Towards a Seamless Integration of Word Senses into Downstream NLP Applications

Mohammad Taher Pilehvar[2], Jose Camacho-Collados[1], Roberto Navigli[1] and Nigel Collier[2]

[1]Department of Computer Science, Sapienza University of Rome
[2]Department of Theoretical and Applied Linguistics, University of Cambridge
[1]{collados,navigli}@di.uniroma1.it
[2]{mp792,nhc30}@cam.ac.uk

Abstract

Lexical ambiguity can impede NLP systems from accurate understanding of semantics. Despite its potential benefits, the integration of sense-level information into NLP systems has remained understudied. By incorporating a novel disambiguation algorithm into a state-of-the-art classification model, we create a pipeline to integrate sense-level information into downstream NLP applications. We show that a simple disambiguation of the input text can lead to consistent performance improvement on multiple topic categorization and polarity detection datasets, particularly when the fine granularity of the underlying sense inventory is reduced and the document is sufficiently large. Our results also point to the need for sense representation research to focus more on *in vivo* evaluations which target the performance in downstream NLP applications rather than artificial benchmarks.

1 Introduction

As a general trend, most current Natural Language Processing (NLP) systems function at the word level, i.e. individual words constitute the most fine-grained meaning-bearing elements of their input. The word level functionality can affect the performance of these systems in two ways: (1) it can hamper their efficiency in handling words that are not encountered frequently during training, such as multiwords, inflections and derivations, and (2) it can restrict their semantic understanding to the level of words, with all their ambiguities, and thereby prevent accurate capture of the intended meanings.

The first issue has recently been alleviated by techniques that aim to boost the generalisation power of NLP systems by resorting to sub-word or character-level information (Ballesteros et al., 2015; Kim et al., 2016). The second limitation, however, has not yet been studied sufficiently. A reasonable way to handle word ambiguity, and hence to tackle the second issue, is to *semantify* the input text: transform it from its surface-level semantics to the deeper level of word senses, i.e. their intended meanings. We take a step in this direction by designing a pipeline that enables seamless integration of word senses into downstream NLP applications, while benefiting from knowledge extracted from semantic networks. To this end, we propose a quick graph-based Word Sense Disambiguation (WSD) algorithm which allows high confidence disambiguation of words without much computation overload on the system. We evaluate the pipeline in two downstream NLP applications: polarity detection and topic categorization. Specifically, we use a classification model based on Convolutional Neural Networks which has been shown to be very effective in various text classification tasks (Kalchbrenner et al., 2014; Kim, 2014; Johnson and Zhang, 2015; Tang et al., 2015; Xiao and Cho, 2016). We show that a simple disambiguation of input can lead to performance improvement of a state-of-the-art text classification system on multiple datasets, particularly for long inputs and when the granularity of the sense inventory is reduced. Our pipeline is quite flexible and modular, as it permits the integration of different WSD and sense representation techniques.

2 Motivation

With the help of an example news article from the BBC, shown in Figure 1, we highlight some of the potential deficiencies of word-based models.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1857–1869
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1170

Lewis Hamilton is heading to his fourth **F1** drivers' title after **German GP** win

The **German Grand Prix** was the last **race** before **Formula 1** heads off for its four-week summer break, so it was fitting that it consolidated the two overriding trends that have emerged so far this year.

[...]

Hockenheim was **Hamilton**'s sixth win in seven races, a remarkable run that has seen a 62-point swing between himself and **Mercedes** team-mate **Nico Rosberg**, turning a 43-point deficit into a 19-point lead.

Figure 1: Excerpt of a news article from the BBC.

Algorithm 1 Disambiguation algorithm

Input: Input text T and semantic network N
Output: Set of disambiguated senses $\hat{S}$

Graph representation of T: $(S, E) \leftarrow \text{getGraph}(T, N)$
$\hat{S} \leftarrow \emptyset$
for each iteration $i \in \{1, ..., len(T)\}$
  $\hat{s} = \text{argmax}_{s \in S} |\{(s, s') \in E : s' \in S\}|$
  $\text{maxDeg} = |\{(\hat{s}, s') \in E : s' \in S\}|$
  if $\text{maxDeg} < \theta |S| \ / \ 100$ **then**
    break
  else
    $\hat{S} \leftarrow \hat{S} \cup \{\hat{s}\}$
    $E \leftarrow E \setminus \{(s, s') : s \vee s' \in \text{getLex}(\hat{s})\}$
return Disambiguation output $\hat{S}$

Ambiguity. Language is inherently ambiguous. For instance, *Mercedes*, *race*, *Hamilton* and *Formula* can refer to several different entities or meanings. Current neural models have managed to successfully represent complex semantic associations by effectively analyzing large amounts of data. However, the word-level functionality of these systems is still a barrier to the depth of their natural language understanding. Our proposal is particularly tailored towards addressing this issue.

Multiword expressions (MWE). MWE are lexical units made up of two or more words which are idiosyncratic in nature (Sag et al., 2002), e.g, *Lewis Hamilton*, *Nico Rosberg* and *Formula 1*. Most existing word-based models ignore the interdependency between MWE's subunits and treat them as individual units. Handling MWE has been a long-standing problem in NLP and has recently received a considerable amount of interest (Tsvetkov and Wintner, 2014; Salehi et al., 2015). Our pipeline facilitates this goal.

Co-reference. Co-reference resolution of concepts and entities is not explicitly tackled by our approach. However, thanks to the fact that words that refer to the same meaning in context, e.g., *Formula 1-F1* or *German Grand Prix-German GP-Hockenheim*, are all disambiguated to the same concept, the co-reference issue is also partly addressed by our pipeline.

3 Disambiguation Algorithm

Our proposal relies on a seamless integration of word senses in word-based systems. The goal is to semantify the text prior to its being fed into the system by transforming its individual units from word surface form to the deeper level of word senses. The semantification step is mainly tailored towards resolving ambiguities, but it brings about other advantages mentioned in the previous section. The aim is to provide the system with an input of reduced ambiguity which can facilitate its decision making.

To this end, we developed a simple graph-based joint disambiguation and entity linking algorithm which can take any arbitrary semantic network as input. The gist of our disambiguation technique lies in its speed and scalability. Conventional knowledge-based disambiguation systems (Hoffart et al., 2012; Agirre et al., 2014; Moro et al., 2014; Ling et al., 2015; Pilehvar and Navigli, 2014) often rely on computationally expensive graph algorithms, which limits their application to on-the-fly processing of large number of text documents, as is the case in our experiments. Moreover, unlike supervised WSD and entity linking techniques (Zhong and Ng, 2010; Cheng and Roth, 2013; Melamud et al., 2016; Limsopatham and Collier, 2016), our algorithm relies only on semantic networks and does not require any sense-annotated data, which is limited to English and almost non-existent for other languages.

Algorithm 1 shows our procedure for disambiguating an input document T. First, we retrieve from our semantic network the list of candidate senses[1] for each content word, as well as semantic relationships among them. As a result, we obtain a graph representation (S, E) of the input text, where S is the set of candidate senses and E is the set of edges among different senses in S. The graph is, in fact, a small sub-graph of the input semantic network, N. Our algorithm then selects the best candidates iteratively. In each iteration, the

[1] As defined in the underlying sense inventory, up to trigrams. We used Stanford CoreNLP (Manning et al., 2014) for tokenization, Part-of-Speech (PoS) tagging and lemmatization.

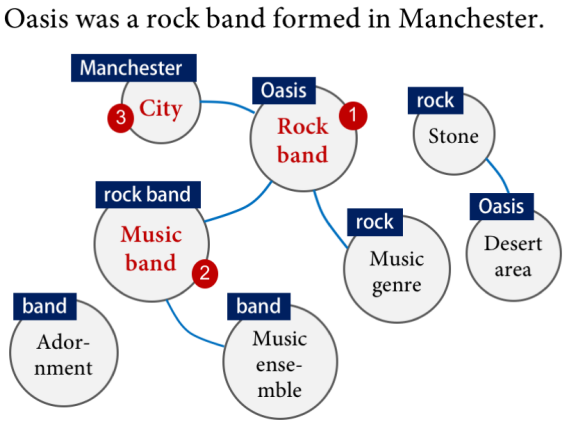

Figure 2: Simplified graph-based representation of a sample sentence.

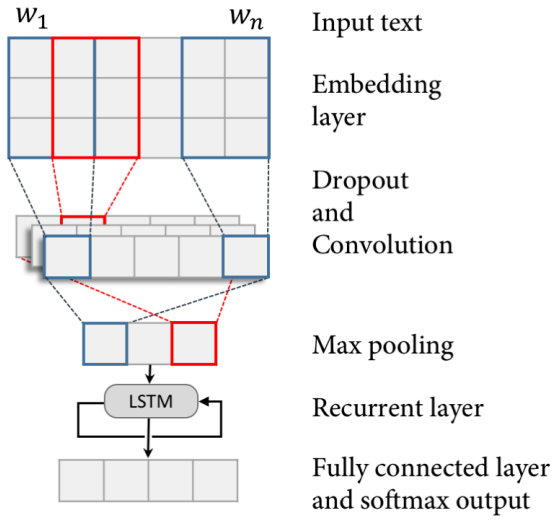

Figure 3: Text classification model architecture.

candidate sense that has the highest graph degree maxDeg is chosen as the winning sense:

$$\text{maxDeg} = \max_{s \in S} |\{(s, s') \in E : s' \in S\}| \quad (1)$$

After each iteration, when a candidate sense $\hat{s}$ is selected, all the possible candidate senses of the corresponding word (i.e. $\text{getLex}(\hat{s})$) are removed from E (line 10 in the algorithm).

Figure 2 shows a simplified version of the graph for a sample sentence. The algorithm would disambiguate the content words in this sentence as follows. It first associates *Oasis* with its *rock band* sense, since its corresponding node has the highest degree, i.e. 3. On the basis of this, the *desert* sense of *Oasis* and its link to the *stone* sense of *rock* are removed from the graph. In the second iteration, *rock band* is disambiguated as *music band* given that its degree is 2.[2] Finally, *Manchester* is associated with its *city* sense (with a degree of 1).

In order to enable disambiguating at different confidence levels, we introduce a threshold θ which determines the stopping criterion of the algorithm. Iteration continues until the following condition is fulfilled: $\text{maxDeg} < \theta|S| \,/\, 100$. This ensures that the system will only disambiguate those words for which it has a high confidence and backs off to the word form otherwise, avoiding the introduction of unwanted noise in the data for uncertain cases or for word senses that are not defined in the inventory.

[2]For bigrams and trigrams whose individual words might also be disambiguated (such as *rock* and *band* in *rock band*), the longest unit has the highest priority (i.e. *rock band*).

4 Classification Model

In our experiments, we use a standard neural network based classification approach which is similar to the Convolution Neural Network classifier of Kim (2014) and the pioneering model of Collobert et al. (2011). Figure 3 depicts the architecture of the model. The network receives the concatenated vector representations of the input words, $\mathbf{v}_{1:n} = \mathbf{v}_1 \oplus \mathbf{v}_2 \oplus \cdots \oplus \mathbf{v}_n$, and applies (convolves) filters F on windows of h words, $m_i = f(\text{F}.\mathbf{v}_{i:i+h-1} + b)$, where b is a bias term and $f()$ is a non-linear function, for which we use ReLU (Nair and Hinton, 2010). The convolution transforms the input text to a feature map $m = [m_1, m_2, \ldots, m_{n-h+1}]$. A max pooling operation then selects the most salient feature $\hat{m} = \max\{m\}$ for each filter.

In the network of Kim (2014), the pooled features are directly passed to a fully connected softmax layer whose outputs are class probabilities. However, we add a recurrent layer before softmax in order to enable better capturing of long-distance dependencies. It has been shown by Xiao and Cho (2016) that a recurrent layer can replace multiple layers of convolution and be beneficial, particularly when the length of input text grows. Specifically, we use a Long Short-Term Memory (Hochreiter and Schmidhuber, 1997, LSTM) as our recurrent layer which was originally proposed to avoid the vanishing gradient problem and has proven its abilities in capturing distant dependencies. The LSTM unit computes three gate vectors

(forget, input, and output) as follows:

$$\begin{aligned}\mathbf{f}_t &= \sigma(\mathbf{W}_f\ g_t + \mathbf{U}_f\ h_{t-1} + \mathbf{b}_f),\\ \mathbf{i}_t &= \sigma(\mathbf{W}_i\ g_t + \mathbf{U}_i\ h_{t-1} + \mathbf{b}_i),\\ \mathbf{o}_t &= \sigma(\mathbf{W}_o\ g_t + \mathbf{U}_o\ h_{t-1} + \mathbf{b}_o),\end{aligned} \quad (2)$$

where $\mathbf{W}$, $\mathbf{U}$, and $\mathbf{b}$ are model parameters and g and h are input and output sequences, respectively. The cell state vector $\mathbf{c}_t$ is then computed as $\mathbf{c}_t = \mathbf{f}_t\ \mathbf{c}_{t-1} + \mathbf{i}_t\ \tanh(\tilde{\mathbf{c}}_t)$ where $\tilde{c}_t = \mathbf{W}_c\ g_t + \mathbf{U}_c\ h_{t-1}$. Finally, the output sequence is computed as $h_t = \mathbf{o}_t\ \tanh(\mathbf{c}_t)$. As for regularization, we used dropout (Hinton et al., 2012) after the embedding layer.

We perform experiments with two configurations of the embedding layer: (1) *Random*, initialized randomly and updated during training, and (2) *Pre-trained*, initialized by pre-trained representations and updated during training. In the following section we describe the pre-trained word and sense representation used for the initialization of the second configuration.

4.1 Pre-trained Word and Sense Embeddings

One of the main advantages of neural models is that they usually represent the input words as dense vectors. This can significantly boost a system's generalisation power and results in improved performance (Zou et al., 2013; Bordes et al., 2014; Kim, 2014; Weiss et al., 2015, *inter-alia*). This feature also enables us to directly plug in pre-trained sense representations and check them in a downstream application.

In our experiments we generate a set of sense embeddings by extending DeConf, a recent technique with state-of-the-art performance on multiple semantic similarity benchmarks (Pilehvar and Collier, 2016). We leave the evaluation of other representations to future work. DeConf gets a pre-trained set of word embeddings and computes sense embeddings in the same semantic space. To this end, the approach exploits the semantic network of WordNet (Miller, 1995), using the Personalized PageRank (Haveliwala, 2002) algorithm, and obtains a set of *sense biasing words* $\mathcal{B}_s$ for a word sense s. The sense representation of s is then obtained using the following formula:

$$\hat{\mathbf{v}}(s) = \frac{1}{|\mathcal{B}_s|}\sum_{i=1}^{|\mathcal{B}_s|} \mathrm{e}^{\frac{-i}{\delta}}\mathbf{v}(w_i), \quad (3)$$

where δ is a decay parameter and $\mathbf{v}(w_i)$ is the embedding of w_i, i.e. the i^{th} word in the sense biasing list of s, i.e. $\mathcal{B}_s$. We follow Pilehvar and Collier (2016) and set $\delta = 5$. Finally, the vector for sense s is calculated as the average of $\hat{\mathbf{v}}(s)$ and the embedding of its corresponding word.

Owing to its reliance on WordNet's semantic network, DeConf is limited to generating only those word senses that are covered by this lexical resource. We propose to use Wikipedia in order to expand the vocabulary of the computed word senses. Wikipedia provides a high coverage of named entities and domain-specific terms in many languages, while at the same time also benefiting from a continuous update by collaborators. Moreover, it can easily be viewed as a sense inventory where individual articles are word senses arranged through hyperlinks and redirections.

Camacho-Collados et al. (2016b) proposed NASARI[3], a technique to compute the most salient words for each Wikipedia page. These salient words were computed by exploiting the structure and content of Wikipedia and proved effective in tasks such as Word Sense Disambiguation (Tripodi and Pelillo, 2017; Camacho-Collados et al., 2016a), knowledge-base construction (Lieto et al., 2016), domain-adapted hypernym discovery (Espinosa-Anke et al., 2016; Camacho-Collados and Navigli, 2017) or object recognition (Young et al., 2016). We view these lists as *biasing words* for individual Wikipedia pages, and then leverage the exponential decay function (Equation 3) to compute new sense embeddings in the same semantic space. In order to represent both WordNet and Wikipedia sense representations in the same space, we rely on the WordNet-Wikipedia mapping provided by BabelNet[4] (Navigli and Ponzetto, 2012). For the WordNet synsets which are mapped to Wikipedia pages in BabelNet, we average the corresponding Wikipedia-based and WordNet-based sense embeddings.

4.2 Pre-trained Supersense Embeddings

It has been argued that WordNet sense distinctions are too fine-grained for many NLP applications (Hovy et al., 2013). The issue can be tackled by grouping together similar senses of the same word, either using automatic clustering techniques (Navigli, 2006; Agirre and Lopez, 2003; Snow et al., 2007) or with the help of WordNet's lexicographer

[3]We downloaded the salient words for Wikipedia pages (NASARI English lexical vectors, version 3.0) from http://lcl.uniroma1.it/nasari/

[4]We used the Java API from http://babelnet.org

files[5]. Various applications have been shown to improve upon moving from senses to supersenses (Rüd et al., 2011; Severyn et al., 2013; Flekova and Gurevych, 2016). In WordNet's lexicographer files there are a total of 44 sense clusters, referred to as supersenses, for categories such as *event*, *animal*, and *quantity*. In our experiments we use these supersenses in order to reduce granularity of our WordNet and Wikipedia senses. To generate supersense embeddings, we simply average the embeddings of senses in the corresponding cluster.

5 Evaluation

We evaluated our model on two classification tasks: topic categorization (Section 5.2) and polarity detection (Section 5.3). In the following section we present the common experimental setup.

5.1 Experimental setup

Classification model. Throughout all the experiments we used the classification model described in Section 4. The general architecture of the model was the same for both tasks, with slight variations in hyperparameters given the different natures of the tasks, following the values suggested by Kim (2014) and Xiao and Cho (2016) for the two tasks. Hyperparameters were fixed across all configurations in the corresponding tasks. The embedding layer was fixed to 300 dimensions, irrespective of the configuration, i.e. Random and Pre-trained. For both tasks the evaluation was carried out by 10-fold cross-validation unless standard training-testing splits were available. The disambiguation threshold θ (cf. Section 3) was tuned on the training portion of the corresponding data, over seven values in [0,3] in steps of 0.5.[6] We used Keras (Chollet, 2015) and Theano (Team, 2016) for our model implementations.

Semantic network. The integration of senses was carried out as described in Section 3. For disambiguating with both WordNet and Wikipedia senses we relied on the joint semantic network of Wikipedia hyperlinks and WordNet via the mapping provided by BabelNet.[7]

Pre-trained word and sense embeddings. Throughout all the experiments we used Word2vec (Mikolov et al., 2013) embeddings, trained on the Google News corpus.[8] We truncated this set to its 250K most frequent words. We also used WordNet 3.0 (Fellbaum, 1998) and the Wikipedia dump of November 2014 to compute the sense embeddings (see Section 4.1). As a result, we obtained a set of 757,262 sense embeddings in the same space as the pre-trained Word2vec word embeddings. We used DeConf (Pilehvar and Collier, 2016) as our pre-trained WordNet sense embeddings. All vectors had a fixed dimensionality of 300.

Supersenses. In addition to WordNet senses, we experimented with supersenses (see Section 4.2) to check how reducing granularity would affect system performance. For obtaining supersenses in a given text we relied on our disambiguation pipeline and simply clustered together senses belonging to the same WordNet supersense.

Evaluation measures. We report the results in terms of standard accuracy and F1 measures.[9]

5.2 Topic Categorization

The task of topic categorization consists of assigning a label (i.e. topic) to a given document from a pre-defined set of labels.

5.2.1 Datasets

For this task we used two newswire and one medical topic categorization datasets. Table 1 summarizes the statistics of each dataset.[10] The **BBC news** dataset[11] (Greene and Cunningham, 2006) comprises news articles taken from BBC, divided into five topics: business, entertainment, politics, sport and tech. **Newsgroups** (Lang, 1995) is a collection of 11,314 documents for training and 7532 for testing[12] divided into six topics: computing, sport and motor vehicles, science, politics, reli-

[5] https://wordnet.princeton.edu/man/lexnames.5WN.html

[6] We observed that values higher than 3 led to very few disambiguations. While the best results were generally achieved in the [1.5,2.5] range, performance differences across threshold values were not statistically significant in most cases.

[7] For simplicity we refer to this joint sense inventory as Wikipedia, but note that WordNet senses are also covered.

[8] https://code.google.com/archive/p/word2vec/

[9] Since all models in our experiments provide full coverage, accuracy and F1 denote micro- and macro-averaged F1, respectively (Yang, 1999).

[10] The coverage of the datasets was computed using the 250K top words in the Google News Word2vec embeddings.

[11] http://mlg.ucd.ie/datasets/bbc.html

[12] We used the train-test partition available at http://qwone.com/~jason/20Newsgroups/

Dataset	Domain	No. of classes	No. of docs	Avg. doc. size	Size of vocab.	Coverage	Evaluation
BBC	News	5	2,225	439.5	35,628	87.4%	10 cross valid.
Newsgroups	News	6	18,846	394.0	225,046	83.4%	Train-Test
Ohsumed	Medical	23	23,166	201.2	65,323	79.3%	Train-Test

Table 1: Statistics of the topic categorization datasets.

Initialization	Input type		BBC News		Newsgroups		Ohsumed	
			Acc	F1	Acc	F1	Acc	F1
Random	**Word**		93.0	92.8	87.7	85.6	30.1	20.7
	Sense	**WordNet**	**93.5**	**93.3**	**88.1**	**86.9**	27.2†	18.3
		Wikipedia	92.7	92.5	86.7	84.9	29.7	**20.9**
	Supersense	**WordNet**	**93.6**	**93.4**	**90.1***	**89.0**	**31.8***	**22.0**
		Wikipedia	**94.6***	**94.4**	**88.5**	**85.8**	**31.1**	**21.3**
Pre-trained	**Word**		97.6	97.5	91.1	90.6	29.4	20.1
	Sense	**WordNet**	97.3	97.1	90.2	88.6	**30.2**	**20.4**
		Wikipedia	96.3	96.2	89.6†	88.9	**32.4**	**22.3**
	Supersense	**WordNet**	96.8	96.7	89.6	88.9	**29.5**	19.9
		Wikipedia	96.9	96.9	88.6	87.4	**30.6***	**20.3**

Table 2: Classification performance at the word, sense, and supersense levels with random and pre-trained embedding initialization. We show in bold those settings that improve the word-based model.

gion and sales.[13] Finally, **Ohsumed**[14] is a collection of medical abstracts from MEDLINE, an online medical information database, categorized according to 23 cardiovascular diseases. For our experiments we used the partition split of 10,433 documents for training and 12,733 for testing.[15]

5.2.2 Results

Table 2 shows the results of our classification model and its variants on the three datasets.[16] When the embedding layer is initialized randomly, the model integrated with word senses consistently improves over the word-based model, particularly when the fine-granularity of the underlying sense inventory is reduced using supersenses (with statistically significant gains on the three datasets). This highlights the fact that a simple disambiguation of the input can bring about performance gain for a state-of-the-art classification system. Also, the better performance of supersenses suggests that the sense distinctions of WordNet are too fine-grained for the topic categorization task. However, when pre-trained representations are used to initialize the embedding layer, no improvement is observed over the word-based model. This can be attributed to the quality of the representations, as the model utilizing them was unable to benefit from the advantage offered by sense distinctions. Our results suggest that research in sense representation should put special emphasis on real-world evaluations on benchmarks for downstream applications, rather than on artificial tasks such as word similarity. In fact, research has previously shown that word similarity might not constitute a reliable proxy to measure the performance of word embeddings in downstream applications (Tsvetkov et al., 2015; Chiu et al., 2016).

Among the three datasets, Ohsumed proves to be the most challenging one, mainly for its larger number of classes (i.e. 23) and its domain-specific nature (i.e. medicine). Interestingly, unlike for the other two datasets, the introduction of pre-trained word embeddings to the system results in reduced performance on Ohsumed. This suggests that general domain embeddings might not be beneficial

[13] The dataset has 20 fine-grained categories clustered into six general topics. We used the coarse-grained labels for their clearer distinction and consistency with BBC topics.

[14] ftp://medir.ohsu.edu/pub/ohsumed

[15] http://disi.unitn.it/moschitti/corpora.htm

[16] Symbols * and † indicate the sense-based model with the smallest margin to the word-based model whose accuracy is statistically significant at 0.95 confidence level according to unpaired t-test (* for positive and † for negative change).

in specialized domains, which corroborates previous findings by Yadav et al. (2017) on a different task, i.e. entity extraction. This performance drop may also be due to diachronic issues (Ohsumed dates back to the 1980s) and low coverage: the pre-trained Word2vec embeddings cover 79.3% of the words in Ohsumed (see Table 1), in contrast to the higher coverage on the newswire datasets, i.e. Newsgroups (83.4%) and BBC (87.4%). However, also note that the best overall performance is attained when our pre-trained Wikipedia sense embeddings are used. This highlights the effectiveness of Wikipedia in handling domain-specific entities, thanks to its broad sense inventory.

5.3 Polarity Detection

Polarity detection is the most popular evaluation framework for sentiment analysis (Dong et al., 2015). The task is essentially a binary classification which determines if the sentiment of a given sentence or document is negative or positive.

5.3.1 Datasets

For the polarity detection task we used five standard evaluation datasets. Table 1 summarizes statistics. **PL04** (Pang and Lee, 2004) is a polarity detection dataset composed of full movie reviews. **PL05**[18] (Pang and Lee, 2005), instead, is composed of short snippets from movie reviews. **RTC** contains critic reviews from Rotten Tomatoes[19], divided into 436,000 training and 2,000 test instances. **IMDB** (Maas et al., 2011) includes 50,000 movie reviews, split evenly between training and test. Finally, we used the **Stanford** Sentiment dataset (Socher et al., 2013), which associates each review with a value that denotes its sentiment. To be consistent with the binary classification of the other datasets, we removed the neutral phrases according to the dataset's scale (between 0.4 and 0.6) and considered the reviews whose values were below 0.4 as negative and above 0.6 as positive. This resulted in a binary polarity dataset of 119,783 phrases. Unlike the previous four datasets, this dataset does not contain an even distribution of positive and negative labels.

5.3.2 Results

Table 4 lists accuracy performance of our classification model and all its variants on five polarity detection datasets. Results are generally better than those of Kim (2014), showing that the addition of the recurrent layer to the model (cf. Section 4) was beneficial. However, interestingly, no consistent performance gain is observed in the polarity detection task, when the model is provided with disambiguated input, particularly for datasets with relatively short reviews. We attribute this to the nature of the task. Firstly, given that words rarely happen to be ambiguous with respect to their sentiment, the semantic sense distinctions provided by the disambiguation stage do not assist the classifier in better decision making, and instead introduce data sparsity. Secondly, since the datasets mostly contain short texts, e.g., sentences or snippets, the disambiguation algorithm does not have sufficient context to make high-confidence judgements, resulting in fewer disambiguations or less reliable ones. In the following section we perform a more in-depth analysis of the impact of document size on the performance of our sense-based models.

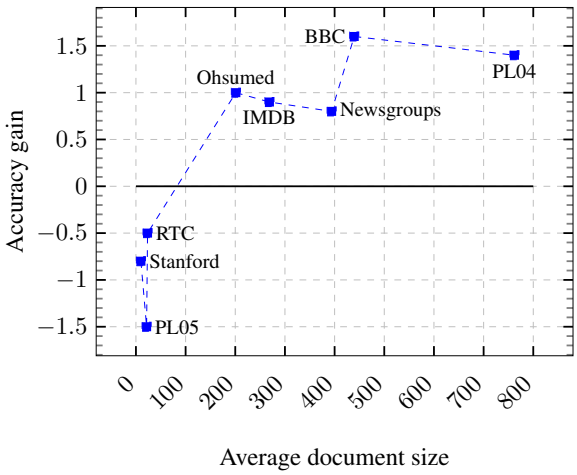

Figure 4: Relation between average document size and performance improvement using Wikipedia supersenses with random initialization.

5.4 Analysis

Document size. A detailed analysis revealed a relation between document size (the number of tokens) and performance gain of our sense-level model. We show in Figure 4 how these two vary for our most consistent configuration, i.e. Wikipedia supersenses, with random initialization. Interestingly, as a general trend, the performance gain increases with average document size, irre-

[18] Both PL04 and PL05 were downloaded from http://www.cs.cornell.edu/people/pabo/movie-review-data/

[19] http://www.rottentomatoes.com

[19] Stanford is the only unbalanced dataset, but F1 results were almost identical to accuracy.

Dataset	Type	No. of docs	Avg. doc. size	Vocabulary size	Coverage	Evaluation
RTC	Snippets	438,000	23.4	128,056	81.3%	Train-Test
IMDB	Reviews	50,000	268.8	140,172	82.5%	Train-Test
PL05	Snippets	10,662	21.5	19,825	81.3%	10 cross valid.
PL04	Reviews	2,000	762.1	45,077	82.4%	10 cross valid.
Stanford	Phrases	119,783	10.0	19,400	81.6%	10 cross valid.

Table 3: Statistics of the polarity detection datasets.

Initialization	Input type		RTC	IMDB	PL05	PL04	Stanford
Random	**Word**		83.6	87.7	77.3	67.9	91.8
	Sense	**WordNet**	83.2	87.4	76.6	67.4	91.3
		Wikipedia	83.1	**88.0**	75.9†	67.1	91.0
	Supersense	**WordNet**	**84.4**	**88.0**	75.9	66.2	91.4†
		Wikipedia	83.1	**88.4***	75.8	**69.3***	91.0
Pre-trained	**Word**		85.5	88.3	80.2	72.5	93.1
	Sense	**WordNet**	83.4	**88.3**	79.2	69.7†	92.6
		Wikipedia	83.8	87.0†	79.2	**73.1**	92.3
	Supersense	**WordNet**	85.2	**88.8**	79.5	**73.8**	92.7†
		Wikipedia	84.2	87.9	78.3†	**72.6**	92.2

Table 4: Accuracy performance on five polarity detection datasets. Given that polarity datasets are balanced[17], we do not report F1 which would have been identical to accuracy.

spective of the classification task. We attribute this to two main factors:

1. **Sparsity**: Splitting a word into multiple word senses can have the negative side effect that the corresponding training data for that word is distributed among multiple independent senses. This reduces the training instances per word sense, which might affect the classifier's performance, particularly when senses are semantically related (in comparison to fine-grained senses, supersenses address this issue to some extent).

2. **Disambiguation quality**: As also mentioned previously, our disambiguation algorithm requires the input text to be sufficiently large so as to create a graph with an adequate number of coherent connections to function effectively. In fact, for topic categorization, in which the documents are relatively long, our algorithm manages to disambiguate a larger proportion of words in documents with high confidence. The lower performance of graph-based disambiguation algorithms on short texts is a known issue (Moro et al., 2014; Raganato et al., 2017), the tackling of which remains an area of exploration.

Senses granularity. Our results showed that reducing fine-granularity of sense distinctions can be beneficial to both tasks, irrespective of the underlying sense inventory, i.e. WordNet or Wikipedia, which corroborates previous findings (Hovy et al., 2013; Flekova and Gurevych, 2016). This suggests that text classification does not require fine-grained semantic distinctions. In this work we used a simple technique based on WordNet's lexicographer files for coarsening senses in this sense inventory as well as in Wikipedia. We leave the exploration of this promising area as well as the evaluation of other granularity reduction techniques for WordNet (Snow et al., 2007; Bhagwani et al., 2013) and Wikipedia (Dandala et al., 2013) sense inventories to future work.

6 Related Work

The past few years have witnessed a growing research interest in semantic representation, mainly as a consequence of the word embedding tsunami

(Mikolov et al., 2013; Pennington et al., 2014). Soon after their introduction, word embeddings were integrated into different NLP applications, thanks to the migration of the field to deep learning and the fact that most deep learning models view words as dense vectors. The waves of the word embedding tsunami have also lapped on the shores of sense representation. Several techniques have been proposed that either extend word embedding models to cluster contexts and induce senses, usually referred to as unsupervised sense representations (Schütze, 1998; Reisinger and Mooney, 2010; Huang et al., 2012; Neelakantan et al., 2014; Guo et al., 2014; Tian et al., 2014; Šuster et al., 2016; Ettinger et al., 2016; Qiu et al., 2016) or exploit external sense inventories and lexical resources for generating sense representations for individual meanings of words (Chen et al., 2014; Johansson and Pina, 2015; Jauhar et al., 2015; Iacobacci et al., 2015; Rothe and Schütze, 2015; Camacho-Collados et al., 2016b; Mancini et al., 2016; Pilehvar and Collier, 2016).

However, the integration of sense representations into deep learning models has not been so straightforward, and research in this field has often opted for alternative evaluation benchmarks such as WSD, or artificial tasks, such as word similarity. Consequently, the problem of integrating sense representations into downstream NLP applications has remained understudied, despite the potential benefits it can have. Li and Jurafsky (2015) proposed a "multi-sense embedding" pipeline to check the benefit that can be gained by replacing word embeddings with sense embeddings in multiple tasks. With the help of two simple disambiguation algorithms, unsupervised sense embeddings were integrated into various downstream applications, with varying degrees of success. Given the interdependency of sense representation and disambiguation in this model, it is very difficult to introduce alternative algorithms into its pipeline, either to benefit from the state of the art, or to carry out an evaluation. Instead, our pipeline provides the advantage of being modular: thanks to its use of disambiguation in the pre-processing stage and use of sense representations that are linked to external sense inventories, different WSD techniques and sense representations can be easily plugged in and checked. Along the same lines, Flekova and Gurevych (2016) proposed a technique for learning supersense representations, using automatically-annotated corpora. Coupled with a supersense tagger, the representations were fed into a neural network classifier as additional features to the word-based input. Through a set of experiments, Flekova and Gurevych (2016) showed that the supersense enrichment can be beneficial to a range of binary classification tasks. Our proposal is different in that it focuses directly on the benefits that can be gained by semantifying the input, i.e. reducing lexical ambiguity in the input text, rather than assisting the model with additional sources of knowledge.

7 Conclusion and Future Work

We proposed a pipeline for the integration of sense level knowledge into a state-of-the-art text classifier. We showed that a simple disambiguation of the input can lead to consistent performance gain, particularly for longer documents and when the granularity of the underlying sense inventory is reduced. Our pipeline is modular and can be used as an *in vivo* evaluation framework for WSD and sense representation techniques. We release our code and data (including pre-trained sense and supersense embeddings) at https://pilehvar.github.io/sensecnn/ to allow further checking of the choice of hyperparameters and to allow further analysis and comparison. We hope that our work will foster future research on the integration of sense-level knowledge into downstream applications. As future work, we plan to investigate the extension of the approach to other languages and applications. Also, given the promising results observed for supersenses, we plan to investigate task-specific coarsening of sense inventories, particularly Wikipedia, or the use of SentiWordNet (Baccianella et al., 2010), which could be more suitable for polarity detection.

Acknowledgments

The authors gratefully acknowledge the support of the MRC grant No. MR/M025160/1 for PheneBank and ERC Consolidator Grant MOUSSE No. 726487. Jose Camacho-Collados is supported by a Google Doctoral Fellowship in Natural Language Processing. Nigel Collier is supported by EPSRC Grant No. EP/M005089/1. We thank Jim McManus for his suggestions on the manuscript and the anonymous reviewers for their helpful comments.

References

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40(1):57–84.

Eneko Agirre and Oier Lopez. 2003. Clustering WordNet word senses. In *Proceedings of Recent Advances in Natural Language Processing*. Borovets, Bulgaria, pages 121–130.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*. volume 10, pages 2200–2204.

Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of EMNLP*.

Sumit Bhagwani, Shrutiranjan Satapathy, and Harish Karnick. 2013. Merging word senses. In *Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing*. Seattle, Washington, USA, pages 11–19.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *EMNLP*.

José Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato, and Roberto Navigli. 2016a. A Large-Scale Multilingual Disambiguation of Glosses. In *Proceedings of LREC*. Portoroz, Slovenia, pages 1701–1708.

Jose Camacho-Collados and Roberto Navigli. 2017. BabelDomains: Large-Scale Domain Labeling of Lexical Resources. In *Proceedings of EACL (2)*. Valencia, Spain.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016b. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence* 240:36–64.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP*. Doha, Qatar, pages 1025–1035.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of EMNLP*. Seattle, Washington, pages 1787–1796.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the Workshop on Evaluating Vector Space Representations for NLP, ACL*.

Franois Chollet. 2015. Keras. https://github.com/fchollet/keras.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12:2493–2537.

Bharath Dandala, Chris Hokamp, Rada Mihalcea, and Razvan C. Bunescu. 2013. Sense clustering using Wikipedia. In *Proceedings of Recent Advances in Natural Language Processing*. Hissar, Bulgaria, pages 164–171.

Li Dong, Furu Wei, Shujie Liu, Ming Zhou, and Ke Xu. 2015. A statistical parsing framework for sentiment classification. *Computational Linguistics* 41(2):293–336.

Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of EMNLP*. pages 424–435.

Allyson Ettinger, Philip Resnik, and Marine Carpuat. 2016. Retrofitting sense-specific word vectors using parallel text. In *Proceedings of NAACL-HLT*. San Diego, California, pages 1378–1383.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

Lucie Flekova and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of ACL*.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International conference on Machine learning*. ACM, pages 377–384.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *COLING*. pages 497–507.

Taher H. Haveliwala. 2002. Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web*. Hawaii, USA, pages 517–526.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Compasutation* 9(8):1735–1780.

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of CIKM*. pages 545–554.

Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence* 194:2–27.

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 ACL*. Jeju Island, Korea, pages 873–882.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*. Beijing, China, pages 95–105.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of NAACL*. Denver, Colorado, pages 683–693.

Richard Johansson and Luis Nieto Pina. 2015. Embedding a semantic network in a word space. In *Proceedings of NAACL*. Denver, Colorado, pages 1428–1433.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of NAACL*. Denver, Colorado, pages 103–112.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*. Baltimore, USA, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*. Doha, Qatar, pages 1746–1751.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proceedings of AAAI*. Phoenix, Arizona, pages 2741–2749.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*. Tahoe City, California, pages 331–339.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of EMNLP*. Lisbon, Portugal, pages 683–693.

Antonio Lieto, Enrico Mensa, and Daniele P Radicioni. 2016. A resource-driven approach for anchoring linguistic resources to conceptual spaces. In *AI* IA 2016 Advances in Artificial Intelligence*, Springer, pages 435–449.

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of ACL*. Berlin, Germany, pages 1014–1023.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics* 3:315–328.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL-HLT*. Portland, Oregon, USA, pages 142–150.

Massimiliano Mancini, José Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2016. Embedding words and senses together via joint knowledge-enhanced training. *CoRR* abs/1612.02703. http://arxiv.org/abs/1612.02703.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany, pages 51–61.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

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

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)* 2:231–244.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*. pages 807–814.

Roberto Navigli. 2006. Meaningful clustering of senses helps boost Word Sense Disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*. Sydney, Australia, pages 105–112.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

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 EMNLP*. Doha, Qatar, pages 1059–1069.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*. Barcelona, Spain, pages 51–61.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*. Ann Arbor, Michigan, pages 115–124.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*. pages 1532–1543.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of EMNLP*. Austin, TX, pages 1680–1690.

Mohammad Taher Pilehvar and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art Word Sense Disambiguation. *Computational Linguistics* 40(4).

Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-dependent sense embedding. In *Proceedings of EMNLP*. Austin, Texas, pages 183–191.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of EACL*. Valencia, Spain, pages 99–110.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of ACL*. pages 109–117.

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of ACL*. Beijing, China, pages 1793–1803.

Stefan Rüd, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of ACL-HLT*. Portland, Oregon, USA, pages 965–975.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City, Mexico, pages 1–15.

Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015. A word embedding approach to predicting the compositionality of multiword expressions. In *NAACL-HTL*. Denver, Colorado, pages 977–983.

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

Aliaksei Severyn, Massimo Nicosia, and Alessandro Moschitti. 2013. Learning semantic textual similarity with structural representations. In *Proceedings of ACL (2)*. Sofia, Bulgaria, pages 714–718.

Rion Snow, Sushant Prakash, Daniel Jurafsky, and Andrew Y. Ng. 2007. Learning to merge word senses. In *Proceedings of EMNLP*. Prague, Czech Republic, pages 1005–1014.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng, and Christopher Potts. 2013. Parsing with compositional vector grammars. In *Proceedings of EMNLP*. Sofia, Bulgaria, pages 455–465.

Simon Šuster, Ivan Titov, and Gertjan van Noord. 2016. Bilingual learning of multi-sense embeddings with discrete autoencoders. In *Proceedings of NAACL-HLT*. San Diego, California, pages 1346–1356.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Porceedings of EMNLP*. Lisbon, Portugal, pages 1422–1432.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *COLING*. pages 151–160.

Rocco Tripodi and Marcello Pelillo. 2017. A game-theoretic approach to word sense disambiguation. *Computational Linguistics* 43(1):31–70.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of EMNLP (2)*. Lisbon, Portugal, pages 2049–2054.

Yulia Tsvetkov and Shuly Wintner. 2014. Identification of multiword expressions by combining multiple linguistic information sources. *Computational Linguistics* 40(2):449–468.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of ACL*. Beijing, China, pages 323–333.

Yijun Xiao and Kyunghyun Cho. 2016. Efficient character-level document classification by combining convolution and recurrent layers. *CoRR* abs/1602.00367.

Shweta Yadav, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. 2017. Entity extraction in biomedical corpora: An approach to evaluate word embedding features with pso based feature selection. In

Proceedings of EACL. Valencia, Spain, pages 1159–1170.

Yiming Yang. 1999. An evaluation of statistical approaches to text categorization. *Information retrieval* 1(1-2):69–90.

Jay Young, Valerio Basile, Lars Kunze, Elena Cabrio, and Nick Hawes. 2016. Towards lifelong object learning by integrating situated robot perception and semantic web mining. In *Proceedings of the European Conference on Artificial Intelligence conference*. The Hague, Netherland, pages 1458–1466.

Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proceedings of the ACL System Demonstrations*. Uppsala, Sweden, pages 78–83.

Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*. Seattle, USA, pages 1393–1398.