A Hybrid Distributional and Knowledge-based Model of Lexical Semantics

Nikolaos Aletras
Department of Computer Science, University College London, Gower Street, London WC1E 6BT United Kingdom
nikos.aletras@gmail.com

Mark Stevenson
Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, Sheffield S1 4DP United Kingdom
mark.stevenson@sheffield.ac.uk

Abstract

A range of approaches to the representation of lexical semantics have been explored within Computational Linguistics. Two of the most popular are distributional and knowledge-based models. This paper proposes hybrid models of lexical semantics that combine the advantages of these two approaches. Our models provide robust representations of synonymous words derived from WordNet. We also make use of WordNet’s hierarchy to refine the synset vectors. The models are evaluated on two widely explored tasks involving lexical semantics: lexical similarity and Word Sense Disambiguation. The hybrid models are found to perform better than standard distributional models and have the additional benefit of modelling polysemy.

1 Introduction

The representation of lexical semantics is a core problem in Computational Linguistics and a variety of approaches have been developed. Two of the most widely explored have been knowledge-based and distributional semantics.

Knowledge-based approaches make use of some external information source which defines the set of possible meanings for each lexical item. The most widely used information source is WordNet (Fellbaum, 1998), although other resources, such as Machine Readable Dictionaries, thesauri and ontologies have also been used (seeNavigli (2009)).

One advantage of these resources is that they represent the various possible meanings of lexical items which makes it straightforward to identify ones that are ambiguous. For example, these resources would include multiple meanings for the word ball including the ‘event’ and ‘sports equipment’ senses. However, the fact that there are multiple meanings associated with ambiguous lexical items can also be problematic since it may not be straightforward to identify which one is being used for an instance of an ambiguous word in text. This issue has lead to significant exploration of the problem of Word Sense Disambiguation (Ide and Véronis, 1998; Navigli, 2009).

More recently distributional semantics has become a popular approach to representing lexical semantics (Turney and Pantel, 2010; Erk, 2012). These approaches are based on the premise that the semantics of lexical items can be modelled by their context (Firth, 1957; Harris, 1985). Distributional semantic models have the advantages of being robust and straightforward to create from unannotated corpora. However, problems can arise when they are used to represent the semantics of polysemous words. Distributional semantic models are generally constructed by examining the context of lexical items in unannotated corpora. But for ambiguous words, like ball, it is not clear if a particular instance of the word in a corpus refers to the ‘event’, ‘sports equipment’ or another sense which can lead to the distributional semantic model becoming a mixture of different meanings without representing any of the meanings individually.

This paper proposes models that merge elements of distributional and knowledge-based approaches to lexical semantics and combines advantages of both techniques. A standard distributional semantic model
is created from an unannotated corpus and then refined using WordNet. The resulting models can be viewed as enhanced distributional models that have been refined using the information from WordNet to reduce the problems caused by ambiguous terms when models are created. Alternatively, it can be used as a version of the WordNet hierarchy in which distributional semantic models are attached to synsets. Thereby creating a version of WordNet for which the appropriate synsets can be identified more easily for ambiguous lexical items that occur in text.

We evaluate our models on two standard tasks: lexical similarity and word sense disambiguation. Results show that the proposed hybrid models perform consistently better than traditional distributional semantic models.

The reminder of the paper is organised as follows. Section 2 describes our hybrid models which combine information from WordNet and a standard distributional semantic model. These models are augmented using Latent Semantic Analysis and Canonical Correlation Analysis. Sections 3 and 4 describe evaluation of the models on the word similarity and word sense disambiguation tasks. Related work is presented in Section 5 and conclusions in Section 6.

2 Semantic Models

First, we consider a standard distributional semantic space to represent words as vectors (Section 2.1). Then, we make use of the WordNet’s clusters of synonyms and hierarchy in combination with the standard distributional space to build hybrid models (Section 2.2) which are augmented using Latent Semantic Analysis (Section 2.3) and Canonical Correlation Analysis (Section 2.4).

2.1 Distributional Model

We consider a semantic space $D$, as a word by context feature matrix, $L \times C$. Vector representations consist of context features $C$ in a reference corpus. We made use of pre-computed publicly available vectors\footnote{http://clic.cimec.unitn.it/composes/semantic-vectors.html} optimised for word similarity tasks (Baroni et al., 2014). Word co-occurrence counts are extracted using a symmetric window of two words over a corpus of 2.8 billion tokens obtained by concatenating ukWaC, the English Wikipedia and the British National Corpus. Vectors are weighted using positive Pointwise Mutual Information and the set of context features consists of the top 300K most frequent words in the corpus.

2.2 Hybrid Models

2.2.1 Synset Distributional Model

We assume that making use of information about the structure of WordNet can reduce noise introduced in vectors of $D$ due to polysemy. We make use of all noun and verb synsets (excluding numbers and compounds) that contain at least one of the words in $L$ to create a vector-based synset representation, $H$. Where $H$ is a synset by context feature matrix, i.e. $S \times C$. Each synset vector is generated by computing the centroid of its lemma vectors in $S$ (i.e. the sum of the lemma’s vectors normalised by the number of the lemmas in the synset). For example, the vector of the synset car.n.01 is computed as the centroid of its lemma vectors, i.e. car, auto, automobile, machine and motorcar (see Figure 1).

2.2.2 Synset Rank Model

The Synset Distributional Model provides a vector representation for each synset in WordNet which is created using information about which lemmas share synset membership. An advantage of this approach is that vectors from multiple lemmas are combined to form the synset representation. However, a disadvantage is that many of these lemmas are polysemous and their vectors represent multiple senses, not just the one that is relevant to the synset. For example, in WordNet the lemma machine has several possible meanings, only one of which is a member of the synset car.n.01.

WordNet also contains information about the relations between synsets, in the form of the synset hierarchy, which can be exploited to re-weight the importance of context features for particular synsets. We employ a graph-based algorithm that makes use of the WordNet is-a hierarchy. The intuition behind this approach is that context features that are relevant to a given synset are likely to be shared by its neighbours in the hierarchy while those that are not relevant (i.e. have been introduced via an irrelevant sense of a synset member) will not be. The graph-based algorithm increases the weight of context features
Figure 1: In the Synset Distributional Model the vector representing a synset (white box) is computed as the centroid of its lemma vectors (grey boxes)

that synsets share with neighbours and reduces those that are not shared.

PageRank (Page et al., 1999) is a graph-based algorithm for identifying important nodes in a graph that has been applied to a range of NLP tasks including word sense disambiguation (Agirre and Soroa, 2009) and keyword extraction (Mihalcea and Tarau, 2004).

Let $G = (V, E)$ be a graph with a set of vertices, $V$, denoting synsets and a set of edges, $E$, denoting links between synsets in the WordNet hierarchy. The PageRank score ($Pr$) over $G$ for a synset ($V_i$) can be computed by the following equation:

$$Pr(V_i) = d \cdot \sum_{V_j \in I(V_i)} \frac{1}{O(V_j)} Pr(V_j) + (1 - d)v$$

where $I(V_i)$ denotes the in-degree of the vertex $V_i$ and $O(V_j)$ is the out-degree of vertex $V_j$. $d$ is the damping factor which is set to the default value of $d = 0.85$ (Page et al., 1999). In standard PageRank all elements of the vector $v$ are the same, $\frac{1}{N}$ where $N$ is the number of nodes in the graph.

Personalised PageRank (PPR) (Haveliwala et al., 2003) is a variant of the PageRank algorithm in which extra importance is assigned to certain vertices in the graph. This is achieved by adjusting the values of the vector $v$ in equation 1 to prefer certain nodes. The values in $v$ effectively initialises the graph and assigning high values to nodes in $v$ makes them more likely to be assigned a high PPR score.

For each context feature $c$ in $C$ if $c \in LM$ where $LM$ contains all the lemma names of synsets in $S$, we apply PPR to assign importance to synsets. The score of each synset $S_c$ in the personalisation vector $v$, is set to $\frac{1}{|S_c|}$ where $|S_c|$ is the number of synsets that context feature $i$ belongs. The personalisation value of all the other synsets is set to 0.

We apply PPR over WordNet for each context feature using UKB (Agirre et al., 2009) and obtain weights for each synset-context feature pair resulting to a new semantic space $H_p$, $S \times C$, where vector elements are weighted by PageRank values. Figure 2 shows how the synset scores are computed by applying PPR over WordNet given the context feature $car$. Note that we use the context features of the distributional model $D$.

### 2.3 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Deerwester et al., 1990; Landauer and Dumais, 1997) has been used to reduce the dimensionality of semantic spaces leading to improved performance. LSA applies Singular Value Decomposition (SVD) to a matrix $X$, $W \times C$, which represents a distributional semantic space. This is a form of factor analysis where $X$ is decomposed into three other matrices:

$$X = U \Sigma V^T$$

where $U$ is a $W \times W$ matrix of row vectors where its columns are eigenvectors of $XX^T$, $\Sigma$ is a diagonal $W \times C$ matrix containing the singular values and $V$ is a $C \times C$ matrix of context feature vectors where its columns are eigenvectors of $X^T X$. The multiplication of the three component matrices results in the original matrix, $X$. Any matrix can be decomposed perfectly if the number of singular values is no smaller than the smallest dimension of $X$. When
fewer singular values are used then the matrix product is an approximation of the original matrix. LSA reduces the dimensionality of the SVD by deleting coefficients in the diagonal matrix \( \Sigma \) starting with the smallest. The approximation of matrix \( X \) retaining the \( K \) largest singular values, \( \tilde{X} \), is then given by:

\[
\tilde{X} \approx U_K \Sigma_K V_K^T
\]

(3)

where \( U_K \) is a \( W \times K \) matrix of word vectors, \( \Sigma_K \) is a \( K \times K \) diagonal matrix with singular values and \( V_K \) is a \( K \times C \) matrix of context feature vectors.

We apply LSA on the Synset Distributional Model, \( H \) and the Synset Rank model, \( H_p \) to obtained the reduced semantic spaces \( \tilde{H} \) and \( \tilde{H}_p \) respectively.

### 2.4 Joint Representation using CCA

Recent work has demonstrated that distributional models can benefit from combining alternative views of data (see Section 5). \( H \) and \( H_p \) provide two different views of the synsets and we incorporate evidence from both to learn a joint representation using Canonical Correlation Analysis (CCA) (Hardoon et al., 2004). Given two multidimensional variables \( x \) and \( y \), CCA finds two projection vectors by maximising the correlations of the variables onto these projections. The function to be maximised is:

\[
\rho = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}}
\]

(4)

The dimensionality of the projection vectors is lower or equal to the dimensionality of the original variables.

The computation of CCA directly over \( H \) and \( H_p \) is computationally infeasible because of their high dimensionality (300K). We apply CCA over the reduced spaces learned using LSA, \( \tilde{H} \) and \( \tilde{H}_p \) to obtain two joint semantic spaces following a similar approach to Faruqui and Dyer (2014). These are the spaces \( H^* \), resulting from the projection of the Synset Distributional Model \( \tilde{H} \), and \( H^*_p \), resulting from the projection of the Synset Rank Model \( \tilde{H}_p \).

### 3 Word Similarity

#### 3.1 Computing Similarity

Since hybrid models represent words as synset vectors, similarity between two words can be computed following two ways. First, we compute similarity between two words as the maximum of their pairwise synset similarity. On the other hand, similarity can be computed as the average pairwise synset similarity using the synsets that the two words belong. Similarity is computed as the cosine of the angle between word or synset vectors.

#### 3.2 Data

We make use of six standard data sets that have been widely used for evaluating lexical similarity and relat-
edness. First, we make use of WS-353 (Finkelstein et al., 2001) which contains 353 pairs of words annotated by humans. Furthermore, we make use of the similarity (WS-Sim) and relatedness (WS-Rel) pairs of words created by Agirre et al. (2009) from the original WS-353 data set.

We also made use of the RG (Rubenstein and Goodenough, 1965) and MC (Miller and Charles, 1991) data sets which contain 65 and 30 pairs of nouns respectively. Finally, we make use of the larger MEN data set (Bruni et al., 2012) which contains 3,000 pairs of words that has been used as image tags. Annotations are obtained using crowdsourcing.

### 3.3 Model Parameters

The parameters we need to tune are the number of the top components in LSA spaces, $\tilde{H}$ and $\tilde{H}_p$, and CCA spaces, $H^*$ and $H_p^*$. For the LSA spaces, we tune the number of the top $k$ components in RG. We set $k \in \{50, 100, ..., 1000\}$ and select the value that maximises performance which is $k = 700$ for $\tilde{H}$ and $k = 650$ for $\tilde{H}_p$. For the joint spaces learned using CCA, we also tune the number of the top $l$ correlated features in RG. We set $l \in \{10, 20, ..., 650\}$ and select the value that maximises performance which is $l = 250$ for $H^*$ and $l = 40$ for $H_p^*$.

### 3.4 Evaluation Metric

Performance is measured as the correlation between the similarity scores returned by each proposed method and the human judgements. This is the standard approach to evaluate word and text similarity tasks, e.g. (Budanitsky and Hirst, 2001; Agirre et al., 2009; Agirre et al., 2012). Our experiments use Spearman’s correlation coefficient.

### 3.5 Results

Table 1 shows the Spearman’s correlation of similarity scores generated by each model and human judgements of similarity across various data sets by taking the maximum pairwise similarity score of two words’ synsets. The first row of the table shows the results obtained by the word distributional model of Baroni et al. (2014). The full hybrid models $H$ and $H_p$ perform consistently worse than the original distributional model $D$ across data sets. The main reason is that a large number of synsets contain only one lemma name which might be polysemous. For example, the only lemma name of the synsets ‘ball.n.01’ ('round object that is hit or thrown or

<table>
<thead>
<tr>
<th>Model</th>
<th>WS-353</th>
<th>WS-Sim</th>
<th>WS-Rel</th>
<th>RG</th>
<th>MC</th>
<th>MEN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Distributional Model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D$</td>
<td>0.62</td>
<td>0.70</td>
<td>0.59</td>
<td>0.79</td>
<td>0.72</td>
<td>0.72</td>
</tr>
<tr>
<td><strong>Hybrid Models - Full</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td>0.49</td>
<td>0.60</td>
<td>0.36</td>
<td>0.69</td>
<td>0.64</td>
<td>0.58</td>
</tr>
<tr>
<td>$H_p$</td>
<td>0.58</td>
<td>0.67</td>
<td>0.49</td>
<td>0.82</td>
<td><strong>0.86</strong></td>
<td>0.63</td>
</tr>
<tr>
<td><strong>Hybrid Models - LSA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tilde{H}$</td>
<td>0.55</td>
<td>0.69</td>
<td>0.42</td>
<td>0.71</td>
<td>0.71</td>
<td>0.54</td>
</tr>
<tr>
<td>$\tilde{H}_p$</td>
<td>0.58</td>
<td>0.68</td>
<td>0.46</td>
<td>0.85</td>
<td><strong>0.86</strong></td>
<td>0.55</td>
</tr>
<tr>
<td><strong>Hybrid Models - CCA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H^*$</td>
<td>0.67</td>
<td><strong>0.76</strong></td>
<td>0.57</td>
<td>0.81</td>
<td>0.79</td>
<td><strong>0.72</strong></td>
</tr>
<tr>
<td>$H_p^*$</td>
<td>0.52</td>
<td>0.62</td>
<td>0.41</td>
<td><strong>0.86</strong></td>
<td>0.80</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 1: Spearman’s correlation on various data sets. Maximum similarity between pairs of synsets.
Table 2: Spearman’s correlation on various data sets. Average pairwise similarity between pairs of synsets.

<table>
<thead>
<tr>
<th>Average</th>
<th>Model</th>
<th>WS-353</th>
<th>WS-Sim</th>
<th>WS-Rel</th>
<th>RG</th>
<th>MC</th>
<th>MEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributional Model</td>
<td>D</td>
<td>0.62</td>
<td>0.70</td>
<td>0.59</td>
<td>0.79</td>
<td>0.72</td>
<td>0.72</td>
</tr>
<tr>
<td>Hybrid Models - Full</td>
<td>H</td>
<td>0.61</td>
<td>0.71</td>
<td>0.52</td>
<td>0.72</td>
<td>0.65</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>H_p</td>
<td>0.65</td>
<td>0.73</td>
<td>0.56</td>
<td>0.79</td>
<td>0.81</td>
<td>0.58</td>
</tr>
<tr>
<td>Hybrid Models - LSA</td>
<td>H</td>
<td>0.59</td>
<td>0.70</td>
<td>0.48</td>
<td>0.68</td>
<td>0.68</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>H_p</td>
<td>0.65</td>
<td>0.73</td>
<td>0.56</td>
<td>0.81</td>
<td>0.86</td>
<td>0.58</td>
</tr>
<tr>
<td>Hybrid Models - CCA</td>
<td>H</td>
<td>0.70</td>
<td>0.77</td>
<td>0.64</td>
<td>0.78</td>
<td>0.84</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>H_p</td>
<td>0.61</td>
<td>0.69</td>
<td>0.52</td>
<td>0.72</td>
<td>0.76</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Note that Baroni et al. (2014) found that applying SVD to $D$ did not improve performance over using the full space.

4 Word Sense Disambiguation

4.1 Data
We test the efficiency of our hybrid models on the English All Words tasks of Senseval-2 (Palmer et al., 2001) and Senseval-3 (Snyder and Palmer, 2004), two standard data sets for evaluating WSD. Our experiments focus on the disambiguation of nouns in these data sets.

4.2 Word Sense Tagging
A simple approach to all-words WSD was implemented in which each sense of an ambiguous word is compared against its context and the most similar chosen.

For example, suppose that we want to disambiguate
<table>
<thead>
<tr>
<th>Nouns</th>
<th>Senseval-2</th>
<th>Senseval-3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Hybrid Models - Full</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td>0.46</td>
<td>0.45</td>
</tr>
<tr>
<td>$H_p$</td>
<td>0.65</td>
<td>0.63</td>
</tr>
<tr>
<td>Hybrid Models - LSA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tilde{H}$</td>
<td>0.45</td>
<td>0.44</td>
</tr>
<tr>
<td>$\tilde{H}_p$</td>
<td>0.60</td>
<td>0.58</td>
</tr>
<tr>
<td>Hybrid Models - CCA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H^*$</td>
<td>0.44</td>
<td>0.43</td>
</tr>
<tr>
<td>$H_p^*$</td>
<td>0.61</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 3: Results obtained by hybrid models on SenseEval-2 and SenseEval-3 data sets (nouns only).

the word bank in the following sentence:

“Banks provide payment services.”

Assume that the word bank consists of two senses ‘bank.n.01’ and bank.n.02 defined as ‘sloping land (especially the slope beside a body of water)’ and “a financial institution that accepts deposits and channels the money into lending activities’ respectively.

First we consider the vectors of all the possible noun synsets containing the word bank as a synset name. Then for each context word (provide, payment and service) that exists in our semantics spaces we compute a centroid vector from its constituent senses. Finally, we compute a context vector for the entire context by summing up all the context word vectors. We select the synset of the target word that its vector has the highest cosine similarity to the context vector.

4.3 Model Parameters

The parameters we need to tune are the same as for the word similarity task and we use the best settings obtained for that task. We also experimented with varying the number of surrounding sentences used as context by testing values between ±1 and ±4. The best performance was obtained using a context created from the sentence containing the target word and ±1 sentences surrounding it.

4.4 Evaluation Metrics

Word sense disambiguation systems are evaluated by computing precision and recall. Precision measures the proportion of disambiguated words that have been correctly assigned with a sense. Recall measures the proportion of words disambiguated correctly out of all words available for disambiguation.

4.5 Results

Table 3 shows the results obtained by using our hybrid models on the two word sense disambiguation data sets. The full Synset Rank model $H_p$ is consistently better method in terms of precision and recall in both data sets. On the other hand, it is somewhat surprising that dimensionality reduction and integration of semantic spaces do not help in improving performance. That is the $\tilde{H}_p$ and $H_p^*$ models achieve lower precision and recall than the fuller $H_p$.

The Synset Distributional models $H$, $\tilde{H}$ and $H^*$ consistently fail to perform well. The difference in precision and recall compared to the Synset Rank models is between 12% and 19%. This suggests that the knowledge-based weighting of the context features generates less noisy vectors for sense tagging.

The pattern of results observed for the WSD task is somewhat different to those obtained for word similarity, where applying LSA and CCA improved performance (see Section 3). The most likely expla-
nation of this difference is that WSD requires the model to represent the possible senses of each ambiguous word. It is also important that these senses correspond to the ones used in the relevant lexicon (WordNet in this case). The Synset Rank model $H_p$ does this by making use of information from WordNet. However, these synset representations are disrupted by LSA and CCA which compress the semantic space by extracting general features from them. This is not a problem for word similarity since there is no need to model the senses found in the lexicon.

5 Related Work

Dealing with polysemy in distributional semantics is a fundamental issue since the various senses of a word type are conflated in a single vector. Previous work tackled the problem through vector adaptation, clustering and language models (Erk, 2012). Vector adaptation methods modify a traditional (i.e. polysemous) target word vector by applying pointwise operations such as addition or multiplication to that and the surrounding words in a sentence (Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al., 2011; Van de Cruys et al., 2011). Alternatively, clustering methods have been used to cluster together the different contexts a target word appears assuming that each cluster of contexts captures a different sense of the target word (Dinu and Lapata, 2010; Erk and Pado, 2010; Reisinger and Mooney, 2010). Language models have also been used to remove polysemy from word vectors by predicting words that could replace the target word given a context (Deschacht and Moens, 2009; Washtell, 2010; Moon and Erk, 2013). More recently, Polajnar and Clark (2014) applied context selection and normalisation to improve the quality of word vectors. Our hybrid models are related to the vector adaptation methods since we modify the synset vectors using its lemmas’ vectors to remove noise.

Our work is also inspired by recent work on improving classic distributional vector representations of words by incorporating information from different modalities. For example, researchers have developed methods that make use of both visual and contextual information to improve word vectors (Bruni et al., 2011; Silberer et al., 2013; Lazaridou et al., 2014). Following a similar direction, Faruqui and Dyer (2014) found that learning joint spaces from multilingual vector spaces using CCA improves the performance of standard monolingual vector spaces on semantic similarity. Fyshe et al. (2014) showed that integrating textual vector space models with brain activation data when people are reading words achieves better correlation to behavioural data than models of one modality.

Our hybrid models are also closely related to a supervised method proposed by Faruqui et al. (2015). Their method refines distributional semantic models using relational information from various semantic lexicons, including WordNet, by making linked words in these lexicons to have similar vector representations. While our models are also based on using information from WordNet for refining vector representations, they are fundamentally different. They create synset vectors in an unsupervised fashion and more importantly can be used for sense tagging.

6 Conclusions

This paper proposed hybrid models of lexical semantics that combine distributional and knowledge-based approaches and offer advantages of both techniques. A standard distributional semantic model is created from an unannotated corpus and then refined by (1) using WordNet synsets to create synset vectors; and (2) applying a graph-based technique over WordNet to reweight synset vectors. The resulting hybrid models can be viewed as enhanced distributional models using the information from WordNet to reduce the problems caused by ambiguous terms when models are created. Results show that our models perform better than traditional distributional models on lexical similarity tasks. Unlike standard distributional approaches the techniques proposed here also model polysemy and can be used to carry out word sense disambiguation.

References


Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study


