Combining Relational and Distributional Knowledge for Word Sense Disambiguation

Richard Johansson **Luis Nieto Piña**
Språkbanken, Department of Swedish, University of Gothenburg
Box 200, SE-40530 Gothenburg, Sweden
{richard.johansson, luis.nieto.pina}@svenska.gu.se

Abstract

We present a new approach to word sense disambiguation derived from recent ideas in distributional semantics. The input to the algorithm is a large unlabeled corpus and a graph describing how senses are related; no sense-annotated corpus is needed. The fundamental idea is to embed meaning representations of *senses* in the same continuous-valued vector space as the representations of *words*. In this way, the knowledge encoded in the lexical resource is combined with the information derived by the distributional methods. Once this step has been carried out, the sense representations can be plugged back into e.g. the skip-gram model, which allows us to compute scores for the different possible senses of a word in a given context.

We evaluated the new word sense disambiguation system on two Swedish test sets annotated with senses defined by the SALDO lexical resource. In both evaluations, our system soundly outperformed random and first-sense baselines. Its accuracy was slightly above that of a well-known graph-based system, while being computationally much more efficient.

1 Introduction

For NLP applications such as word sense disambiguation (WSD), it is crucial to use some sort of representation of the meaning of a word. There are two broad approaches commonly used in NLP to represent word meaning: representations based on the structure of a formal knowledge representation, and those derived from co-occurrence statistics in corpora (*distributional* representations). In a knowledge-based word meaning representation, the meaning of a word string is defined by mapping it to a symbolic concept defined in a knowledge base or ontology, and the meaning of the concept itself is defined in terms of its relations to other concepts, which can be used to deduce facts that were not stated explicitly: a *mouse* is a type of *rodent*, so it has prominent *teeth*. On the other hand, in a data-driven meaning representation, the meaning of a word in defined as a point in a geometric space, which is derived from the word's cooccurrence patterns so that words with a similar meaning end up near each other in the vector space (Turney and Pantel, 2010). The most important relation between the meaning representations of two words is typically *similarity*: a *mouse* is something quite similar to a *rat*. Similarity of meaning is often operationalized in terms of the geometry of the vector space, e.g. by defining a distance metric.

These two broad frameworks obviously have very different advantages: while the symbolic representations contain explicit and very detailed relational information, the data-driven representations handle the notion of graded similarity in a very natural way, and the fact that they typically have a wide vocabulary coverage makes it attractive to integrate them in NLP systems for additional robustness (Turian et al., 2010). However, there are many reasons to study how these two very dissimilar approaches can complement each other. Mikolov et al. (2013c) showed that vector spaces represent more structure than previously thought: they implicitly encode a wide range of syntactic and semantic relations, which can be recovered using simple linear algebra operations. For instance, the geometric relation between *Rome* and *Italy* is similar to that between *Cairo* and *Egypt*. Levy and Goldberg (2014) further analyzed how this property can be explained.

One aspect where symbolic representations seem to have an advantage is in describing *word sense ambiguity*: the fact that one surface form may correspond to more than one underlying concept. For instance, the word *mouse* can refer to a *rodent* or an *electronic device*. Except for scenarios where a small number of senses are used, lexical-semantic resources such as WordNet (Fellbaum, 1998) for English and SALDO (Borin et al., 2013) for Swedish are crucial in applications that rely on sense meaning, WSD above all.

Corpus-derived representations on the other hand typically have only one representation per surface form, which makes it hard to search e.g. for a group of words similar to the rodent sense of *mouse*[1] or to reliably use the vector in machine learning methods that generalize from the semantics of the word (Erk and Padó, 2010). One straightforward solution could be to build a vector-space semantic representation from a sense-annotated corpus, but this is infeasible since fairly large corpora are needed to induce data-driven representations of a high quality, while sense-annotated corpora are small and scarce. Instead, there have been several attempts to create vectors representing the senses of ambiguous words, most of them based on some variant of the idea first proposed by Schütze (1998): that senses can be seen as clusters of similar contexts. Further examples where this idea has reappeared include the work by Purandare and Pedersen (2004), as well as a number of recent papers (Huang et al., 2012; Moen et al., 2013; Neelakantan et al., 2014; Kågebäck et al., 2015). However, sense distributions are often highly imbalanced, it is not clear that context clusters can be reliably created for senses that occur rarely.

In this work, we build a word sense disambiguation system by combining the two approaches to representing meaning. The crucial stepping stone is the recently developed algorithm by Johansson and Nieto Piña (2015), which derives vector-space representations of word senses by embedding the graph structure of a semantic network in the word vector space. A scoring function for selecting a sense can then be derived from a word-based distributional model in a very intuitive way simply by reusing the scoring function used to construct the original word-based vector space. This approach to WSD is attractive because it can leverage corpus statistics similar to a supervised method trained on an annotated corpus, but also use the lexical-semantic resource for generalization. Moreover, the sense representation algorithm also estimates how common the different senses are; finding the predominant sense of a word also gives a strong baseline for WSD (McCarthy et al., 2007), and is of course also interesting from a lexicographical perspective.

We applied the algorithm to derive vector representations for the senses in SALDO, a Swedish semantic network (Borin et al., 2013), and we used these vectors to build a disambiguation system that can assign a SALDO sense to ambiguous words occurring in free text. To evaluate the system, we created two new benchmark sets by processing publicly available datasets. On these benchmarks, our system outperforms a random baseline by a wide margin, but also a first-sense baseline significantly. It achieves a slightly higher score than UKB, a highly accurate graph-based WSD system (Agirre and Soroa, 2009), but is several orders of magnitude faster. The highest disambiguation accuracy was achieved by combining the probabilities output by the two systems. Furthermore, in a qualitative inspection of the most ambiguous words in SALDO for each word class, we see that the sense distribution estimates provided by the sense embedding algorithm are good for nouns, adjectives, and adverbs, although less so for verbs.

2 Representing the meaning of words and senses

In NLP, the idea of representing word meaning geometrically is most closely associated with the *distributional* approach: the meaning of a word is reflected in the set of contexts in which it appears. This idea has a long tradition in linguistics and early NLP (Harris, 1954).

The easiest way to create a geometric word representation is to implement the distributional idea directly: for each word, we create a vector where each dimension corresponds to a feature describing the frequency of contexts where the target word has appeared. Typically, such a feature corresponds to the document identity or another word with which the target word has cooccurred (Sahlgren, 2006), but in principle we can define arbitrary contextual features, for instance the syntactic context (Padó and Lapata, 2007). In addition, a dimensionality reduction step may be used to map the high-dimensional sparse vector space onto a smaller-dimensional space (Landauer and

[1] According to Gyllensten and Sahlgren (2015), this problem can be remedied by making better use of the topology of the neighborhood around the search term.

Dumais, 1997; Kanerva et al., 2000).

As an alternative to context-counting vectors, geometric word representations can be derived indirectly, as a by-product when training classifiers that predict the context of a focus word. While these representations have often been built using fairly complex machine learning methods (Collobert and Weston, 2008; Turian et al., 2010), such representations can also be created using much simpler and computationally more efficient log-linear methods that seem to perform equally well (Mnih and Kavukcuoglu, 2013; Mikolov et al., 2013a). In this work, we use the *skip-gram* model by Mikolov et al. (2013a): given a focus word, the contextual classifier predicts the words around it.

2.1 From word meaning to sense meaning

The crucial stepping stone to WSD used in this work is to *embed* the semantic network in a vector space: that is, to associate each sense s_{ij} with a *sense embedding*, a vector $E(s_{ij})$ of real numbers, in a way that makes sense given the topology of the semantic network but also reflects that the vectors representing the lemmas are related to those corresponding to the underlying senses (Johansson and Nieto Piña, 2015).

Figure 1 shows an example involving an ambiguous word. The figure shows a two-dimensional projection[2] of the vector-space representation of the Swedish word *rock* (meaning either 'coat' or 'rock music') and some words related to it: *morgonrock* 'dressing gown', *jacka* 'jacket', *kappa* 'coat', *oljerock* 'oilskin coat', *långrock* 'long coat', *musik* 'music', *jazz* 'jazz', *hårdrock* 'hard rock', *punkrock* 'punk rock', *funk* 'funk'. The words for styles of popular music and the words for pieces of clothing are clearly separated, and the polysemous word *rock* seems to be dominated by its music sense.

The sense embedding algorithm will then produce vector-space representations of the two senses of *rock*. Our lexicon tells us that there are two senses, one related to clothing and the other to music. The embedding of the first sense ('coat') ends up near the other items of clothing, and the second sense ('rock music') near other styles of music. Furthermore, the embedding of the lemma consists of a mix of the embeddings of the two senses: mainly of the music sense, which reflects the fact that this sense is most frequent in corpora.

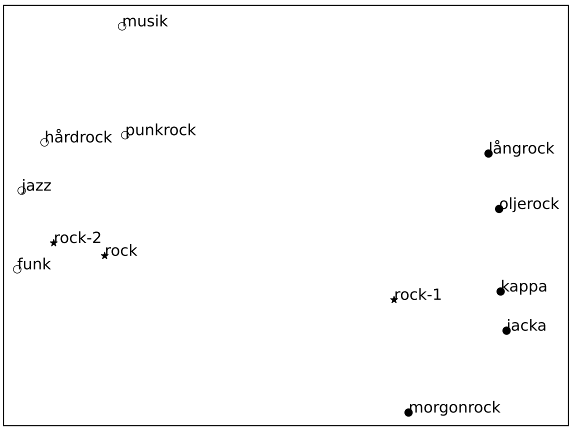

Figure 1: Vector-space representation of the Swedish word *rock* and its two senses, and some related words.

2.2 Embedding the semantic network

We now summarize the method by Johansson and Nieto Piña (2015) that implements what we described intuitively above,[3] and we start by introducing some notation. For each lemma l_i, there is a set of possible underlying concepts (senses) $s_{i1}, \ldots, s_{im_i}$ for which l_i is a surface realization. Furthermore, for each sense s_{ij}, there is a *neighborhood* set consisting of concepts semantically related to s_{ij}. Each neighbor n_{ijk} of s_{ij} is associated with a weight w_{ijk} representing the degree of semantic relatedness between s_{ij} and n_{ijk}. How we define the neighborhood, i.e. what we mean by the notion of "semantically related," will obviously have an impact on the result of the embedding process. In this work, we simply assume that it can be computed from any semantic network, e.g. by picking a number of hypernyms and hyponyms in a lexicon such as WordNet for English, or primary and secondary descriptors if we are using SALDO for Swedish.

We assume that for each lemma l_i, there exists a D-dimensional vector $F(l_i)$ of real numbers; these vectors can be computed using any method described in Section 2. Finally, we assume that there exists a *distance function* $\Delta(x,y)$ that returns a non-negative real number for each pair of vectors in $\mathbb{R}^D$; in this work, this is assumed to be the squared Euclidean distance.

[2] The figures were computed in `scikit-learn` (Pedregosa et al., 2011) using multidimensional scaling of the distances in a 512-dimensional vector space.

[3] `http://demo.spraakdata.gu.se/richard/scouse`

The goal of the algorithm is to associate each sense s_{ij} with a sense embedding, a real-valued vector $E(s_{ij})$ in the same vector space as the lemma embeddings. The lemma embeddings and the sense embeddings will be related through a *mix constraint*: the lemma embedding $F(l_i)$ is decomposed as a convex combination $\sum_j p_{ij}E(s_{ij})$, where the $\{p_{ij}\}$ are picked from the probability simplex. Intuitively, the mix variables correspond to the occurrence probabilities of the senses, but strictly speaking this is only the case when the vectors are built using context counting.

We now have the machinery to state the optimization problem that formalizes the intuition described above: the weighted sum of distances between each sense and its neighbors is minimized, and the solution to the optimization problem so that the mix constraint is satisfied for the senses for each lemma. To summarize, we have the following constrained optimization program:

$$\begin{aligned}
\underset{E,p}{\text{minimize}} \quad & \sum_{i,j,k} w_{ijk}\Delta(E(s_{ij}), E(n_{ijk})) & \\
\text{subject to} \quad & \sum_j p_{ij}E(s_{ij}) = F(l_i) & \forall i \\
& \sum_j p_{ij} = 1 & \forall i \\
& p_{ij} \geq 0 & \forall i, j
\end{aligned} \tag{1}$$

This optimization problem is hard to solve with off-the-shelf methods, but Johansson and Nieto Piña (2015) presented an approximate algorithm that works in an iterative fashion by considering one lemma at a time, while keeping the embeddings of the senses of all other lemmas fixed.

It can be noted that the vast majority of words are monosemous, so that the procedure will leave the embeddings of these words unchanged. These will then serve as as anchors when creating the embeddings for the polysemous words; the requirement that lemma embeddings are a mix of the sense embeddings will also constrain the solution.

3 Using the skip-gram model to derive a scoring function for word senses

When sense representations have been created using the method described in Section 2, they can be used in applications including WSD. Exactly how this is done in practice will depend on the properties of the original word-based vector space; in this paper, we focus on the *skip-gram* model by Mikolov et al. (2013a).

In its original formulation, the skip-gram model is based on modeling the conditional probability that a context feature c occurs given the lemma l:

$$P(c|l) = \frac{e^{F'(c) \cdot F(l)}}{Z(l)}$$

The probability is expressed in terms of lemma embeddings $F(l)$ and context $F'(c)$: note that the word and context vocabularies can be distinct, and that the corresponding embedding spaces F and F' are separate. $Z(l)$ is a normalizer so that the probabilities sum to 1.

The skip-gram training algorithm then maximizes the following objective:

$$\sum_{i,j} \log P(c_{ij}|l_i)$$

Here, the l_i are the lemmas occurring in a corpus, and c_{ij} the contextual features occurring around l_i. In practice, a number of approximations are typically applied to speed up the optimization; in this work, we applied the *negative sampling* approach (Mikolov et al., 2013b), which uses a few random samples instead of computing the normalizer $Z(l)$.

By embedding the senses in the same space as the words using the algorithm in Section 2, our implicit assumption is that contexts can be predicted by senses in the same way they can be predicted by words: that is, we can use the sense embeddings $E(s)$ in place of $F(l)$ to model the probability $P(c|s)$. Assuming the context features occurring around a token are conditionally independent, we can compute the joint probability of a sense and the context, conditioned on the lemma:

$$\begin{aligned}
P(s, c_1, \ldots, c_n|l) &= P(s|l)P(c_1, \ldots, c_n|s) \\
&= P(s|l)P(c_1|s) \cdots P(c_n|s).
\end{aligned}$$

Now we have what we need to compute the posterior sense probabilities[4]:

$$\begin{aligned}
P(s|c_1, \ldots, c_n, l) &= \frac{P(s|l)P(c_1, \ldots, c_n|s)}{\sum_{s_i} P(s_i|l)P(c_1, \ldots, c_n|s_i)} \\
&= \frac{P(s|l)e^{(F'(c_1)+\ldots+F'(c_n)) \cdot E(s)}}{\sum_{s_i} P(s_i|l)e^{(F'(c_1)+\ldots+F'(c_n)) \cdot E(s_i)}}
\end{aligned}$$

[4] We are using unnormalized probabilities here. Including $Z(s)$ makes the computation much more complex, but changes the result very little.

Finally, we note that we can use a simpler formula if we are only interested in ranking the senses, not of their exact probabilities:

$$\text{score}(s) = \log P(s|l) + \sum_{c_i} F'(c_i) \cdot E(s) \quad (2)$$

We weighted the context vector $F'(c_i)$ by the distance of the context word from the target word, corresponding to the random window sizes commonly used in the skip-gram model. We leave the investigation of more informed weighting schemes (Kågebäck et al., 2015) to future work. Furthermore, we did not make a thorough investigation of the effect of the choice of the probability distribution $P(s|l)$ of the senses, but just used a uniform distribution throughout; it would be interesting to investigate whether the accuracy could be improved by using the mix variables estimated in Section 2, or a distribution that favors the first sense.

4 Application to Swedish data

The algorithm described in Section 2 was applied to Swedish data: we started with lemma embeddings computed from a corpus, and then created sense embeddings by using the SALDO semantic network (Borin et al., 2013).

4.1 Creating lemma embeddings

We created a corpus of 1 billion words downloaded from Språkbanken, the Swedish language bank.[5] The corpora are distributed in a format where the text has been tokenized, part-of-speech-tagged and lemmatized. Compounds have been segmented automatically and when a lemma was not listed in SALDO, we used the parts of the compounds instead. The input to the software computing the lemma embedding consisted of lemma forms with concatenated part-of-speech tags, e.g. *dricka..vb* for the verb 'to drink' and *dricka..nn* for the noun 'drink'. We used the `word2vec` tool[6] to build the lemma embeddings. All the default settings were used, except the vector space dimensionality which was set to 512. We made a small modification to `word2vec` so that it outputs the context vectors as well, which we need to compute the scoring function defined in Section 3.

4.2 SALDO, a Swedish semantic network

SALDO (Borin et al., 2013) is the most comprehensive open lexical resource for Swedish. As of May 2014, it contains 125,781 entries organized into a single semantic network. Compared to WordNet (Fellbaum, 1998), there are similarities as well as considerable differences. Both resources are large, manually constructed semantic networks intended to describe the language in general rather than any specific domain. However, while both resources are hierarchical, the main lexical-semantic relation of SALDO is the *association* relation based on centrality, while in WordNet the hierarchy is taxonomic. In SALDO, when we go up in the hierarchy we move from specialized vocabulary to the most central vocabulary of the language (e.g. 'move', 'want', 'who'); in WordNet we move from specific to abstract (e.g. 'entity'). Every entry in SALDO corresponds to a specific sense of a word, and the lexicon consists of word senses only. There is no correspondence to the notion of synonym set as in WordNet. The sense distinctions in SALDO are more coarse-grained than in WordNet, which reflects a difference between the Swedish and the Anglo-Saxon traditions of lexicographical methodologies.

Each entry except a special root is connected to other entries, its *semantic descriptors*. One of the semantic descriptors is called the *primary* descriptor, and this is the entry which better than any other entry fulfills two requirements: (1) it is a semantic neighbor of the entry to be described and (2) it is more central than it. That two words are semantic neighbors means that there is a direct semantic relationship between them, for instance synonymy, hyponymy, antonymy, meronymy, or argument–predicate relationship; in practice most primary descriptors are either synonyms or hypernyms. Centrality is determined by means of several criteria. The most important criterion is frequency: a frequent word is more central than an infrequent word. Other criteria include stylistic value (a stylistically unmarked word is more central) and derivation (a derived form is less central than its base form), semantic criteria (a hypernym being more central than a hyponym).

To exemplify, here are a few instances of entries in SALDO and their descriptors.

[5] `http://spraakbanken.gu.se`

[6] `https://code.google.com/p/word2vec`

Entry	Primary	Secondary
bröd 'bread'	*mat* 'food'	*mjöl* 'flour'
äta 'eat'	*leva* 'to live'	
kollision 'collision'	*kollidera* 'to collide'	
cykel 'bicycle'	*åka* 'to go'	*hjul* 'wheel'

When using SALDO in the algorithm described in Section 2, we need to define a set of neighbors n_{ijk} for every sense s_{ij}, as well as weights w_{ijk} corresponding to the neighbors. We defined the neighbors to be the primary descriptor and inverse primaries (the senses for which s_{ij} is the primary descriptor); we excluded neighbors that did not have the same part-of-speech tag as s_{ij}. The secondary descriptors were not used. For instance, *bröd* has the primary descriptor *mat*, and a large set of inverse primaries mostly describing kinds (e.g. *rågbröd* 'rye bread') or shapes (e.g. *limpa* 'loaf') of bread. The neighborhood weights were set so that the primary descriptor and the set of inverse primaries were balanced: e.g. 1 for *mat* and 1/*N* if there were *N* inverse primaries. After computing all the weights, we normalized them so that their sum was 1. We additionally considered a number of further heuristics to build the neighborhood sets, but they did not seem to have an effect on the end result.

5 Inspection of predominant senses of highly ambiguous words

Before evaluating the full WSD system in Section 6, we carry out a qualitative study of the mix variables computed by the algorithm described in Section 2. Determining which sense of a word is the most common one gives us a strong baseline for word sense disambiguation which is often very hard to beat in practice (Navigli, 2009). McCarthy et al. (2007) presented a number of methods to find the predominant word sense in a given corpus.

In Section 2, we showed how the embedding of a lemma is decomposed into a mix of sense embeddings. Intuitively, if we assume that the mix variables to some extent correspond to the occurrence probabilities of the senses, they should give us a hint about which sense is the most frequent one. For instance, in Figure 1 the embedding of the lemma *rock* is closer to that of the second sense ('rock music') than to that of the first sense ('coat'), because the music sense is more frequent.

For each lemma, we estimated the predominant sense by selecting the sense for which the corresponding mix variable was highest. To create a dataset for evaluation, an annotator selected the most polysemous verbs, nouns, adjectives, and adverbs in SALDO (25 of each class) and determined the most frequent sense by considering a random sample of the occurrences of the lemma. Table 1 shows the accuracies of the predominant sense selection for all four word classes, as well as the average polysemy for each of the classes.

Part of speech	Accuracy	Avg. polysemy
Verb	0.48	6.28
Noun	0.76	6.12
Adjective	0.76	4.24
Adverb	0.84	2.20
Overall	0.71	4.71

Table 1: Predominant sense selection accuracy.

For nouns, adjectives, and adverbs, this heuristic works quite well. However, similar to what was seen by McCarthy et al. (2007), verbs are the most difficult to handle correctly. In our case, this has a number of reasons, not primarily that this is the most polysemous class. First of all, the most frequent verbs, which we evaluate here, often participate in multi-word units such as particle verbs and in light verb constructions. While SALDO contains information about many multi-word units, we have not considered them in this study since our preprocessing step could not deterministically extract them (as described in Section 4). Secondly, we have noticed that the sense embedding process has a problem with verbs where the sense distinction is a distinction between transitive and intransitive use, e.g. *koka* 'to boil'. This is because the transitive and intransitive senses typically are neighbors in the SALDO network, so their context sets will be almost identical and the algorithm will try to minimize the distance between them.

6 WSD evaluation

To evaluate our new WSD system, we applied it to two test sets and first compared it to a number of baselines, and finally to UKB, a well-known graph-based WSD system.

Our two test sets were the *SALDO examples* (SALDO-ex)[7] and the *Swedish FrameNet examples* (SweFN-ex)[8]. Both resources consist of sentences selected by lexicographers for illustration of word senses. At the time of our experiments, SALDO-ex contained 4,489 sentences. In each

[7] `http://spraakbanken.gu.se/resurs/saldoe`

[8] `http://spraakbanken.gu.se/resurs/swefn`

sentence, one of the tokens (the target word) has been marked up by a lexicographer and assigned a SALDO sense. SweFN-ex contained 7,991 sentences, and as in SALDO-ex the annotation consists of disambiguated target words: the difference is that instead of a SALDO sense, the target word is assigned a FrameNet frame (Fillmore and Baker, 2009). However, using the Swedish FrameNet lexicon (Friberg Heppin and Toporowska Gronostaj, 2012), frames can in most cases be deterministically mapped to SALDO senses: for instance, the first SALDO sense of the noun *stam* ('trunk' or 'stem') belongs to the frame PLANT_SUBPART, while the second sense ('tribe') is in the frame AGGREGATE.

We preprocessed these two test sets using Språkbanken's annotation services[9] to tokenize, compound-split, and lemmatize the texts and to determine the set of possible senses in a given context. All unambiguous instances were removed from the sets, and we also excluded sentences where the target consisted of more than one word. We then ended up with 1,177 and 1,429 instances in SALDO-ex and SweFN-ex, respectively. Figure 2 shows the distribution of the number of senses for target word in the combination of the two sets.

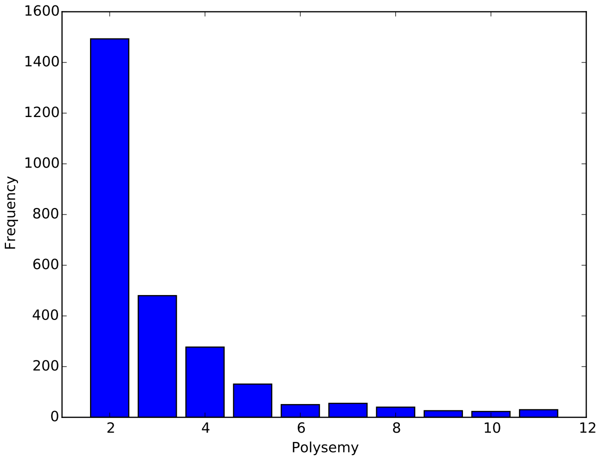

Figure 2: Histogram of the number of senses for target words in the test sets.

6.1 Comparison to baselines

We applied the contextual WSD method defined by Eq. 2 to the two test sets. As the simplest baseline, we used a random selection. A much more difficult baseline is to select the first sense[10] in the inventory; this baseline is often very hard to beat for WSD systems (Navigli, 2009). Furthermore, we evaluated a simple approach that selects the sense whose value of the mix variable in Section 2 is highest. Table 2 shows the result.

System	SALDO-ex	SweFN-ex
Random	39.3	40.3
Sense 1	52.5	53.5
By mix variables	47.6	53.9
Contextual WSD	62.7	63.3

Table 2: Comparison to baselines.

We see that our WSD system clearly outperforms not only the trivial but also the first-sense baseline. Selecting the sense by the value of the mix variable (which can be regarded as a prior probability) gives a result very similar to the first-sense baseline: this can be useful in sense inventories where senses are not ranked by frequency or importance. (This result is lower in SALDO-ex, which is heavily dominated by verbs; as we saw in Section 5, the mix variables seem less reliable for verbs.)

6.2 Analysis by part of speech

The combined set of examples from SALDO-ex and SweFN-ex contains 1,723 verbs, 575 nouns, 287 adjectives, and 15 adverbs. We made a breakdown of the result by the part of speech of the target word, and we show the result in Table 3.

PoS tag	Accuracy	Avg. polysemy
Adjective	62.3	2.7
Adverb	80.0	2.4
Noun	71.1	2.6
Verb	60.5	3.3

Table 3: Results for different parts of speech.

Again, we see that verbs pose the greatest difficult for our methods, while disambiguation accuracy is higher for nouns. Adjectives are also difficult to handle, with an accuracy just slightly higher than what we had for the verbs. (There are too few adverbs to allow any reliable conclusion to be drawn about them.) To some extent, the differences in accuracy might be expected to be correlated with the degree of polysemy, but there are

[9] `http://spraakbanken.gu.se/korp/annoteringslabb/`

[10] Unlike in WordNet, SALDO's senses are not explicitly sorted by frequency. The first sense is the one that the lexicographers regarded as the most important, which will often but not always be the same as the most frequent one.

also other factors involved, such as the structure of the SALDO network. We leave an investigation of the causes of these differences to future work.

6.3 Comparison to graph-based WSD

To find a more challenging comparison than the baselines, we applied the UKB system, a WSD system based on personalized PageRank in the sense graph, which has achieved a very competitive result for a system without any annotated training data (Agirre and Soroa, 2009). Because of limitations in the UKB software, the test sets are slightly smaller (1,055 and 1,309 instances, respectively), since we only included test instances where the lemmas could be determined unambiguously. The result is presented in Table 4. This table also includes the result of a combined system where we simply added Eq. 2 to the log of the probability output by UKB.

System	SALDO-ex	SweFN-ex
Contextual WSD	64.0	64.2
UKB	61.2	61.2
Combined	66.4	66.0

Table 4: Comparison to the UKB system.

Our system outperforms the UKB system by a slight margin; while the difference is not statistically significant, the consistent figures in the two evaluations suggest that the results reflect a true difference. However, in both evaluations, the combination comes out on top, suggesting that the two systems have complementary strengths.

Finally, we note that our system is much faster: UKB processes the SweFN-ex set in 190 seconds, while our system processes the same set in 450 milliseconds, excluding startup time.

7 Conclusion

We have presented a new method for word sense disambiguation derived from the skip-gram model. The crucial step is to embed a semantic network consisting of linked word senses into a continuous-vector word space. Unlike previous approaches for creating vector-space representations of senses, and due to the fact that we rely on the network structure, we can create representations for senses that occur very rarely in corpora. Once the senses have been embedded in the vector space, deriving a WSD model is straightforward. The word sense embedding algorithm (Johansson and Nieto Piña, 2015) takes a set of embeddings of lemmas, and uses them and the structure of the semantic network to induce the sense representations. It hinges on two ideas: 1) that sense embeddings should preserve the structure of the semantic network as much as possible, i.e. that two senses should be close geometrically if they are neighbors in the graph, and 2) that lemma embeddings can be decomposed into separate sense embeddings.

We applied the sense embedding algorithm to the senses of SALDO, a Swedish semantic network, and a vector space trained on a large Swedish corpus. These vectors were then used to implement a WSD system, which we evaluated on two new test sets annotated with SALDO senses. The results showed that our new WSD system not only outperforms the baselines, but also UKB, a high-quality graph-based WSD implementation. While the accuracies were comparable, our system is several hundred times faster than UKB.

Furthermore, we carried out a qualitative inspection of the mix variables estimated by the embedding algorithms and found that they are relatively good for predicting the predominant word senses: more so for nouns, adjectives and adverbs, less so for verbs. This result is consistent with what we saw in the quantitative evaluations, where selecting a sense based on the mix variable gave an accuracy similar to the first-sense baseline.

In future work, we will carry out a more systematic evaluation of the word sense disambiguation system in several languages. For Swedish, a more large-scale evaluation requires an annotated corpus, which will give more reliable quality estimates than the lexicographical examples we have used in this work. Fortunately, a 100,000-word multi-domain corpus of contemporary Swedish is currently being annotated on several linguistic levels in the KOALA project (Adesam et al., 2015), including word senses as defined by SALDO.

Acknowledgments

This research was funded by the Swedish Research Council under grant 2013–4944, *Distributional methods to represent the meaning of frames and constructions*, and grant 2012–5738, *Towards a knowledge-based culturomics*. We also acknowledge the University of Gothenburg for its support of the Centre for Language Technology and Språkbanken.

References

Yvonne Adesam, Gerlof Bouma, and Richard Johansson. 2015. Defining the Eukalyptus forest – the Koala treebank of Swedish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics*, Vilnius, Lithuania.

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 33–41, Athens, Greece.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47(4):1191–1211.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden.

Christiane Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT Press.

Charles J. Fillmore and Collin Baker. 2009. A frames approach to semantic analysis. In B. Heine and H. Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, pages 313–340. Oxford: OUP.

Karin Friberg Heppin and Maria Toporowska Gronostaj. 2012. The rocky road towards a Swedish FrameNet – creating SweFN. In *Proceedings of the Eighth conference on International Language Resources and Evaluation (LREC-2012)*, pages 256–261, Istanbul, Turkey.

Amaru Cuba Gyllensten and Magnus Sahlgren. 2015. Navigating the semantic horizon using relative neighborhood graphs. *CoRR*, abs/1501.02670.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23).

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Association for Computational Linguistics 2012 Conference (ACL 2012)*, Jeju Island, Korea.

Richard Johansson and Luis Nieto Piña. 2015. Embedding a semantic network in a word space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, Denver, United States.

Mikael Kågebäck, Fredrik Johansson, Richard Johansson, and Devdatt Dubhashi. 2015. Neural context embeddings for automatic discovery of word senses. In *Proceedings of the Workshop on Vector Space Modeling for NLP*, Denver, United States. To appear.

Pentti Kanerva, Jan Kristoffersson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180, Ann Arbor, United States.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.

Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations, Workshop Track*, Scottsdale, USA.

Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, USA.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26*, pages 2265–2273.

Hans Moen, Erwin Marsi, and Björn Gambäck. 2013. Towards dynamic word sense discrimination with random indexing. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 83–90, Sofia, Bulgaria.

Roberto Navigli. 2009. Word sense disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(1).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 41–48, Boston, United States.

Magnus Sahlgren. 2006. *The Word-Space Model*. Ph.D. thesis, Stockholm University.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.