that are part of the observation. We shall use the following sentence as the basis for examples:

```
Sentence:  I gave a book to the girl
Positions: 1 2    3 4    5  6   7
```

2.1.1 Observations

Let us now look at two different observations: dependency parses and token 4-grams. The collapsed dependency parse (Marneffe *et al.*, 2006) yields the following list of observations:

a) Dep.Parse:
(nsubj;gave$_2$;I$_1$), (det;book$_4$;a$_3$), (dobj;gave$_2$;book$_4$),
(det;girl$_7$;the$_6$), (prep_to;gave$_2$;girl$_7$)

Another pre-processing step that e.g. splits the language material into token 4-grams could produce these observations on the same sentence:

b) 4-gram:
($0;I$_1$;gave$_2$;a$_3$), (I$_1$;gave$_2$;a$_3$;book$_4$),
(gave$_2$;a$_3$;book$_4$;to$_5$), (a$_3$;book$_4$;to$_5$;the$_6$),
(book$_4$;to$_5$;the$_6$;girl$_7$), (to$_5$;the$_6$;girl$_7$;$$_8$),
(the$_6$;girl$_7$;$$_8$;$$_9$)

2.1.2 Holing operation

For a given set of observations extracted during pre-processing, a holing operation has to be defined that performs the split into language element(s) and context features. In the following examples the language element will be a word. However, the holing operation is not restricted to single words: arbitrary binary masks to define the parts of the observation tuples can be applied. For our example, we

assume that we want to characterize single observed words a) by the dependency relation and the word it is connected to, and b) by the surrounding 4-gram context, where the observed word is located at the second position in the 4-gram. Further, we want to characterize pairs of observed words c) by their connecting two-edge dependency path. The application of the holing operation results in a set of pairs $< x, y >$ that identify the holing operation, as well as the parts it results in. The position of the language element x in its context tuple y is indicated by the hole symbol "@". For the single word examples, this could look like this:

a) Dep.Parse:

<I$_1$,(nsubj;gave$_2$;@)>, <gave$_2$,(nsubj;@;I$_1$)>, <book$_4$, (det;@;a$_3$)>,
<a$_3$, (det;book$_4$;@)>,..., <gave$_2$, (prep_to;@;girl$_7$)>,
<girl$_7$, (prep_to;gave$_2$;@)> .

b) 4-gram, second position:

<I$_1$, ($$_0$;@,gave$_2$;a$_3$)>, <gave$_2$, (I$_1$;@;a$_3$;book$_4$)>,
<a$_3$, (gave$_2$;@;book$_4$;to$_5$)> , ..., <girl$_7$, (the$_6$;@;$$_8$;$$_9$)> .

For characterizing the pairs, the first part of the tuple is actually an ordered pair, and the second part contains two holes:

c) Dep.Parse two-edge paths:

<(I$_1$,book$_4$), (nsubj;gave$_2$;@$_1$;dobj;gave$_2$;@$_2$)>,
<(I$_1$,girl$_7$), (nsubj;gave$_2$;@$_1$;prep_to;gave$_2$;@$_2$)>,
<(gave$_2$,a$_3$),(dobj;@$_1$;book$_4$;det;book$_4$;@$_2$)>,
<(gave$_2$,the$_6$),(prep_to;@$_1$;girl$_7$;det;girl$_7$;@$_2$)>,
<(book$_4$,girl$_7$), (dobj;gave$_2$;@$_1$;prep_to;gave$_2$;@$_2$)> .

Note that a single observation can result in multiple pairs, as shown in a), where a dependency produces two pairs. Also, some observations need not produce any pairs, e.g. when deciding to exclude the `det` dependency relation, or constraining contexts along particular relations (cf. Lee, 1999).

The result of the holing operation, i.e. the list of pairs as shown above, is the only representation that further steps operate on. The pairs fully encode observed language elements and their contexts. For

the computation of distributional similarity, the positional indices will be ignored, but they are required for the contextual expansion step.

The representation as shown here is more general than representations used by e.g. Lin (1998) and Curran (2004): whereas these previous works only allow a single term to be characterized with features, we allow arbitrary splits over arbitrarily complex observations, as shown in example c). This gives rise to the comparison of pairs, as e.g. conducted by Turney and Littman (2005) for extracting analogies of semantic relations in what they call *relational similarity*.

For the remainder of this paper, however, we mostly stick to the notion of *attributional similarity*, which is the basic element of the two-dimensional text expansion described above.

2.2 *MapReduce for similarity computation*

We now describe an implementation of the similarity computation for the Distributional Thesaurus (DT) based on the Apache Hadoop MapReduce framework,[3] which allows parallel processing of large (textual) data. The principle, developed by Dean and Ghemawat (2004), uses two steps, namely Map and Reduce. The Map step converts input text to key-value pairs, sorted by key. The Reduce step operates on all values that have the same key, producing again a data table with a key. As these steps do not require a global information flow, many Map and Reduce steps can be executed in parallel, allowing the system to scale to huge amounts of data. Further, we use Apache Pig,[4] a query language similar to SQL that allows us to perform database joins, sorting and limit operations on Hadoop data tables. To explain the workflow, we will refer to a holing system that extracts single terms as language elements for simplicity. However, the same workflow can be executed for more complex holing systems.

The data flow of the DT is illustrated in Figure 2. The example shown in this workflow uses a text file as input, where each line contains one sentence. The first MapReduce step in the workflow, called the *Context Feature Extractor*, implements a single holing operation as described in Section 2.1. For example, in Figure 2, the language element (which we will also call a term) is a word, concatenated with

[3] `http://hadoop.apache.org`

[4] `http://pig.apache.org/`

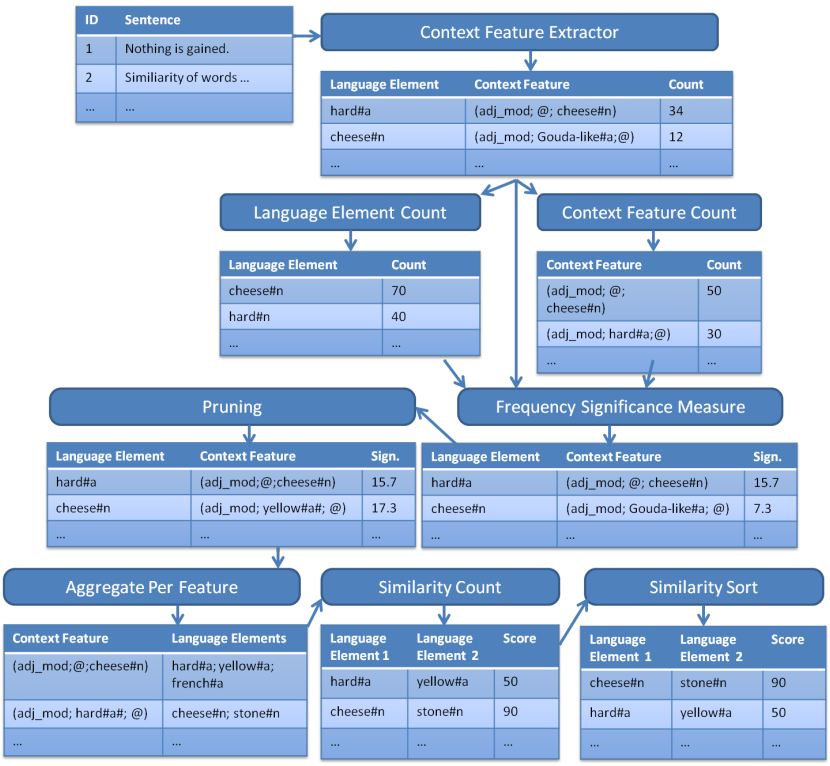

Figure 2: Workflow of the data processing using MapReduce

the corresponding part-of-speech; and the context feature is the dependency relation. Note that positional offsets are dropped here. For different holing operations (e.g. dependencies or 4-grams as in the previous section), the computation is executed separately.

In the next step, the frequencies of terms (*Language Element Count*) and single contexts (*Feature Count*) are collected, as they are needed to calculate the significance of each feature-term pair. For this work, we implemented different significance measures in *Frequency Significance Measure* and evaluate them in Section 3.2. For computing these measures, the tables produced by *Language Element Count* and *Feature Count* are joined to the table holding frequencies of term-feature pairs using an Apache Pig script. For a similar computation of word co-occurrences, Lin and Dyer (2010) propose to load the single frequencies into memory to avoid the join operation and to speed up the overall computation. While this works for a limited (albeit large) vo-

cabulary of terms when carefully tuning the number of Mappers per computation node, this imposes a severe limitation on the number of (arbitrarily complex and productive) context features, which is why we do not adhere to this design pattern.

There are a total of three parameters for pruning the data during the *Pruning* step: t as a lower bound for the term-feature counts, s as a lower bound for the score of the respective significance measure, and p regulating the maximum number of context features per term. We argue that it is sufficient to keep only the p most salient features per term, as features of low saliency generally should not contribute much to the similarity of terms, and also could lead to spurious similarity scores. These pruning steps are especially important when using large data sets. The influence of the parameters on the quality of the DT will be examined in detail in Section 3.2.

Afterwards, all terms are aggregated by their features (*Aggregate Per Feature*), which allows us to compute similarity scores between all terms that share at least one feature (*Similiarity Count*). Here, we skip very frequent features (such as determiner modifiers), as they do not contribute meaningfully to similarities despite increasing computation time.

In comparison, Lin (1998) and Curran (2002) specify the similarity of terms using an "information" formula for each term-context relation and then calculate the similarity between terms using similarity measures. We show our similarity measure, as well as the measure used by Lin (1998) and a measure recommended by Curran (2002) in Table 1.

Function $f(.)$ returns the frequency of the selected element and $p(.)$ returns the probability. In contrast to the notation of Lin and Curran, we combine the relation name and the feature elements. To formulate Lin's information measurement in this notation, we define a $relation(.)$ function, which extracts only the relation name for a given context feature, and a $feature(.)$ function, returning all features for a term. Comparing our approach to other distributional similarity measurements (cf. Lee, 1999; Lin, 1998; Weeds, 2003), we do not need a "two-staged" formula, but can directly calculate the similarity by counting the overlap of features of two terms. This has the advantage that we do not need to calculate similarities between all pairs. Additionally, using only the p features per term having the

Table 1: Similarity measures used for calculating the distributional similarity between terms

Information measurements	
Lin's formula	$I(term,feature) = lin(term,feature) = = \log \frac{f(term,feature) * f(relation(feature))}{\sum(f(word,relation(feature))f(word)}$
Curran's t-test	$I(term,feature) = ttest(term,feature) = = \frac{p(term,feature) - p(feature) * p(term)}{\sqrt{p(feature) * p(term)}}$
Similarity measurements	
Lin's formula	$sim(t_1,t_2) = \frac{\sum_{f \in features(t_1) \cap features(t_2)} (I(t_1,f) + I(t_2,f))}{\sum_{f \in features(t_1)} I(t_1,f) + \sum_{f \in features(w_2)} I(w_2,f)}$
Curran's dice	$sim(t_1,t_2) = \frac{\sum_{f \in features(t_1) \cap features(t_2)} \min(I(t_1,f), I(t_2,f))}{\sum_{f \in features(t_1) \cap features(t_2)} (I(t_1,f) + I(t_2,f))}$
Our measure	$sim(t_1,t_2) = \sum_{f \in features(t_1) \cap features(t_2)} 1$
w. filtering	$sim(t_1,t_2) = \sum_{\substack{f \in rankedfeatures(t_1,p) \cap rankedfeatures(t_2,p) \\ f(t_1) > t \wedge f(t_2) > t \\ score(f) > s \wedge score(f) > s}} 1$

highest significance scores (which are retrieved using the function $rankedfeatures(term, p)$) speeds up our approach tremendously and acts as a noise filter.

This constraint makes this approach more scalable to larger data, as we do not need to know the full list of features for a term pair at any time. As we will demonstrate in Section 3, this simplification does not impair the quality of the obtained similarities, especially for very large corpora.

The last step sorts the list by term and by descending score. To reduce the size of the output, only the most similar n terms per entry are kept. The overall computation results in second order (paradigmatic) similarity scores that are ready to be imported to a storage database, as to be accessible for the contextualization component. Further, we store the first order (syntagmatic) significant pairs $< x, y >$, together with their significance score, as we will need them for contextualization.

Our small Hadoop cluster (64 cores on 8 servers) was able to perform the entire computation (excluding pre-processing, i.e. parsing) of our similarity measure for the whole vocabulary of our largest corpus

of 120 million sentences in well under a day. Within our framework, we also provide Pig scripts for the computation of other similarity measures (cf. Table. 1), although they take much longer to compute. The implementation is available via the JoBimText[5] project as open-source software under the ASL 2.0 for download.

2.3 *Contextualizing distributional similarity*

Now, we explore a way of contextualizing semantic similarity. The task of contextualization is cast as a ranking problem (in accordance with most literature on lexical substitution): given a set of candidate expansions as provided by the DT, we aim at ranking them so that the most similar terms in context will be ranked higher. Intuitively, candidates that are not compatible with the given context should be ranked lower, whereas candidates that fit well should land on top of the list.

When expanding a target, we run the holing system on the lexical material containing our target, and select all pairs $< x, y >$ where $x =$ target. Further, we obtain a set of candidate expansions X' by selecting the most similar n terms from the DT entry of the target. For each pair, we iterate over the elements x' in X' and retrieve the significance score of $< x', y >$. If the candidate expansion has been observed in the context of y before, this will result in a positive score. If the candidate has not been observed, it is probably incompatible with y and gets assigned a score of 0 for this context. In this way, each candidate x' gets as many scores as there are pairs containing x in the holing system output. An overall score per x' is then calculated as the harmonic mean of the add-one-smoothed single scores. Smoothing is necessary to be able to rank candidates x' that are not compatible with all contexts.

In Figure 3, we illustrate this using the noun target "cold" in the sentence "I caught a nasty cold". Our dependency-parse-based holing system produced the following pairs for "cold":

<cold$_5$,(amod;@;nasty$_4$)>, <cold$_5$,(dobj;caught$_2$;@)> .

The top 10 candidates for "cold" as a noun are $X' =$ {heat, weather, temperature, rain, flue, wind, chill, disease}. In Figure 3, the scores per pair are listed: e.g. the pair <heat,(dobj;caught;@)>

[5] http://sourceforge.net/p/jobimtext/wiki/Home/

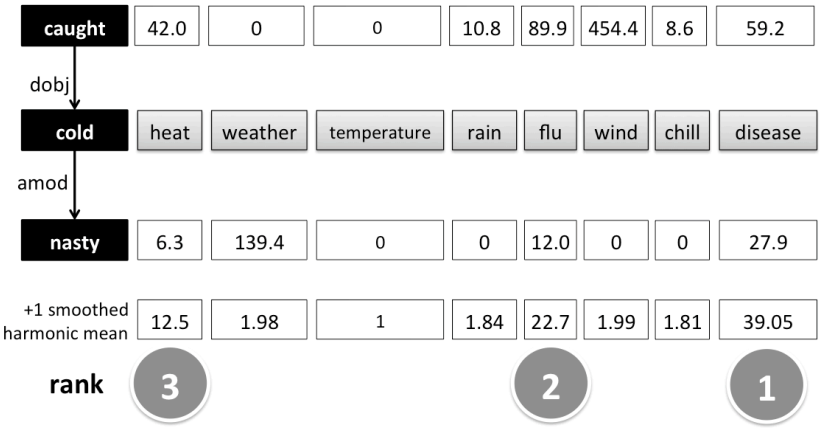

Figure 3: Contextualized ranking for target “cold” in the sentence “I caught a nasty cold” for the 10 most similar terms from the DT (here: 10 million sentences, LMI, $p = 1000$)

has a Lexicographer’s Mutual Information (LMI) score of 42.0, the pair `<weather,(amod;@;nasty)>` has a score of 139.4, and the pair `<weather,(dobj;caught;@)>` was not contained in our first-order data. Ranking the candidates by their overall scores as given in the figure, the top three contextualized expansions are “disease, flu, heat”, which are compatible with both pairs. For $n = 200$, the ranking of fully compatible candidates is: “virus, disease, infection, flu, problem, cough, heat, water”, which is clearly preferring the disease-related sense of “cold” over the temperature-related sense.

Context features differ in their usefulness: a context feature like `(det; @; a)` is much less useful for ranking expansions than context features with more specific language elements, such as `(amod; tasty; @)`, which e.g. selects edibles and thus could distinguish between “Turkey” the country and “turkey” the bird. To compensate for this effect, we found it advantageous to divide the score by the corpus frequency of the context feature language element, and to only take context features containing content words (i.e. nouns, verbs, adjectives) into account. Of course, many more weighting schemes would be possible.

Iterating the per-word expansion over the whole sentence to expand all the terms yields a two-dimensional contextualized text.

3 EVALUATING TWO-DIMENSIONAL TEXT

Directly evaluating the quality of a (non-contextualized) DT is intrinsically hard. It is known that distributional similarity somewhat reflects semantic relations in lexical resources, but it is clear that a DT will never correspond exactly to a lexical resource, e.g. for the reasons of vocabulary mismatch, skewed word sense distributions in the underlying collection and rare senses in the resource, cf. Curran (2002) and Henestroza Anguiano and Denis (2011). We follow a pragmatic approach and evaluate DTs of different parameterizations against WordNet, using a new path-based approach. While the aforementioned shortcomings make it hard to draw conclusions about the absolute quality of the DTs, our evaluation methodology still allows to compare DTs relatively to each other.

Regarding the contextualization, we chose to evaluate our technique in lexical substitution tasks. We stress again that – as compared to previous methods – we do not use a lexical resource for substitution candidates, but generate them using the DT. Therefore, our overall system solves a harder task than merely ranking a given set of alternatives.

Finally, we show how to apply our two-dimensional text processing to an existing NLP system that performs part-of-speech tagging in Section 3.4. In the same way, other existing NLP components could be extended by this two-dimensional representation.

3.1 *Data sets and methodology*

For DT evaluation, we use a word list of English nouns of varying frequency. For evaluation of the contextualization, we use two different lexical substitution data sets. We briefly describe the two datasets and the metrics we used in each case:

- **1000 frequent and 1000 infrequent nouns using WordNet path similarity**
  To evaluate our method under several parameter settings and against previous measures, we use the list of 1000 frequent and 1000 infrequent nouns from the British National Corpus previously employed in Weeds (2003). To calculate similarity scores between these target words and the most similar words in the distributional thesauri, we use the WordNet::Similarity path measure

(Pedersen *et al.*, 2004). For pairs of words that are members of several synsets, we use the shortest path between them. While the path measure has been criticized because of the varying granularity in different regions of WordNet, it is well-suited for relative comparison and has an intuitive interpretation: two words are fairly similar if the shortest route between them is small, and are less similar if the shortest route between them is long.

- **Lexical Substitution Task 2007 dataset (LexSub)**
  The LexSub[6] data were introduced in the Lexical Substitution task at Semeval 2007 (McCarthy and Navigli, 2009). It consists of 2010 sentences for a total of 201 target words (10 sentences for each word). For each target in context, five English native speaker annotators were asked to provide as many paraphrases or substitutions as they found appropriate. This way, valid substitutions are assigned a weight (or frequency) which denotes how many annotators suggested that particular word. We used the evaluation methodology as provided by the task organizers, tuned our approach on the trial data (300 sentences), and evaluated on the official test data (1710 sentences).

3.2 *Distributional similarity*

For computing the DT, we used newspaper corpora of up to 120 million sentences (about two gigawords), compiled from freely available corpora from LCC[7] and from the Gigaword corpus (Parker *et al.*, 2011). We examine the influence of the corpus size by computing DTs on corpora of different magnitudes, and evaluate the influence of parameters and significance measures.

3.2.1 Evaluation methodology

In this work, two different holing systems were used in the first step of the DT computation:

- As a simple baseline holing system, we employ token bigrams: for each token, the preceding and the following word are used as con-

[6] `http://nlp.cs.swarthmore.edu/semeval/tasks/task10/data.shtml`

[7] Leipzig Corpora Collection, `http://corpora.uni-leipzig.de`, (Richter *et al.*, 2006).

text features. This holing system uses information that is equivalent to the information available in a bigram language model.

- As a more informed holing system, we use collapsed dependency parses from the Stanford parser,[8] as depicted in Figure 2 and as described in Section 2.1.

To avoid confusion between words with different part-of-speech (POS) tags, we do not use the word itself, but rather the lemmatized[9] word combined with a POS tag[10] for both holing systems.

For all corpora, we only calculated similarities based on single word expressions and did not address multiword expressions, which is subject to further work. For this reason, we ignored multi-word entries in our evaluation data sets entirely.

3.2.2 Evaluation of DT parameters

In an initial exploration, we use 10 million sentences from the LCC to compute DTs for different parameters. We do not filter on occurrence frequency t and significance thresholds s, but merely vary the number of context features per term p. This parameter has a direct consequence for the run-time of the DT computation and the intermediate and final disk space.

To rank context features by their significance, we compare three significance measures,[11] two of which we show in Table 2:

- PMI Pointwise Mutual Information: a widely used significance measure since its introduction to NLP by Church and Hanks (1990).
- LMI Lexicographer's Mutual Information (Kilgarriff *et al.*, 2004), also known as Local Mutual Information (Evert, 2005): since PMI is known to assign high significance scores to pairs formed by low-frequent items, the LMI measure tries to balance this by multiplying the PMI score with the pair frequency.

[8] `http://nlp.stanford.edu/software/lex-parser.shtml`, (Marneffe *et al.*, 2006).

[9] The verbs, nouns and adjectives are lemmatized, using a Compact Patricia Trie classifier (Biemann *et al.*, 2008) trained on the verbs, nouns and adjectives.

[10] As produced by the Stanford parser.

[11] For a comparison of measures, see e.g. Evert (2005) and Bordag (2008).

- LL Log-likelihood: also a widely used measure since it was introduced by Dunning (1993), known to be less susceptible to overestimation of low frequency pairs. We omit its lengthy expanded formula here, which can be found e.g. in Bordag (2008).

Table 2: Significance measures used to rank the term feature pairs

PMI	$\mathrm{PMI}(term, feature) = \log_2 \left(\frac{f(term, feature)}{f(term) f(feature)} \right)$
LMI	$\mathrm{LMI}(term, feature) = f(term, feature) \log_2 \left(\frac{f(term, feature)}{f(term) f(feature)} \right)$

The results are calculated based on the 1000 frequent and 1000 infrequent target nouns. Average WordNet path similarities are computed between the target and the highest-ranked 5 and 10 words in its DT entry that occur in WordNet. For words invoking several synsets, we compute all possible pairs and use the minimal path distance. The results for the 1000 frequent nouns are shown in Table 3.

Note that the PMI measure does not play well with our pruning scheme regulated by the p parameter: while the other two measures yield very similar scores, PMI produces clearly inferior results. This confirms previous observations that PMI overestimates context features with low frequency: these context features might characterize the terms extremely well, but are too sparse to serve as a basis for the computation of second-order similarity (cf. Bordag, 2008). For high-frequency words, the most significant context features ranked by PMI are largely rare contexts of high specificity, whereas for low-frequency

Table 3: Wordnet Path Similarity for 1000 frequent nouns for DTs computed on 10 million sentences

Top	Sign.	max number of context features p				
words	Meas.	10	100	300	500	1000
top10	LL	0.04178	0.25744	0.27699	0.27635	0.27574
top10	LMI	0.03636	0.25449	0.27746	0.27554	0.27530
top10	PMI	0.00000	0.00213	0.04480	0.09104	0.16877
top5	LL	0.12034	0.29345	0.31106	0.31515	0.31182
top5	LMI	0.11666	0.29272	0.31378	0.31307	0.31028
top5	PMI	0.00000	0.00510	0.05836	0.11063	0.19268

words, this problem is less severe since there are fewer contexts to begin with, and so the top 1000 PMI contexts contain enough context features to produce similarities almost on par with the other measures.

More interestingly, there seems to be an optimal value for p, as more context features apparently do not improve the similarity and the highest values are obtained for $p = 300$ in this experiment. However, degradation for larger values of p is small. Values for average path similarities over the top 5 words are consistently higher than for the top 10 words, indicating that the ranking is valid with respect to semantic closeness.

Looking at the results of the infrequent nouns (see Table 4), we observe much lower average values throughout.

This is partially due to the words in the given noun list that do not have an entry in the DT at all; but more plausibly the lack of overall data for these words causes less reliable similarities. A further reason is the incomplete WordNet coverage for senses that are dominant in our collection. For example, the word *anime* belongs to two synsets: "a hard copal derived from an African tree" and "any of various resins or oleoresins", whereas an entry for *anime* in the sense of the Japanese animation movie is missing. The entries of the DT using LMI and $p = 500$ contains "novel, music, manga, comic, cartoon, book, film, shows, scifi", which all receive a low score. For infrequent words, the difference between PMI and the other measures is much less pronounced, yet we can still safely conclude from these experiments that PMI is not the optimal measure in our setup.

Table 4: Wordnet Path Similarity for 1000 infrequent nouns for DTs computed on 10 million sentences

Top	Sign.	max number of context features p				
words	Meas.	10	100	300	500	1000
top10	LL	0.03252	0.18560	0.20426	0.20572	0.20238
top10	LMI	0.03349	0.18516	0.20315	0.20577	0.20373
top10	PMI	0.00000	0.05892	0.14757	0.16597	0.16931
top5	LL	0.09268	0.21497	0.23231	0.23680	0.23108
top5	LMI	0.09469	0.21512	0.23208	0.23541	0.23179
top5	PMI	0.00012	0.10502	0.17446	0.18966	0.19318

For the next experiment, we examine the influence of corpus size and the difference between using dependency parses or neighboring tokens, again evaluating against our set of frequent and infrequent nouns using WordNet path similarity. Figure 4 displays the average WordNet path similarity score for the top-ranked five words for the 1000 frequent nouns (infrequent nouns show qualitatively similar results).

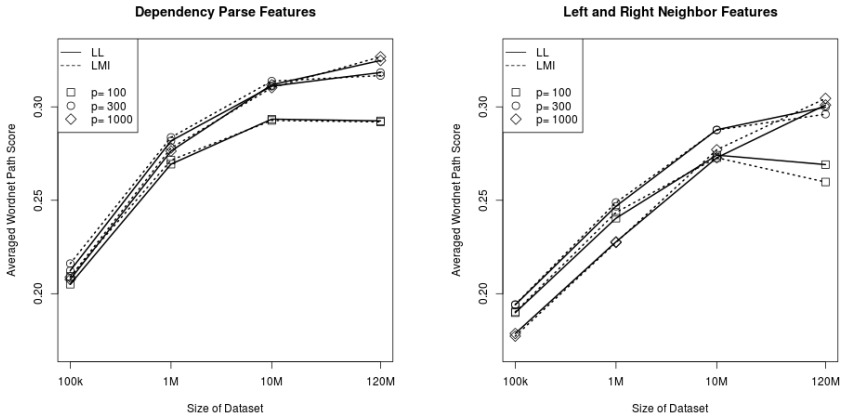

Figure 4: Corpus size vs. WordNet path similarity for different max. numbers of context features p, comparing LMI and LL measures, for two holing systems

As a general trend, larger corpora call for larger p – an effect that is especially pronounced for the token bigrams: whereas $p = 100$ produces the best results on the 1M sentence corpus, $p = 300$ excels for 10M sentences and the best scores overall for 120M sentences are obtained with $p = 1000$. However, differences between $p = 500$ and $p = 300$ respectively $p = 1000$ are small, so choosing p in the range of 500–1000 can be recommended for very large corpora. Comparing the holing systems, the dependency parse features result in much higher performance for small corpora, but do not outperform bigram features on large corpora by a great extent. This is consistent with a previous, similar evaluation by Curran (2004).

To support our qualitative observations, we list the DT entries for the LL measure and $p = 1000$ for the frequent noun “answer” and for the rather infrequent noun “tint” for different corpus sizes in Figures 5

Target: **answer**

r	100K	WP	1M	WP	10M	WP	120M	WP
1	question	1/5	solution	1	solution	1	explanation	1/3
2	reason	1/4	outcome	1/7	response	1	response	1
3	solution	1	explanation	1/3	explanation	1/3	reply	1
4	guy	1/11	way	1/6	question	1/5	solution	1
5	deal	1/4	excuse	1/6	reply	1	conclusion	1/4
6	decision	1/7	reaction	1/4	information	1/4	description	1/3
7	money	1/10	response	1	thing	1/3	question	1/5
8	plan	1/10	copy	1/6	rationale	1/12	information	1/4
9	story	1/4	thing	1/3	choice	1/6	remedy	1/10
10	goal	1/9	truth	1/3	reason	1/4	retort	1/3
Ø		0.25		0.41		0.46		0.48

Figure 5: DT entries for “answer” with WordNet path similarities (WP), comparing different corpus sizes from 100K sentences up to 120M sentences

Target: **tint**

rank	100K	WP	1M	WP	10M	WP	120M	WP
1	–	–	button	1/12	color	1/2	hue	1/5
2	–	–	clothing	1/13	hue	1/5	shade	1
3	–	–	meat	1/10	tone	1	color	1
4	–	–	suit	1/12	shade	1	tinge	1/2
5	–	–	arrow	1/12	tinge	1/2	shading	1/14
6	–	–	beer	1/16	hair	1/10	texture	1/4
7	–	–	berry	1/14	glow	1/7	tone	1
8	–	–	blazer	1/18	haze	1/11	coloration	1/3
9	–	–	box	1/10	light	1/4	palette	1/8
10	–	–	carpet	1/12	odor	1/5	patina	1/14
Ø		0		0.08		0.40		0.41

Figure 6: DT entries for “tint” with WordNet path similarities (WP), comparing different corpus sizes from 100K sentences up to 120M sentences

and 6. We provide the WordNet path similarities in fractional notation, where $1/x$ indicates a path length of $x - 1$ between target and similar term.

It is apparent that for a frequent word like “answer”, already a small collection can produce some reasonable top-ranked words, yet the list quickly degrades for 100K and 1M sentences. A typical effect for the largest of our corpora is illustrated with “retort”, which is about 20 times less frequent than “answer”, yet can collect enough significant contexts to enter its top 10 list. We frequently observed rather rare hyponyms and co-hyponyms of targets in the DTs computed from 120M sentences, which tremendously increases coverage for applications.

Looking at another example, the noun “tint” is too infrequent to receive any entry in the 100K sentence DT, and has a rather random

collection of words for 1M sentences, stemming from the shared adjective modifier “dark”. The larger collections produce quite suitable lists, again with a higher specialization for the 120M sentence corpus.

Next, we compare our similarity measure to similarities based on Lin’s and Curran’s measures, as introduced in Section 2.2. For both LL and LMI, we fixed $p = 1000$.

According to the results shown in Table 5, we can see that our method leads to much better results for frequent words.

In the evaluation of the 100k sentence dataset we observe that Lin’s measure beats all other measures for the frequent words. For this small corpus, our measure is the second best measure and Curran’s measure leads to the lowest scores. For infrequent nouns, our approach produces the best results for this dataset. For the 120M sentence dataset, Lin’s measure and our measure produce similar results, with our method being at slight advantage. Curran’s measure shows inferior performance. We can observe that all measures improve when based on larger data. It seems surprising that our comparably simple measure matches and outperforms, respectively, two well-established measures from the literature. We will spend the remainder of this section discussing possible reasons.

Since Lin’s measure was optimized on a much smaller corpus of about three million sentences using a different parser in Lin (1998),

Table 5: Wordnet Path Similarity for 1000 frequent and 1000 infrequent nouns, computed on 100K and 120M sentences comparing our measure to measures by Lin (1998) and Curran (2002)

corpus	Freq./	Top	Other methods		Our method	
size	infreq.	words	Lin	Curran	LL	LMI
100k	freq	top 10	**0.21322**	0.17779	0.19566	0.19645
100k	freq	top 5	**0.23295**	0.18031	0.20736	0.20798
100k	infreq	top 10	0.08186	0.09565	**0.12239**	0.12213
100k	infreq	top 5	0.10128	0.10164	**0.12759**	0.12683
120M	freq	top 10	0.27874	0.25429	0.28270	**0.28339**
120M	freq	top 5	0.31742	0.28355	0.32479	**0.32679**
120M	infreq	top 10	0.21480	0.17829	**0.22139**	0.21902
120M	infreq	top 5	0.24640	0.19490	0.25773	**0.25798**

it seems to be reasonable to assume that the factor regarding the frequency of the relation $f(relation(feature))$ (cf. Table 1) suppresses the influence of noise, but at the same time puts too much emphasis on frequent relations, which prevents a more fine-grained characterization of items by features. This is also confirmed by the results based on the 100k dataset. Our measure, on the other hand, increases in quality when more evidence (higher frequency) is available, which results in higher quality overall as collections are scaled up, and the p parameter on the number of characterizing features takes care of the noise.

Curran's measure was optimized on a collection larger than that in Lin's work, measuring about 300 million words (15 million sentences, Curran 2002), which is still about one order of magnitude smaller than our large corpus. Surprisingly, we could not confirm that Curran's measure performs better than Lin's measure (Curran, 2002).[12] This might be explained by the use of a different parser and different test words. Additionally, Curran uses a different evaluation method, as he compares his DT against entries from a combined set of entries taken from various thesauri, and only using a small number of nouns.

Wrapping up the DT evaluation, we can state that the most important factor for obtaining a high-quality DT is the amount of data. Comparing our proposal with existing measures, we feel that the effectiveness of semantic similarity measures on large corpora has been reconfirmed: on more data, simpler measures perform as well or even better than measures that were intended to give good results for small collections – an insight similar to that described in the seminal work of Banko and Brill (2001) for machine learning methods.

When using our measure, which is highly optimized for speed of computation, a suitable significance measure for ranking context features is required: measures that favor frequent items are preferable in our setup. Here, LMI and LL produced very similar scores, hence LMI is preferable because of its simpler, and thus more efficient, computation. There is no need to retain more than 500–1000 context features

[12] Following his Dice formula, it is not clear whether to take the intersection or the union of the features of two words. We tested different possibilities that, however, did not yield improvements. We decided to use the intersection, as it is unclear how to interpret the minimum function otherwise.

per term even for large corpora, which allows us to speed up the computation of the DT by a large degree. Equipped with this result, we can proceed to evaluate the effects of contextualization.

3.3 *Contextual similarity*

The contextualization evaluation was performed using the distributional thesaurus that was compiled using up to 120M sentences and using the LMI measure and $p = 1000$, as this combination showed the best performance in the previous section. The outcome for the contextualization is shown using the test set of the LexSub dataset, described in Section 3.1.

3.3.1 Evaluation methodology

For the evaluation of the LexSub dataset we used the out of ten (OOT) precision and OOT mode precision on the LexSub test set of 1710 sentences, as described in McCarthy and Navigli (2009). The OOT measure allows us to make up to 10 guesses, discarding further guesses. Both measures calculate how many substitutions have been detected within ten guesses over the complete subset. The difference is the "detection" of a correct match per entry. Whereas the OOT precision sums up the number of correct guesses divided by the number of possible answers, in the OOT mode precision evaluation the system is credited if the mode from the annotators (most frequent response(s)) is found within the system's 10 responses. We do not apply any special handling regarding multiwords (terms consisting of more than one word), which are not contained in our DT and are therefore always missed. For comparison, we use the results of the distributional thesaurus as a baseline to evaluate the contextualization. Note that our system does not yield duplicate entries, which are known to influence the OOT metric. We chose the OOT measure over the 'best' metric, since it better fits the metaphor of expanding text with several words.

As already mentioned in Section 2.3, we only use context features that contain another content word[13] and divide the weight by their corpus frequency. Furthermore, we use a threshold for the significance value of the LMI values of 40.0, and the most similar 30 terms from the

[13] Words with part-of-speech prefixes V, N, J, R.

DT entries as candidates for the contextual ranking. These parameters have been determined by optimizing OOT scores on the LexSub trial set.

3.3.2 Results

Since it can be expected that the contextualization algorithm is dependent on the number of context features for the target occurrence, we report scores for targets with at least two and at least three dependencies separately. In the LexSub test data, all targets have at least one, 49.2% of the targets have at least two and 26.0% have at least three dependencies. Furthermore, we also evaluated the results broken down into separate parts-of-speech of the target. The results for the OOT precision and the mode precision for both the entries of the distributional thesaurus (DT) and the contextualization (CT) are shown in Table 6.

Table 6: Results on the LexSub test dataset for global (DT) and contextualized (CT) similarities, per min number of dependencies to target

		Precision			Mode precision		
min. # dep.		1	2	3	1	2	3
POS	Alg.						
adjective	DT	32.81	33.64	35.02	43.56	43.53	42.86
adjective	CT	33.27	35.41	36.08	44.48	48.24	46.43
noun	DT	25.29	25.00	28.07	35.06	34.48	36.76
noun	CT	26.76	26.67	28.63	39.08	38.92	39.71
verb	DT	24.41	22.63	22.10	30.00	29.35	29.14
verb	CT	24.48	24.33	23.80	32.58	33.33	34.29
adverb	DT	28.85	26.75	29.88	41.43	34.38	66.67
adverb	CT	20.80	29.46	36.23	30.48	40.63	100.00
ALL	DT	**27.48**	25.10	25.72	37.19	33.39	33.77
ALL	CT	27.02	**26.84**	**27.14**	**37.35**	**37.75**	**38.41**

Inspecting the results for precision and mode precision without filtering entries regarding parts-of-speech (denoted as ALL), only marginal changes can be seen for entries having at least one dependency. But we observe substantial improvements for targets with more than one dependency: more than 1.6 points in precision and more than 4 points in mode precision.

The results regarding different part-of-speech tags of the target words follow a similar trend. For adjectives, nouns and verbs, the contextualization improves results throughout for all targets. Most notably, the largest relative improvements are observed for verbs, which is a notoriously difficult word class in computational semantics. For adverbs, contextualization hurts in cases where the adverb has fewer than two context features, but helps for targets with a minimum of two dependencies. Since there are merely seven instances where adverbs have at least three dependencies in the dataset, the high scores in mode precision are probably not representative.

Regarding performance on the original lexical substitution task (McCarthy and Navigli, 2009), we did not come close to the performance of the participating systems, which range between 32–50 precision points and 43–66 mode precision points (only taking into account systems without duplicate words in the result set). However, all participants used one or several lexical resources for generating substitution candidates, as well as a large number of features. Our system, on the other hand, merely requires a holing system – in this case based on a dependency parser – and a large amount of unlabeled text, as well as a very small number of contextual clues. Scores for a DT computed on the British National Corpus using Lin's measure as reported in McCarthy and Navigli (2009) are slightly higher than what we observe here, which we attribute to a different underlying background corpus.

3.4 *Two-dimensional representation for part-of-speech tagging*

In this section, we demonstrate how the notion of two-dimensional text can be used directly in NLP tasks using part-of-speech (POS) tagging as an example. While POS tagging is generally regarded as solved for languages and domains with sufficient amounts of training data, there are still challenges in domain adaptation, e.g. for user-generated content (Gimpel *et al.*, 2011) or for domain-specific texts (e.g. Biemann 2009 reports a 20% out-of-vocabulary (OOV) rate of news models on medical texts). The largest source of errors in POS assignment is observed for out-of-vocabulary words, i.e. words that were not contained in the training data and have to be classified according to context and surface features only. A sequence of OOV words can throw off the sequence classification algorithm, resulting in poor performance. For

classifiers that do not normalize over the whole sequence, this has been described as the label bias problem, cf. Lafferty *et al.* (2001).

Two-dimensional text provides a possibility to overcome the OOV problem by resorting to the most similar in-vocabulary word, when encountering a word unseen in training. For this, merely a list of in-vocabulary words has to be maintained. Presupposing an existing supervised POS tagger, the scheme is executed as follows.

Model training

1. Train the POS tagger on training text and construct the list of in-vocabulary words.
2. Compute a distributional thesaurus (DT) on a large background corpus.

POS tagging task

1. Determine the OOV words of the input text by checking the in-vocabulary word list.
2. For all OOV words, replace the OOV word by its most similar in-vocabulary word according to the DT.
3. Tag the altered text with the POS tagger, and project tags back to the original text.

For our experiments, we trained the well-known TreeTagger (Schmid, 1995) on the Penn Treebank (PTB, Marcus *et al.* 1993), following Collins (2002) by training on Sections 0–18 and testing on Sections 22–24.[14] The distributional thesaurus was induced on 120M sentences of English newswire, using a holing system based on word trigrams: the center word of each trigram served as the word, the two neighboring words (left and right together) served as the context. We retained the most similar 100 words per entry.

Figure 7 illustrates this method using an example: in the sentence “Renting out an unfurnished one-bedroom triplex in San Francisco”, the words “unfurnished”, “one-bedroom” and “triplex” are OOV words, not being part of the PTB training set. In the case of “one-bedroom” this might seem surprising, but the Penn Treebank consistently uses a spelling without the hyphen, resulting in two tokens

[14] We do not perform parameter optimization and therefore do not use Sections 19–21, which are normally used for development.

Figure 7: Illustrating the two-dimensional extension for POS tagging

"one bedroom". While the top-most similar words to "unfurnished" and "triplex" ("empty" and "duplex") are in-vocabulary words of our POS tagger, the most similar in-vocabulary word for "one-bedroom", "two-room", is the third most similar expansion according to our DT. Tagging the alternate sentence "Renting out an empty two-room duplex in San Francisco" results in correct assignment of POS tags, cf. Figure 7.

Evaluating the improvement over the whole test set, we improved the accuracy on the 3562 OOV words (the majority of them are verbs, nouns and adjectives) from 37.82% to 74.12%.[15] Overall, the accuracy of the tagger improved from 95.28% to 96.07%, only by altering the tagging strategy on the portion of 2.1% OOV words.

This overall performance is well below state-of-the-art POS tagging on this dataset (which is at 98.5%, Søgaard 2011), where successful approaches make heavy use of surface feature backoff, word clustering on background corpora, and advanced machine learning techniques. Our setup, however, illustrates how the metaphor of two-dimensional text can be used in the context of existing NLP software, while neither needing to alter the feature representation nor the learning algorithm for machine learning. The key, and its novelty with respect to word-space approaches, is that the DT is able to *generate* the most similar words, so that they can be used in lieu of words that impose difficulties for the software (i.e. OOV words for POS tagging). A comparable approach of expanding text representations with similar words from our process was successfully used by Miller *et al.* (2012) for state-of-the-art knowledge-based all-words word sense disambiguation.

[15] We enabled the heuristics of the TreeTagger (-hypen-heuristics, -ignore-prefix, -cap-heuristics) which improved the accuracy by 0.15% without any OOV replacement.

4 FUTURE WORK

There are a number of ways in which our framework for the metaphor of two-dimensional text can be filled and extended. In the remainder, we will briefly describe approaches that we intend to try in the future.

4.1 *Generalization of the holing system*

Experiments presented here used holing systems that extract context features for single words. While it is straightforward to extend it to pre-defined multi-word units, it would be promising to allow arbitrary, not necessarily contiguous sets of language elements, and determine their appropriateness by means of the similarity computation. The current framework also supports the computation of context feature similarities by exchanging the columns “language elements” and “context features” in the DT computation depicted in Figure 2, yet it still needs to be worked out how similarities of contexts could be used in the contextualization. Along these lines, a further generalization of the holing system is to use an arbitrary number of holes, which could e.g. allow us to detect similarities between active and passive constructions.

4.2 *Combination of signals for contextualization*

While we have only shown experiments using a single holing system at a time, it is possible to combine signals from several holing systems for contextualization, as well as signals from other semantic models such as topic models (cf. Thater *et al.*, 2011). Further, there is a large space of parameterization of the holing system with respect to the use of priors, the numerical transformation of word-context-significances to path probabilities, and the weighting of signals from different models.

4.3 *Other sampling methods for contextualization*

While we have demonstrated that a simple contextualization method as described in Section 2.3 is already able to achieve improvements of the lexical expansion quality, we would like to employ sampling methods that adjust path probabilities based on previous trials, like Metropolis-Hastings Sampling (Hastings, 1970), or dynamic programming approaches to compute the ranking of expansions efficiently (cf.

Viterbi, 1967; Lafferty *et al.*, 2001). In contrast to our simple method, these approaches normalize over the whole expanded sequence and perform expansions for all terms at the same time.

4.4 *Word sense induction clustering*

As the contextualization was described, the problem of word sense disambiguation is handled implicitly by down-ranking lexical expansions that refer to the wrong sense of the word in the context. It might be advantageous, however, to add word sense induction clustering on the DT entry (cf. Schütze, 1998; Widdows and Dorow, 2002; Biemann, 2010), and to perform the contextualization per cluster instead of per word to alleviate sparsity. Note that this per-entry clustering is different than the whole-vocabulary clustering proposed by Pereira *et al.* (1993) and others.

4.5 *Distinguishing expansions by patterns*

While word sense induction can distinguish similar words in the DT by sense, we need something else in order to obtain typed relations between a target and its potential expansions. One way of typing is to examine what patterns (e.g. is-a, part-of, antonyms) are common between target and expansion in our large corpus. These types would be useful for targeting certain types of expansions, e.g. excluding antonyms for lexical substitution. To keep the approach unsupervised and knowledge-free, we would like to find the patterns automatically in a co-clustering approach based on terms and patterns (Dhillon, 2001) rather than using pre-defined patterns (Hearst, 1992; Lin *et al.*, 2003).

4.6 *Machine learning on delexicalized features*

All the parameters and extensions to our core approach could play the role of features in a machine learning system, which e.g. could learn the weighting of different holing systems or different relations within the same holing system, the pattern type and so on. In this way, the lexical expansions can be tuned towards benefiting a given task at hand. The advantage of using these non-lexicalized features is that a single model can be learned for all expansions, as opposed to one model per language element type (i.e. one classifier per word). Features from the first-order and the second-order representation of

our distributional thesaurus have been employed for state-of-the-art lexical substitution in Szarvas *et al.* (2013).

5 CONCLUSION

In this article, we have introduced the new metaphor of two-dimensional text. This metaphor is rooted in structural linguistics, and expands the one-dimensional linear sequence of language elements in a second dimension of associative relations, especially with semantically similar language elements. We have provided a way of operationalizing semantic similarity by splitting syntagmatic observations into terms and context features, and representing them independent of the kind of syntagmatic observation. A scalable, parallelizable implementation of the computation of a distributional thesaurus was laid out in detail. Further, we provide a conceptually simple and efficient method to perform a contextualization of semantic similarity. Overall, our approach constitutes an unsupervised generative model for lexical expansion in context that implements the metaphor of two-dimensional text. In our experiments regarding the quality of distributional similarity, we demonstrated that our pruning method for DT computation is effective: using only the most n significant features per term greatly reduces processing time, and even improves the results. Further, we show that larger corpora lead to higher-quality distributional thesauri, and that we can effectively compute them without relying on lossy compression techniques. Our measure excels over two competitive measures in the literature on very large collections. We have presented a generic method of contextualizing distributional information, which selects entries from the DT entry of the expansion target, and ranks them with respect to their context compatibility. Evaluating our method on the lexical substitution task (McCarthy and Navigli, 2009), we were able to show consistent improvements across all parts of speech, especially for expansion targets with many informing contextual elements. Further, we demonstrated how the two-dimensional expansion can improve part-of-speech tagging without the need to re-train or otherwise alter the tagger. Finally, we laid out a plethora of possible extensions for improving our implementation of the two-dimensional text metaphor. This work is merely a first step towards creating a new, entirely data-driven *model* for computational seman-

tics, as opposed to mere feature-based machine learning or knowledge-intensive approaches.

ACKNOWLEDGEMENTS

This work has been funded by the Hessian research excellence program *Landes-Offensive zur Entwicklung Wissenschaftlich-Ökonomischer Exzellenz (LOEWE)* as part of the *Digital Humanities* research center. We also thank our partners Alfio Gliozzo, Michael Glass and Bonaventura Coppola at IBM Research for contributing to the open source implementation, and for discussions.

REFERENCES

Michele BANKO and Eric BRILL (2001), Scaling to very very large corpora for natural language disambiguation, in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pp. 26–33, Association for Computational Linguistics, Stroudsburg, PA, USA, http://dx.doi.org/10.3115/1073012.1073017.

Marco BARONI and Alessandro LENCI (2010), Distributional memory: A general framework for corpus-based semantics, *Computational Linguistics*, 36(4):673–721, ISSN 0891-2017, http://dx.doi.org/10.1162/coli_a_00016.

Marco BARONI and Roberto ZAMPARELLI (2010), Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pp. 1183–1193, Cambridge, Massachusetts, http://dl.acm.org/citation.cfm?id=1870658.1870773.

Chris BIEMANN (2009), Unsupervised Part-of-Speech Tagging in the Large, *Research on Language and Computation*, 7(2–4):101–135, ISSN 1570-7075, http://dx.doi.org/10.1007/s11168-010-9067-9.

Chris BIEMANN (2010), Co-occurrence cluster features for lexical substitutions in context, in *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, TextGraphs-5, pp. 55–59, ISBN 978-1-932432-77-0, http://dl.acm.org/citation.cfm?id=1870490.1870499.

Chris BIEMANN and Eugenie GIESBRECHT (2011), Distributional Semantics and Compositionality 2011: Shared Task Description and Results, in *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pp. 21–28, Association for Computational Linguistics, Portland, Oregon, USA, http://www.aclweb.org/anthology/W11-1304.

Chris BIEMANN, Uwe QUASTHOFF, Gerhard HEYER, and Florian HOLZ (2008), ASV Toolbox: a Modular Collection of Language Exploration Tools, in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco, http://www.lrec-conf.org/proceedings/lrec2008/summaries/447.html.

Chris BIEMANN, Stefanie ROOS, and Karsten WEIHE (2012), Quantifying Semantics Using Complex Network Analysis, in *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, Mumbai, India, http://aclweb.org/anthology/C/C12/C12-1017.pdf.

David M. BLEI, Andrew Y. NG, and Michael I. JORDAN (2003), Latent Dirichlet allocation, *Journal of Machine Learning Research*, 3:993–1022, ISSN 1532-4435, http://dl.acm.org/citation.cfm?id=944919.944937.

Stefan BORDAG (2008), A comparison of co-occurrence and similarity measures as simulations of context, in *CICLing'08 Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 52–63, Haifa, Israel, http://dl.acm.org/citation.cfm?id=1787578.1787584.

Jordan BOYD-GRABER and David M. BLEI (2008), Syntactic Topic Models, in *Neural Information Processing Systems*, Vancouver, British Columbia, http://www.cs.princeton.edu/~blei/papers/Boyd-GraberBlei2009.pdf.

Kenneth Ward CHURCH and Patrick HANKS (1990), Word association norms, mutual information, and lexicography, *Computational Linguistics*, 16(1):22–29, ISSN 0891-2017, http://dl.acm.org/citation.cfm?id=89086.89095.

Michael COLLINS (2002), Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms, in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing – Volume 10*, EMNLP '02, pp. 1–8, Association for Computational Linguistics, Stroudsburg, PA, USA, http://dx.doi.org/10.3115/1118693.1118694.

James R. CURRAN (2002), Ensemble methods for automatic thesaurus extraction, in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing – Volume 10*, EMNLP '02, pp. 222–229, http://dx.doi.org/10.3115/1118693.1118722.

James R. CURRAN (2004), *From Distributional to Semantic Similarity*, University of Edinburgh, http://books.google.de/books?id=2iDbSAAACAAJ.

Ferdinand DE SAUSSURE (1916), *Cours de linguistique générale*, Payot, Paris, http://www.bibsonomy.org/bibtex/2e68b895a274b9569189c5ae98db84603/jntr.

Ferdinand DE SAUSSURE (1959), *Course in general linguistics*, Language (Philosophical Library), Philosophical Library, http://books.google.de/books?id=FSpZAAAAMAAJ.

Jeffrey DEAN and Sanjay GHEMAWAT (2004), MapReduce: Simplified Data Processing on Large Clusters, in *Proceedings of Operating Systems, Desing & Implementation (OSDI) '04*, pp. 137–150, San Francisco, CA, USA, http://doi.acm.org/10.1145/1327452.1327492.

Scott DEERWESTER, Susan T. DUMAIS, George W. FURNAS, Thomas K. LANDAUER, and Richard HARSHMAN (1990), Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, 41(6):391–407, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.8490.

Inderjit S. DHILLON (2001), Co-clustering documents and words using bipartite spectral graph partitioning, in *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pp. 269–274, ACM, New York, NY, USA, ISBN 1-58113-391-X, http://doi.acm.org/10.1145/502512.502550.

Ted DUNNING (1993), Accurate methods for the statistics of surprise and coincidence, *Computational Linguistics*, 19(1):61–74, ISSN 0891-2017, http://dl.acm.org/citation.cfm?id=972450.972454.

Katrin ERK and Sebastian PADÓ (2008), A structured vector space model for word meaning in context, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pp. 897–906, Honolulu, Hawaii, http://dl.acm.org/citation.cfm?id=1613715.1613831.

Stefan EVERT (2005), *The Statistics of Word Cooccurrences: Word Pairs and Collocations.*, Ph.D. thesis, Institut für Maschinelle Sprachverarbeitung, University of Stuttgart, http://elib.uni-stuttgart.de/opus/volltexte/2005/2371/.

Eugenie GIESBRECHT (2009), In Search of Semantic Compositionality in Vector Spaces, in *Proceedings of the 17th International Conference on Conceptual Structures: Conceptual Structures: Leveraging Semantic Technologies*, ICCS '09, pp. 173–184, Springer-Verlag, Berlin, Heidelberg, ISBN 978-3-642-03078-9, http://dx.doi.org/10.1007/978-3-642-03079-6_14.

Kevin GIMPEL, Nathan SCHNEIDER, Brendan O'CONNOR, Dipanjan DAS, Daniel MILLS, Jacob EISENSTEIN, Michael HEILMAN, Dani YOGATAMA, Jeffrey FLANIGAN, and Noah A. SMITH (2011), Part-of-speech tagging for Twitter: annotation, features, and experiments, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers – Volume 2*, HLT '11, pp. 42–47, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-932432-88-6, http://dl.acm.org/citation.cfm?id=2002736.2002747.

Gene H. GOLUB and William M. KAHAN (1965), Calculating the singular values and pseudo-inverse of a matrix, *Journal of the Society for Industrial and Applied Mathematics: Series B: Numerical Analysis*, 2:205–224, http://www.citeulike.org/user/rabio/article/2342309.

James GORMAN and James R. CURRAN (2006), Scaling Distributional Similarity to Large Corpora, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 361–368, Association for Computational Linguistics, Sydney, Australia, http://www.aclweb.org/anthology/P06-1046.

Amit GOYAL, Hal DAUMÉ III, and Graham CORMODE (2012), Sketch Algorithms for Estimating Point Queries in NLP, in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1093–1103, Association for Computational Linguistics, http://www.aclweb.org/anthology/D12-1100.

Emiliano GUEVARA (2011), Computing semantic compositionality in distributional semantics, in *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pp. 135–144, Association for Computational Linguistics, Stroudsburg, PA, USA, http://dl.acm.org/citation.cfm?id=2002669.2002684.

Zellig S. HARRIS (1951), *Methods in Structural Linguistics*, University of Chicago Press, Chicago, http://archive.org/details/structurallingui00harr.

W. Keith HASTINGS (1970), Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, 57(1):97–109, ISSN 1464-3510, doi:10.1093/biomet/57.1.97, http://dx.doi.org/10.1093/biomet/57.1.97.

Marti A. HEARST (1992), Automatic acquisition of hyponyms from large text corpora, in *Proceedings of the 14th conference on Computational linguistics – Volume 2*, COLING '92, pp. 539–545, http://dx.doi.org/10.3115/992133.992154.

Enrique HENESTROZA ANGUIANO and Pascal DENIS (2011), FreDist: Automatic construction of distributional thesauri for French, in *TALN – 18ème conférence sur le traitement automatique des langues naturelles*, pp. 119–124, Montpellier, France, France, http://hal.inria.fr/hal-00602004.

Thomas HOFMANN (1999), Probabilistic latent semantic indexing, in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pp. 50–57, ACM, New York, NY, USA, ISBN 1-58113-096-1, http://doi.acm.org/10.1145/312624.312649.

Adam KILGARRIFF, Pavel RYCHLY, Pavel SMRZ, and David TUGWELL (2004), The Sketch Engine, in *Proceedings of EURALEX*, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.180.7984.

Walter KINTSCH (2001), Predication, *Cognitive Science*, 25(2):173–202, ISSN 1551-6709, http://dx.doi.org/10.1207/s15516709cog2502_1.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira (2001), Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, in *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1-55860-778-1, http://dl.acm.org/citation.cfm?id=645530.655813.

Lillian Lee (1999), Measures of distributional similarity, in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pp. 25–32, College Park, Maryland, ISBN 1-55860-609-3, http://dx.doi.org/10.3115/1034678.1034693.

Michael Lesk (1986), Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pp. 24–26, ACM, New York, NY, USA, ISBN 0-89791-224-1, http://doi.acm.org/10.1145/318723.318728.

Dekang Lin (1998), Automatic retrieval and clustering of similar words, in *Proceedings of the 17th International Conference on Computational Linguistics – Volume 2*, COLING '98, pp. 768–774, http://dx.doi.org/10.3115/980432.980696.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou (2003), Identifying synonyms among distributionally similar words, in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pp. 1492–1493, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, http://dl.acm.org/citation.cfm?id=1630659.1630908.

Jimmy Lin and Chris Dyer (2010), *Data-Intensive Text Processing with MapReduce*, Morgan & Claypool Publishers, San Rafael, CA, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.169.6896.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini (1993), Building a large annotated corpus of English: the Penn Treebank, *Computational Linguistics*, 19(2):313–330, ISSN 0891-2017, http://dl.acm.org/citation.cfm?id=972470.972475.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning (2006), Generating typed dependency parses from phrase structure parses, in *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2006*, Genova, Italy, http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.229.775.

Diana McCarthy and Roberto Navigli (2009), The English lexical substitution task., *Language Resources and Evaluation*, 43(2):139–159, http://dblp.uni-trier.de/db/journals/lre/lre43.html#McCarthyN09.

George A. Miller and Walter G. Charles (1991), Contextual correlates of semantic similarity, *Language and Cognitive Processes*, 6(1):1–28, http://dx.doi.org/10.1080/01690969108406936.

Tristan MILLER, Chris BIEMANN, Torsten ZESCH, and Iryna GUREVYCH (2012), Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation, in *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pp. 1781–1796, Mumbai, India, http://aclweb.org/anthology/C/C12/C12-1109.pdf.

Jeff MITCHELL and Mirella LAPATA (2008), Vector-based Models of Semantic Composition, in *Proceedings of ACL-08: HLT*, pp. 236–244, Columbus, Ohio, www.aclweb.org/anthology/P08-1028.pdf.

Sebastian PADÓ and Mirella LAPATA (2007), Dependency-based construction of semantic space models, *Computational Linguistics*, 33(2):161–199, http://citeseer.uark.edu:8080/citeseerx/viewdoc/summary?doi=10.1.1.86.2026.

Sebastian PADÓ and Yves PEIRSMAN, editors (2011), *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, Association for Computational Linguistics, Edinburgh, UK, http://www.aclweb.org/anthology/W11-25.

Robert PARKER, David GRAFF, Junbo KONG, Ke CHEN, and Kazuaki MAEDA (2011), *English Gigaword Fifth Edition*, Linguistic Data Consortium, Philadelphia, http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2011T07.

Ted PEDERSEN, Siddharth PATWARDHAN, and Jason MICHELIZZI (2004), WordNet::Similarity: measuring the relatedness of concepts, in *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL – Demonstrations '04, pp. 38–41, http://dl.acm.org/citation.cfm?id=1614025.1614037.

Fernando PEREIRA, Naftali TISHBY, and Lillian LEE (1993), Distributional clustering of English words, in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, pp. 183–190, Association for Computational Linguistics, Stroudsburg, PA, USA, http://dx.doi.org/10.3115/981574.981598.

Reinhard RAPP (2003), Word sense discovery based on sense descriptor dissimilarity, in *Proceedings of the Ninth Machine Translation Summit*, pp. 315–322, http://www.citeulike.org/user/briordan/article/2911465.

Matthias RICHTER, Uwe QUASTHOFF, Erla HALLSTEINSDÓTTIR, and Chris BIEMANN (2006), Exploiting the Leipzig Corpora Collection, in *Proceesings of the IS-LTC 2006*, Ljubljana, Slovenia, http://nl.ijs.si/is-ltc06/proc/13_Richter.pdf.

Herbert RUBENSTEIN and John B. GOODENOUGH (1965), Contextual correlates of synonymy, *Communications of the ACM*, 8(10):627–633, ISSN 0001-0782, http://doi.acm.org/10.1145/365628.365657.

Gerda RUGE (1992), Experiments on linguistically-based term associations, *Information Processing & Management*, 28(3):317 – 332, ISSN 0306-4573, http://www.sciencedirect.com/science/article/pii/030645739290078E.

Magnus SAHLGREN (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, http://soda.swedish-ict.se/437/.

Helmut SCHMID (1995), Improvements in Part-of-Speech Tagging with an Application to German, in *Proceedings of the ACL SIGDAT-Workshop*, pp. 47–50, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.2255.

Hinrich SCHÜTZE (1993), Word Space, in *Advances in Neural Information Processing Systems 5*, pp. 895–902, Morgan Kaufmann, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.8856.

Hinrich SCHÜTZE (1998), Automatic word sense discrimination, *Computational Linguistics*, 24(1):97–123, ISSN 0891-2017, http://dl.acm.org/citation.cfm?id=972719.972724.

Anders SØGAARD (2011), Semisupervised condensed nearest neighbor for part-of-speech tagging, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers – Volume 2*, HLT '11, pp. 48–52, Portland, Oregon, ISBN 978-1-932432-88-6, http://dl.acm.org/citation.cfm?id=2002736.2002748.

György SZARVAS, Chris BIEMANN, and Iryna GUREVYCH (2013), Supervised All-Words Lexical Substitution using Delexicalized Features, in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-2013)*, Atlanta, GA, USA, http://aclweb.org/anthology/N/N13/N13-1133.pdf.

Ming TAN, Wenli ZHOU, Lei ZHENG, and Shaojun WANG (2012), A scalable distributed syntactic, semantic, and lexical language model, *Computational Linguistics*, 38(3):631–671, ISSN 0891-2017, http://dx.doi.org/10.1162/COLI_a_00107.

Stefan THATER, Hagen FÜRSTENAU, and Manfred PINKAL (2011), Word Meaning in Context: A Simple and Effective Vector Model, in *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 1134–1143, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, http://www.aclweb.org/anthology/I11-1127.

Peter D. TURNEY and Michael L. LITTMAN (2005), Corpus-based Learning of Analogies and Semantic Relations, *Machine Learning*, 60(1-3):251–278, ISSN 0885-6125, http://dx.doi.org/10.1007/s10994-005-0913-1.

Peter D. TURNEY and Patrick PANTEL (2010), From frequency to meaning: vector space models of semantics, *Journal of Artificial Intelligence Research*,

37(1):141–188, ISSN 1076-9757,
http://dl.acm.org/citation.cfm?id=1861751.1861756.

Andrew J. VITERBI (1967), Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory*, 13(2):260–269, ISSN 0018-9448, doi:10.1109/TIT.1967.1054010, http://dx.doi.org/10.1109/TIT.1967.1054010.

Julie WEEDS (2003), *Measures and Applications of Lexical Distributional Similarity*, Ph.D. thesis, Department of Informatics, University of Sussex, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.538.

Dominic WIDDOWS and Beate DOROW (2002), A graph model for unsupervised lexical acquisition, in *Proceedings of the 19th International Conference on Computational Linguistics – Volume 1*, COLING '02, pp. 1–7, http://dx.doi.org/10.3115/1072228.1072342.