# Training Word Sense Embeddings With Lexicon-based Regularization

**Luis Nieto-Piña** and **Richard Johansson**
University of Gothenburg
{luis.nieto.pina, richard.johansson}@gu.se

## Abstract

We propose to improve word sense embeddings by enriching an automatic corpus-based method with lexicographic data. Information from a lexicon is introduced into the learning algorithm's objective function through a regularizer. The incorporation of lexicographic data yields embeddings that are able to reflect expert-defined word senses, while retaining the robustness, high quality, and coverage of automatic corpus-based methods. These properties are observed in a manual inspection of the semantic clusters that different degrees of regularizer strength create in the vector space. Moreover, we evaluate the sense embeddings in two downstream applications: word sense disambiguation and semantic frame prediction, where they outperform simpler approaches. Our results show that a corpus-based model balanced with lexicographic data learns better representations and improve their performance in downstream tasks.

## 1 Introduction

Word embeddings, as a tool for representing the meaning of words based on the context in which they appear, have had a considerable impact on many of the traditional Natural Language Processing tasks in recent years. (Turian et al., 2010; Collobert et al., 2011; Socher et al., 2011; Glorot et al., 2011) This form of semantic representation has come to replace in many instances traditional count-based vectors (Baroni et al., 2014), as they yield high-quality semantic representations in a computationally efficient manner, which allows them to leverage information from large corpora.

Due to this success, some attention has been devoted to the question of whether their representational power can be refined to further advance the state of the art in those tasks that can benefit from semantic representations. One instance in which this could be realized concerns polysemous words, which has led to several attempts at representing word senses instead of simple word forms. Doing so would help avoid the situation in which several meanings of a word have to be conflated into just one embedding, typical of simple word embeddings.

Among the different approaches to learning word sense embeddings, a distinction can be made between those that make use of a semantic network (SN) and those that do not. Approaches in the latter group usually apply an unsupervised strategy for clustering instances of words based on the context formed by surrounding words. The resulting clusters are then used to represent the different meanings of a word. These representations characterize word usage in the training corpus rather than lexicographic senses, and run the risk of marginalizing under-represented word senses. Nonetheless, for well represented word senses, this strategy proves to be effective and adaptable to changes.

The alternative is to integrate an SN in the learning process. This kind of resource encodes a lexicon of word senses, connecting lexically and semantically related concepts, usually in the form of a graph. Methods that take this approach are able to work with lexicographic word senses as defined by experts, usually integrating them in different ways with corpus-learned embeddings. However, their completeness depends on the quality of the underlying SN.

In this paper, we present an approach that tries to achieve a balance between these two variants. We propose to make use of an SN for learn-

*Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 284–294,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

ing word sense embeddings by leveraging its signal through a regularizer function that is applied on top of a traditional objective function used to learn embeddings from corpora. In this manner, our model is able to merge these two opposed sources of data with the expectation that each one will balance the limitations of the other: flexible, high-quality embeddings learned from a corpus, with well defined separation between the expert-defined senses of any given polysemic word. The influence of each source of information can be regulated through a mix parameter.

As the corpus-based part of our model, we use a version of the Skip-gram (Mikolov et al., 2013) model that is modified so that it is able to learn two distinct vocabularies: word senses and word forms as introduced by Nieto-Piña and Johansson (2015). Regarding the SN data, we focus our attention on its underlying graph. We assume that neighboring nodes in such a graph correspond to semantically related concepts. Thus, given a word sense, a sequence of related word senses can be generated from its neighbors. A regularizer function can then be used to update their corresponding embeddings so that they become closer in the vector space. This has the benefit of creating clear separations between the different senses of polysemic words, precisely as they are described in the SN, even in the cases where this separation would not be clear from the data in a corpus.

We give an overview of related work in Section 2, and our model is described in detail in Section 3. The resulting word sense embeddings are evaluated in Section 4 on two separate automated tasks: word sense disambiguation (WSD) and lexical frame prediction (LFP). The experiments used for evaluation allow us to investigate the influence of the lexicographic data on the embeddings by comparing different model parameterizations. We conclude with a discussion of our results in Section 5.

## 2 Related Work

The recent success of word embeddings as effective semantic representations across the broad spectrum of NLP tasks has led to an increased interest in developing embedding methods further in order to acquire finer-grained representations able to handle polysemy and homonymy. This effort can be divided into two approaches: those that tackle the problem as an unsupervised task, aiming to discover different usages of words in corpora, and those that make use of knowledge resources as a way of injecting linguistic knowledge into the models.

Among the earliest efforts in the former group is the work of Reisinger and Mooney (2010) and Huang et al. (2012), who propose to cluster occurrences of words based on their contexts to account for different meanings. With the advent of the Skip-gram model (Mikolov et al., 2013) as an efficient way of training prediction-based word embedding models, much of the research into obtaining word sense representations revolved around it. Neelakantan et al. (2014) and Nieto-Piña and Johansson (2015) make use of context-based word sense disambiguation (WSD) during corpus training to allow on-line learning of multiple senses of a word with modified versions of Skip-gram. Li and Jurafsky (2015) and Bartunov et al. (2016) apply stochastic processes to allow for representations of a variable number of senses per word to be learnt in unsupervised fashion from corpora.

The embeddings obtained using this approach tend to be word-usage oriented, rather than represent formally defined word senses. While this is descriptive of the texts in the corpus at hand, it can be problematic for generalization. For instance, word senses that are underrepresented or absent in the training corpus will not be assigned a functional embedding. On the other hand, due to the ability of these models to process large amounts of data, well-represented word senses will acquire meaningful representations.

The alternative approach to unsupervised methods is to include data from knowledge resources, usually graph-encoded semantic networks (SN) such as WordNet (Miller, 1995). Chen et al. (2014) and Iacobacci et al. (2015) propose to make use of knowledge resources to produce a sense-annotated corpus, on which known techniques can then be applied to generate word sense embeddings. A usual way of circumventing the lack of sense-annotated corpora is to apply post-processing techniques onto pre-trained word embeddings as a way of leveraging lexical information to produce word sense embeddings. The following models share this method: Johansson and Nieto-Piña (2015) formulate an optimization problem to derive multiple word sense representations from word embeddings, while Pilehvar and Collier (2016) and one of the models proposed by Jauhar et al. (2015) use graph learning techniques to do so.

A characteristic of this approach is that these models can generate embeddings for a complete inventory of word senses. However, the dependence on manually crafted resources can potentially lead to incompleteness, in case of unlisted word senses, or to inflexibility in the face of changes in meaning, failing to account for new meanings of a word.

The model that we present in this article tries to preserve desirable characteristics from both approaches. On one side, the model learns word sense embeddings from a corpus using a predictive learning algorithm that is efficient, streamlined, and flexible with respect to being able to discriminate between different usages of a word from running text. This learning algorithm is based on the idea of adding an extra latent variable to the Skip-gram objective function to account for different senses of a word, that has been explored in previous work by Jauhar et al. (2015) and Nieto-Piña and Johansson (2015). On the other side, the learning process is guided by a regularizer function that introduces information from an SN, in an attempt to achieve a clear, complete, and fair division between the different senses of a word. Furthermore, from a technical point of view, the effect of the regularizer function is applied in parallel to the embedding learning process. This eliminates the need for a two-step training process or pre-trained word embeddings, and makes it possible to regulate the influence that each source of data (corpus and SN) has on the learning process.

## 3 Model Description

### 3.1 Learning Word Sense Embeddings

The Skip-gram word embedding model (Mikolov et al., 2013) works on the premise of training the vector for a word $w$ to be able to predict those context words $c_i$ with which it appears often together in a large training corpus, according to the following objective function:

$$\sum_{i=1}^{n} \log p(c_i|w)$$

where $p(c_i|w)$ can be approximated using the softmax function, The model, thus, works by maintaining two separate vocabularies which represent word forms in their roles as *target* and *context* words. The resulting word embeddings (usually those vectors trained for the target word vocabulary) are able to store meaningful semantic information about the words they represent.

The original Skip-gram model is, however, limited to word forms in both its vocabularies. Nieto-Piña and Johansson (2015) introduced a modification of this model in which the target vocabulary holds a variable number of vectors for each word form, intended to represent its different senses. The training objective of such a model now has the following shape:

$$\log p(s|w) + \sum_{i=1}^{n} \log p(c_i|s) \tag{1}$$

Thus the word sense embeddings are trained to maximize the log-probability of context words $c_i$ given a word's sense $s$ plus the log-probability of that sense given the word $w$. For our purposes, this prior is a constant, $p(s|w) = \frac{1}{n}$, as we do not have information on the probability of each sense of a given word.

This formulation requires a sense $s$ of word $w$ to be selected for each instance in which the objective function above is applied. This word sense disambiguation is applied on-line at training time and based on the target word's context: The sense $s$ chosen to disambiguate an instance of $w$ is the one whose embedding maximizes the dot product with the sum of the context words' embeddings.

$$\arg\max_{s} \frac{e^{s \sum_i c_i}}{\sum_s e^{s \sum_i c_i}} \tag{2}$$

This unsupervised model learns different usages of a word with minimal overhead computation on top of the original, word-based Skip-gram. The number of senses per word can be obtained from a lexicon or set to a fixed number.

### 3.2 Embedding a Lexicon

In order to adapt the graph-structured nature of the data in an SN to be used in continuous representations, we propose to introduce it through a regularizer that can act upon the same embeddings trained by the unsupervised model described above.

Any given node $s$ in a graph will have a set of neighbors $n_i$ directly connected to it. In the graph underlying an SN, we assume $n_i$ to be lexically or semantically similar to $s$. In this setting, a collection of sequences composed of word senses $s$

and $n_i$ can be collected by visiting all nodes in the SN's graph and collecting its immediate neighbors. Note that extracting such a collection of sequences from a semantic graph follows quite naturally, but in fact it could be generated from any other resource that relates concepts, such as a thesaurus, even if it is not encoded in a graph, as long as the relations it contains are relevant to the model being trained.

We propose to use a collection of sequences of related word senses to update their corresponding word sense vectors by pulling any two vectors closer together in their geometric space whenever they are encountered in a sequence. This action can be easily modeled by minimizing the following expression:

$$\sum_{i=1}^{k} ||s - n_i||^2 \qquad (3)$$

for each sequence of word senses $(s, n_1, n_2, \ldots, n_k)$. By minimizing the distance in the vector space between vectors representing interconnected concepts according to the SN's organization, the vector model is effectively representing that organization in a way that geometrical distance correlates with lexical or semantical relatedness, a central concept in the word embedding literature.

### 3.3 Combined Model

The two preceding sections describe the two parts of a combined model that is able to learn simultaneously from a corpus and an SN. This is achieved by training embeddings from a corpus with the objective described in Equation 1, and complementing this procedure with lexicographic data by means of using Equation 3 as a regularizer. The extent of the regularizer's influence on the model is adapted by a mix parameter $\rho \in [0, 1]$: the higher the value of $\rho$, the more influence the SN data has on the model, and vice versa.

Thus, the objective function of our model is as follows:

$$\log p(s|w) + (1-\rho) \sum_{i=1}^{n} \log p(c_i|s) - \rho \sum_{j=1}^{m} ||s - n_j||^2$$

In practice, this objective is realized by alternating updates through each of the model's parts, the number of which is regulated by $\rho$. Updates on the corpus-based part are executed with Skip-gram with negative sampling (Mikolov et al., 2013), adapted to work with a vocabulary of word senses as explained in §3.1.

On top of the formulation of the lexicon-based part of the model given in the previous section we propose two variations on this model in order to explore the extent to which the SN data can be used to influence the combined model explained in the following section. The initial formulation of the model will be referenced as V0 in this paper.

In the first variation (henceforth V1) we propose to only apply Equation 3 on word senses pertaining to polysemous words. If by using the SN we intend to learn clear separations between different senses of a word, it attends to reason to limit its application to those cases, while monosemous words can be sufficiently well trained by the usual corpus-based approach, and act as semantic anchors in the broader vector space.

The second variation (henceforth V2) deals with the specific architecture of the corpus-based training algorithm. As mentioned in the previous section, this model trains a target and a context vocabulary. We propose to use the regularizer to act not only on word sense vectors, but also on context (word form) vectors. By doing this we expect the context vocabulary to be ready for instances of different senses of a word, training context vectors to be potentially more effective in the disambiguation scheme introduced in Equation 2. This variation introduces an extra term into Equation 3,

$$\sum_{i=0}^{n} ||w(s) - w(n_i)||^2$$

where $w(x)$ is a mapping from a given sense $x$ to its corresponding word form.

## 4 Experiments

### 4.1 Experimental Setting

We trained the three variants of our model using different parameterizations of $\rho \in (0, 1)$. Each of these instances learned target and context embeddings of 50 dimensions, using a window of size 5 on the corpus-based part of the training algorithm, for a total number of 5 iterations over a number of updates equal to the size of the training corpus.

Below we describe the lexicon and corpus used to train the sense embeddings.

### 4.1.1 SALDO: a Semantic Network of Swedish Word Senses

SALDO (Borin et al., 2013) is the largest graph-structured semantic lexicon available for Swedish. The version used here contains roughly 125,000 concepts (word senses) organized into a single semantic network.

The sense nodes in the SALDO network are connected by edges that are defined in terms of semantic *descriptors*. A descriptor of a sense is another sense used to define its meaning. The most important descriptor is called the *primary* descriptor (PD), and since every sense in SALDO (except an abstract root sense) has a single unique PD, the PD subgraph of SALDO forms a tree. In most cases, the PD of a sense $s$ is a hypernym or a synonym of $s$, but other types of semantic relations are also possible.

To exemplify, Figure 1 shows a fragment of the PD tree. In the example, there are some cases where the PD edges correspond to hypernymy, such as *hard rock* being a type of *rock music*, which in turn is a type of *music*, but there are also other types of relations, such as *music* being defined in terms of *to sound*.

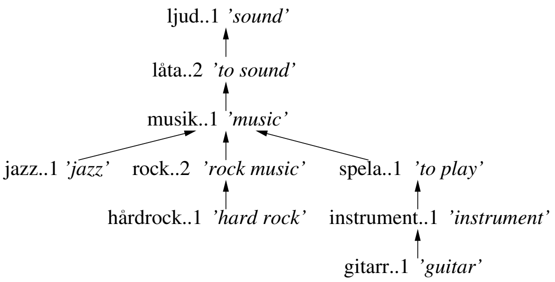


Figure 1: A fragment of the network in SALDO.

### 4.1.2 Training Corpus

For training the embedding models, we created a mixed-genre corpus of approximately 1 billion words downloaded from Språkbanken, the Swedish language bank.[1] The texts were tokenized, part-of-speech-tagged and lemmatized. Compounds were segmented automatically and when a compound-word lemma was not listed as an entry in the SALDO lexicon, we used the compound parts instead. For instance, *hårdrock* 'hard rock' would occur as a single token in the corpus, while *rockstjärna* 'rock star' would be split into two separate tokens.

## 4.2 Qualitative Inspection of Word Senses

By inspecting lists of nearest neighbors to a given embedding, some insight can be gained into how a model represents the meaning of the concept it represents. It is especially interesting in the case of polysemous words, where the neighbors of each of its senses can help judging how well it manages to separate their different meanings.

In Table 1 we list nearest neighbors for each of the two senses of the Swedish word *rock*: 'coat' and 'rock music'. The neighboring concepts in the table are extracted from two separate vector models trained with different parameterizations for the mix parameter $\rho$: The first, $\rho = 0.01$, has little influence from the lexicon and thus is similar to a corpus-only approach; the second, $\rho = 0.5$, allows for more information from the lexicon to influence the embeddings. In our corpus, the music sense is overrepresented; this can be seen in the table, where both senses trained with $\rho = 0.01$ have most of their nearest neighbors semantically related to music. The model that is more influenced by the lexicon with $\rho = 0.5$ is, however, able to learn two distinct senses. Note how the music sense is not negatively affected by this change: many of its nearest neighbors are the same in both models, and all of them keep the music-related topic in common.

It is also interesting to filter these lists of nearest neighbors to limit them to unlisted words; i.e., words that are not present in the lexicon and appear only in the corpus. This provides an observation of how well those embeddings that are trained by both parts of the model are integrated with those others whose training is based only on the corpus. Table 2 contains such lists of unlisted items for the two senses of *rock* on two models with different parameterization. It presents a similar behavior to the previous experiment: In a model with low influence from the lexicon, the representations of both senses tend towards that of the overrepresented one; when more influence from the lexicon is allowed, a clear separation of the two senses into their expected meanings is observed.

## 4.3 Word Sense Disambiguation

We trained and evaluated several parameterizations of our model on a Swedish language word sense disambiguation (WSD) task. The aim of this task is to select a sense of an instance of a polyse-

[1] http://spraakbanken.gu.se

| *rock-1* ‘coat’ | | *rock-2* ‘rock music’ | |
|---|---|---|---|
| $\rho = 0.01$ | $\rho = 0.5$ | $\rho = 0.01$ | $\rho = 0.5$ |
| *syrtut* ‘frock coat’ | *syrtut* ‘frock coat’ | *hårdrock* ‘hard rock music’ | *punk* ‘punk music’ |
| *Rhythm* ‘rhythm music’ | *kappa* ‘coat’ | *pop* ‘pop music’ | *rappa* ‘to rap’ |
| *rockband* ‘rock band’ | *kåpa* ‘cowl’ | *punk* ‘punk music’ | *rap* ‘rap music’ |
| *Peepshows* ‘peep shows’ | *päls* ‘fur coat’ | *jazza* ‘to jazz’ | *pop* ‘pop music’ |
| *skaband* ‘ska band’ | *mudd* ‘cuff’ | *dödsmetall* ‘death metal music’ | *jam* ‘music jam’ |

Table 1: Nearest neighbors for the two senses of *rock* ‘coat’ and ‘ rock music’ for different $\rho$.

| *rock-1* ‘coat’ | | *rock-2* ‘rock music’ | |
|---|---|---|---|
| $\rho = 0.01$ | $\rho = 0.5$ | $\rho = 0.01$ | $\rho = 0.5$ |
| *Rhythm* ‘rhythm music’ | *jesussandaler* ‘Jesus sandals’ | *nu-metal* ‘nu metal’ | *metal* ‘metal music’ |
| *Peepshows* ‘peep shows’ | *tubsockar* ‘tube socks’ | *goth* ‘ goth music’ | *rnb* ‘RnB music’ |
| *skabandk* ‘ska band’ | *blåjeans* ‘blue jeans’ | *psytrance* ‘ psytrance music’ | *indie* ‘indie music’ |
| *Punkrock* ‘punk rock’ | *snowjoggers* ‘snow joggers’ | *boogierock* ‘boogie rock’ | *dubstep* ‘dubstep music’ |
| *sleaze* ‘to sleaze’ | *midjekort* ‘doublet jacket’ | *synthband* ‘synth music band’ | *goth* ‘goth music’ |

Table 2: Nearest unlisted neighbors for the two senses of *rock* ‘coat’ and ‘rock music’ for different $\rho$.

mous word in context. For this purpose, we use a disambiguation mechanism similar to the one introduced in §3.1. Given an ambiguous word in context, a score is calculated for each of its possible senses by applying the expression in Equation 2; however, to correct for skewed sense distributions, we replaced the uniform prior with a power-law prior $P(s_k|w) \propto k^{-2}$, where $k$ is the numerical identifier of the sense. The highest scoring sense is then selected to disambiguate that instance of the word.

As baselines for this experiment, we used random sense and first sense[2] selection. Additionally, we show the results achieved by a disambiguation system, UKB, based on Personalized PageRank (Agirre and Soroa, 2009), and which was trained on the PD tree from SALDO. The implementation of this model makes no assumptions on the underlying graph and thus it is easily adaptable to work with any kind of SN. Our models were all parameterized with $\rho = 0.9$ based on the results obtained on the SweFN dataset. All evaluated systems including the baselines are unsupervised: none of them has used a sense-annotated corpus during training.

### 4.3.1 Sense-annotated Datasets

We evaluated the WSD systems on eleven different datasets, which to our knowledge are all sense-annotated datasets that exist for Swedish. The datasets consist of instances, where each instance is a sentence where a single target word has been selected for disambiguation.

Two datasets consist of *lexicographical examples* (Lex-Ex): the *SALDO examples* (SALDO-ex) and *Swedish FrameNet examples* (SweFN-ex). The latter of these is annotated in terms of semantic frames, but there is a deterministic mapping from frames to SALDO senses.

Two additional datasets are taken from the Senseval-2 Swedish lexical sample task (Kokkinakis et al., 2001). It uses a different sense inventory, which we mapped manually to SALDO senses. The lexical sample originally consisted of instances for 40 lemmas, out of which we removed 7 lemmas because they were unambiguous in SALDO. Since we are using an unsupervised experimental setup, we report results not only on the designated test set but also on the training set.

The other datasets come from the *Koala* annotation project (Johansson et al., 2016). The latest version consists of seven different corpora, each sampled from text in a separate domain: blogs, novels, Wikipedia, European Parliament proceedings, political news, newsletters from a government agency, and government press releases. Unlike the two lexicographical example sets and the Senseval-2 lexical sample, in which the instances have been selected by lexicographers to be prototypical and to have a good coverage of the sense variation, the instances in the Koala corpora are annotated ‘as is’ in running text.

The sentences in all datasets were tokenized, compound-split, and lemmatized, and for each target word we automatically determined the set of possible senses, given its context and inflec-

[2] No frequency information is available for SALDO’s sense inventory and the senses are not ordered by frequency. The senses are ordered by lexicographers so that the lower-numbered senses are more “central” or “primitive”, which often but not always correlates with the sense frequency.

| Test set | Subset | Size | RND | S1 | UKB | V0 | V1 | V2 |
|---|---|---|---|---|---|---|---|---|
| Lex-Ex | Average | 2,365 | 39.86 | 53.90 | 54.76 | 61.23 | **61.26** | 58.34 |
| | SweFN-Ex | 1,197 | 40.43 | 54.80 | 54.64 | 60.90 | **61.90** | 59.06 |
| | SALDO-Ex | 1,168 | 39.29 | 53.00 | 54.88 | **61.56** | 60.62 | 57.62 |
| Senseval | Average | 8,237 | 35.90 | 50.36 | 44.37 | **54.29** | 52.95 | 53.61 |
| | Train | 6,995 | 35.98 | 50.48 | 45.43 | **54.40** | 53.57 | 53.11 |
| | Test | 1,242 | 35.83 | 50.24 | 43.32 | **54.19** | 52.33 | 54.11 |
| Koala | Average | 11,167 | 41.83 | 69.50 | 67.17 | 65.17 | **73.49** | 68.59 |
| | Blogs | 2,222 | 41.86 | **71.02** | 66.70 | 60.98 | 67.78 | 64.27 |
| | Europarl | 1,838 | 41.80 | 66.16 | 65.61 | 61.26 | **71.60** | 68.28 |
| | Novels | 2,446 | 41.04 | 72.85 | 67.46 | 67.95 | **73.47** | 71.30 |
| | Wikipedia | 2,444 | 42.50 | 75.98 | 67.59 | 73.65 | **76.68** | 73.98 |
| | Political news | 1,082 | 40.60 | 69.41 | 69.04 | 67.47 | **75.69** | 69.59 |
| | Newsletters | 280 | 42.04 | 63.57 | 65.00 | 58.93 | **73.21** | 64.29 |
| | Press releases | 855 | 42.99 | 67.49 | 68.77 | 65.96 | **76.02** | 68.42 |
| Total | | 21,769 | 40.40 | 63.18 | 60.77 | 62.48 | **67.53** | 64.00 |

Table 3: WSD accuracy on baselines, UKB, and the three variants of our model ($\rho = 0.9$) on all test sets.

tion. We only considered senses of content words: nouns, verbs, adjectives, and adverbs. Multi-word targets were not included, and we removed all instances where only one sense was available.[3]

#### 4.3.2 Disambiguation Results

Table 3 shows disambiguation accuracies for our models on the datasets described above, along with the scores achieved by our baselines and the UKB model. The results of each variant of our model were obtained with a parameterization of $\rho = 0.9$, which was chosen as the best scoring value on the Swe-FN subset used as validation set. The model which only applies the regularizer to polysemous words (V1) dominates most highest scores, overtaken in some instances by V0 and in one by the first sense baseline. Note how the general magnitudes of the scores within each type of dataset underline their different characteristics explained above.

Additionally, for the sake of making a more detailed analysis of the influence of the parameter $\rho$ that dominates the extent of the lexicon's influence on the model, Figure 2 shows the average performance of our models on each dataset for a wide range of values for $\rho$. There is a clear pattern across all models and datasets by which a greater input from the SN translates into a better performance in WSD. These figures also confirm the superior performance of the variant V1 of our model seen in Table 3.

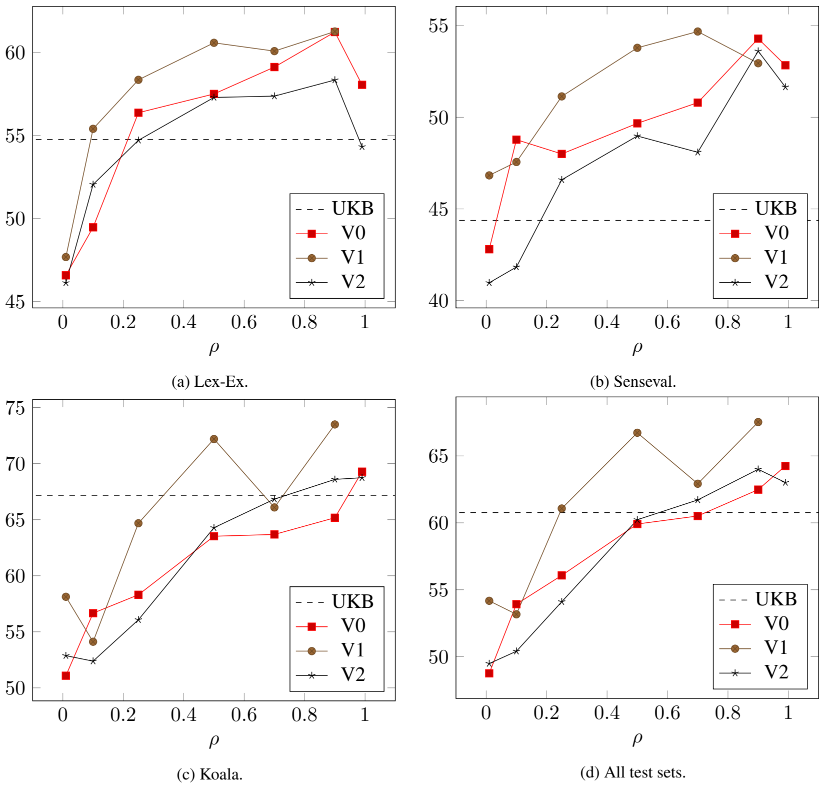
### 4.4 Frame Prediction

In our second evaluation, we investigated how well the sense vector models learned by the different training algorithms correspond to semantic classes defined by the Swedish FrameNet (Friberg Heppin and Toporowska Gronostaj, 2012). In a frame-semantic model of lexical meaning (Fillmore and Baker, 2009), the meaning of words is defined by associating them with broad semantic classes called *frames*; for instance, the word *falafel* would belong to the frame FOOD. Important classes of frames include those corresponding to objects and people, mainly populated by nouns, such as FOOD or PEOPLE_BY_AGE; verb-dominated frames corresponding to events, such as IMPACT, STATEMENT, or INGESTION; and frames dominated by adjectives, often referring to relations, qualities, and states, e.g. ORIGIN or EMOTION_DIRECTED.

In case a word has more than one sense, it may belong to more than one frame. In the Swedish FrameNet, unlike its English counterpart, these senses are explicitly defined using SALDO (see §4.1.1): for instance, for the highly polysemous noun *slag*, its first sense ('type') belongs to the frame TYPE, the second ('hit') to IMPACT, the third ('battle') to HOSTILE_ENCOUNTER, etc.

In the evaluation, we trained classifiers to determine whether a SALDO sense, represented as a sense vector, belongs to a given frame or not. To train the classifiers, we selected the 546 frames from the Swedish FrameNet for which at least 5 entries were available. In total we had 28,842 verb, noun, adjective, and adverb entries, which we split into training (67% of the entries in each

[3] In addition, to facilitate a comparison to the UKB system as a baseline, we removed a small number of instances that could not be lemmatized unambiguously.



(a) Lex-Ex.

(b) Senseval.

(c) Koala.

(d) All test sets.

Figure 2: Average WSD accuracies on all instances of each dataset for different values of $\rho$ on the three variants of our model.

frame) and test sets (33%). For each frame, we used LIBLINEAR (Fan et al., 2008) to train a linear support vector machine, using the vectors of the senses associated with that frame as positive training instances, and all other senses listed in FrameNet as negative instances.

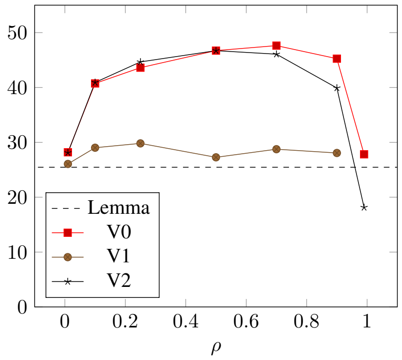


Figure 3: MAP scores for the frame prediction classifiers for the different types of models.

### 4.4.1 Evaluation Results

At test time, for each frame we applied the SVM scoring function of its classifier to each sense in the test set. The ranking induced by this score was evaluated using the Average Precision (AP) metric commonly used to evaluate rankers; the goal of this ranking step is to score the senses belonging to the frame higher than those that do not. We computed the Mean Averaged Precision (MAP) score by macro-averaging the AP scores over the set of frames.

Figure 3 shows the MAP scores of frame predictors based on different sense vector models. We compared the three training algorithms described in Section 3 for different values of the regularization strength parameter $\rho$. As a baseline, we included a model that does not distinguish between different senses: it represents a SALDO sense with the word vector of its lemma.

As the figure shows, almost all sense-aware vector models outperformed the model that just used lemma vectors. The result shows tendencies that are different from what we saw in the WSD experiments. The best MAP scores were achieved with mid-range values of $\rho$, so it seems that this task requires embeddings that strike a balance between representing the lexicon structure faithfully and representing the cooccurrence patterns in the corpus. An model with very light influence of the lexicon was hardly better than just using lemma embeddings, and unlike what we saw for the WSD task we see a strong dropoff when increasing $\rho$.

In addition, the tendencies here differ from the WSD results in that the training algorithm that only applies the lexicon-based regularizer to polysemous words (V1) gives lower scores than the other two approaches. We believe that this is because it is crucial in this task that sense vectors are clustered into coherent groups, which makes it more useful to move sense vectors closer to their neighbors even when they are monosemous; this as opposed to the WSD task, where it is more useful to leave the monosemous sense vectors in place as “anchors” for the senses of polysemous words. The context-regularized training algorithm (V2) gives no improvement over the original approach (V0), which is expected since context vectors are not used in this task.

| Frame | Lemma | V0 | V1 |
|---|---|---|---|
| ANIMALS | 0.73 | **0.86** | 0.76 |
| FOOD | 0.72 | **0.84** | 0.77 |
| REMOVING | 0.20 | **0.50** | 0.22 |
| MAKE_NOISE | 0.40 | **0.62** | 0.46 |
| ORIGIN | 0.90 | **0.90** | 0.89 |
| COLOR | 0.73 | **0.88** | 0.80 |
| FREQUENCY | 0.40 | **0.43** | 0.35 |
| TIME_VECTOR | 0.40 | **0.52** | 0.27 |

Table 4: Frame prediction AP scores for selected frames dominated by nouns, verbs, adjectives, and adverbs respectively.

To get a more detailed picture of the strengths and weaknesses of the models in this task, we selected eight frames: two frames dominated by nouns, two for verbs, two for adjectives, two for adverbs. Table 4 shows the AP scores for these frames of the lemma-vector baseline, the initial approach (V0), and the version that only regularizes senses of polysemous words (V1). All lexicon-aware models used a $\rho$ value of 0.7. Almost across the board, the V0 method gives very strong improvements. The exception is the frame ORIGIN, which contains adjectives of ethnicity and nationality (*Mexican*, *African*, etc); this set of adjectives is already quite coherently clustered by a simple word vector model and is not substantially improved by any lexicon-based approach.

## 5 Conclusion

In this article we have introduced a family of word sense embedding models that are able to leverage information from two concurrent sources of information: a semantic network and a corpus. Our hypothesis was that by combining them, the robustness and coverage of embeddings trained on a large corpus could achieve a more balanced and linguistically informed representation of the senses of polysemic words. This point has been proved in the evaluation of our models on Swedish language tasks.

A manual inspection of the word sense representation through their nearest neighbors exemplified it in §4.2. Indeed, an increased influence from the SN causes a clearer distinction between different senses of a word, even in the case where one of them is underrepresented in the corpus.

A WSD experiment was carried out on a variety of sense-annotated datasets. Our model consistently outperformed random and first sense baselines, as well as a comparable graph-based WSD system trained on a Swedish SN, which underlines the fact that the strength of our model resides in a combination of lexicon- and corpus-learning.

This is further confirmed in the evaluation of our model on a frame prediction task: A well balanced combination of lexicon and corpus data produces word sense embeddings that outperform common word embeddings when used to predict their semantic frame membership. Furthermore, this superiority is uniform across common frames dominated by different parts of speech.

An analysis of different values of our model's mix parameter $\rho$ showed the value of using lexicographic information in conjunction with corpus data. Especially on WSD, larger values of $\rho$ (i.e., more influence from the SN) generally lead to improved results.

In conclusion, we have shown that automatic word sense representation benefits greatly from using a semantic network in addition to the usual corpus-learning. The combination of these sources of information yields robust, high-quality, and balanced embeddings that excel in downstream tasks where accurate representation of word meaning is crucial. Given these findings, we intend to continue exploring more refined ways in which data from a semantic network can be leveraged to increase sense-awareness in embedding models.

## Acknowledgments

This research was funded by the Swedish Research Council under grant 2013–4944. The Koala corpus was developed in a project funded by Riksbankens Jubilemsfond, grant number In13-0320:1.

## References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. 2016. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138.

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

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

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.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520.

Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings forword and relational similarity. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP 2015*. Association for Computational Linguistics (ACL).

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proc. NAACL*, volume 1.

Richard Johansson, Yvonne Adesam, Gerlof Bouma, and Karin Hedberg. 2016. A multi-domain corpus of Swedish word sense annotation. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 3019–3022, Portorož, Slovenia.

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*, pages 1428–1433, Denver, Colorado. Association for Computational Linguistics.

Dimitrios Kokkinakis, Jerker Järborg, and Yvonne Cederholm. 2001. SENSEVAL-2: The Swedish framework. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 45–48, Toulouse, France.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Empirical Methods in Natural Language Processing (EMNLP)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

George A Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

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. Association for Computational Linguistics.

Luis Nieto-Piña and Richard Johansson. 2015. A simple and efficient method to generate word sense representations. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 465–472, Hissar, Bulgaria.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. Association for Computational Linguistics.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Joseph Turian, Lev 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. Association for Computational Linguistics.