ContextBERT: Contextual Graph Representation Learning in Text Disambiguation

Mozhgan Saeidi
orcidID0000-0002-1736-0737

mozhgan.saeidi@dal.ca

Abstract. Word representations derived by neural language models have been shown to effectively carry useful semantic information to improve the final results of various Natural Language Processing tasks. The information provided by these representations encodes the subtle distinction that might occur between different meanings of the same word. However, these representations do not include the input text's information, as the context, and a semantic knowledge base network. This integration of context and semantic network is helpful in NLP tasks, specifically in the lexical ambiguity problem. In this paper, we first analyzed the defects of current state-of-the-art representations learning approaches, and second, we present a word representation learning method, named ContextBERT, that is aware of the semantic knowledge base network and the context. ContextBERT is a novel approach to producing sense embeddings for the lexical meanings within a lexical knowledge base, using pre-trained BERT model The novel difference in our representation is the integration of the knowledge base information and the input text. Our representations enable a simple 1-Nearest-Neighbour algorithm to perform state-of-the-art models in the English Word Sense Disambiguation task.

Keywords: Sense Embedding · Representation Learning · Word Sense Disambiguation · Pre-trained Language Models · Semantic Networks.

1 Introduction

Text disambiguation is one of many problems in Natural Language Processing (NLP) tasks. In this task, we have an input text including a word with multiple possible meanings based on a semantic knowledge base network, and the question is which one of those multiple meanings is the best meaning match for the word in the text, based on its context [17,32]. The context here refers to the input document text. The text disambiguation task is mostly referred to as Word Sense Disambiguation (WSD) task in NLP. Knowledge bases are different in nature [2]; for example, WordNet is a lexical graph database of semantic relations (e.g., synonyms, hyponyms, and meronyms) between words. Synonyms are grouped into synsets with short definitions and usage examples. WordNet can thus be seen as a combination and extension of a dictionary and thesaurus [3]. Wikipedia is a hyperlink-based graph between encyclopedia entries[1].

The text ambiguity task is easy for humans by considering the context. The context enables us to identify the correct meaning of the ambiguous words. In computational methods, we try to enhance the algorithms to mimic this approach. These methods often represent their output by linking each word occurrence to an explicit representation of the chosen sense [37]. Two approaches to tackle this problem are the machine learning-based approach and the knowledge-based approach. In the machine learning-based approach, systems are trained to perform the task [32]. The knowledge-based approach requires external lexical resources such as Wikipedia, WordNet [13], a dictionary, or a thesaurus. The machine learning-based approaches mainly focus on achieving maximal precision or recall and have their drawbacks of run-time and space requirement at the time of classifier training [4]. So, knowledge-based disambiguation methods still have advantages to study. Among different knowledge-based methods, coherence-based has been more effective in explaining it. In the coherence-based approach, one important factor is the coherence of the whole text after disambiguation, while in other approaches, this factor might change to considering the coherence of each sentence or paragraph.

There are different factors that play important roles in solving the WSD problem, including word representation. Word representations have been shown to play an important role in different Natural Language Processing (NLP) tasks, especially in disambiguation tasks. There are many different approaches to generate word representation embeddings. Recently, embeddings based on pre-trained deep language models have attracted much interest. These models have proved to be superior to classical embeddings for several NLP tasks, including Word Sense Disambiguation (WSD). Some of most used models in this category are including ELMO [22], BERT [5], and XLNET [38]. these models encode several pieces of linguistic information in their word representations. These representations differ from static neural word embeddings [21] in that they are dependent on the surrounding context of the word [29].This difference makes these vector representations especially interesting for disambiguation, where effective contextual representations can be highly beneficial for resolving lexical ambiguity. In addition, these representations enabled sense-annotated corpora to be exploited more efficiently [10].

In this study, next section, we overview different current approaches for text embedding with focusing on the contextualized word representation. We analyzed the effectiveness of these methods on different types of words. We show the pros and cons of these state-of-the-art models in word representation learning on parts of speeches are. In our representation, we enhanced this detected defectiveness to improve representations. Our novel contribution provides a new representation of words using the context of the input text and the context of the knowledge base and uses the nearest neighbor heuristic algorithm to disambiguate ambiguous words. We finally compare the performance of our proposed approach with our representations with the most current methods in the disambiguation task.

2 Related Work

The Word Sense Disambiguation is one core problem in NLP, which addresses the ambiguity of words in a given context. In this task, we have access to two main sources of information to disambiguate the ambiguous words. One source is a semantic network,

and the other is sense-annotated corpora. Semantic networks encode a more general knowledge that is not tied to a specific task, and the information enclosed therein is usually employed for WSD by knowledge-based approaches. Instead, sense annotated corpora are tailored to the WSD task and are typically used as training sets for supervised systems. Therefore, we divide the WSD approaches into two categories of knowledge-based and supervised approaches [17].

2.1 Knowledge-Based Approaches

In the knowledge-based methods, the semantic network structure of the knowledge base is used, e.g., Wikipedia [7], WordNet [13], BabelNet [19], to find the correct meaning based on its context for each input word [16]. These approaches employ algorithms on graphs to address the word ambiguity in texts [1]. Disambiguation based on Wikipedia has been demonstrated to be comparable in terms of coverage to domain-specific ontology [36] since it has broad coverage, with documents about entities in a variety of domains [31]. The most widely used lexical knowledge base is WordNet, although it is restricted to the English lexicon, limiting its usefulness to other vocabularies. BabelNet solves this challenge by combining lexical and semantic information from various sources in numerous languages, allowing knowledge-based approaches to scale across all languages it supports. Despite their potential to scale across languages, knowledge-based techniques on English fall short of supervised systems in terms of accuracy.

2.2 Supervised Approaches

The supervised approaches surpass the knowledge-based ones in all English data sets. These approaches use neural architectures, or SVM models, while still suffering from the need of creating large manually-curated corpora, which reduces their usability to scale over unseen words [20]. Automatic data augmentation approaches [33] developed methods to cover more words, senses, and languages.

In recent years, the contextual representation learning approaches have improved the performance of WSD models, where they have been employed for the creation of sense embeddings. Most NLP tasks now use semantic representations derived from language models. There are static word embeddings and contextual embeddings. This section covers aspects of the word and contextual embeddings that are especially important to our work.

Static Word Embeddings Word embeddings are distributional semantic representations usually with one of two goals: predict context words given a target word (Skip-Gram), or the inverse (CBOW) [12]. In both, the target word is at the center, and the context is considered as a fixed-length window that slides over tokenized text. These models produce dense word representations. One limit for word embeddings, as mentioned before, is meaning conflict around word types. This limitation affects the capability of these word embeddings for the ones that are sensitive to their context [28].

Contextual Word Embeddings The problem mentioned as a limitation for the static word embeddings is solved in this type of embeddings. The critical difference is that the contextual embeddings are sensitive to the context. It allows the same word types to have different representations according to their context. The first work in contextual embeddings is ELMO [22], which is followed by BERT [5], as the state-of-the-art model. The critical feature of BERT, which makes it different, is the quality of its representations [30]. Its results are task-specific fine-tuning of pre-trained neural language models. The recent representations which we analyze their effectiveness are based on these two models [24,23].

In our representation, we use different resources to build the vectors. In this section, we provide information on these resources.

2.3 Wikipedia

is the largest electronic encyclopedia freely available on the Web. Wikipedia organized its information via articles called Wikipedia pages. Disambiguation based on Wikipedia has been demonstrated to be comparable in terms of coverage to domain-specific ontology [36] since it has broad coverage with documents about entities in a variety of domains [11]. Moreover, Wikipedia has unique advantages over the majority of other knowledge bases, which include [40]:

- The text in Wikipedia is primarily factual and available in a variety of languages.
- Articles in Wikipedia can be directly linked to the entities they describe in other knowledge bases.
- Mentions of entities in Wikipedia articles often provide a link to the relevant Wikipedia pages, thus providing labeled examples of entity mentions and associated anchor texts in various contexts, which could be used for supervised learning in WSD with Wikipedia as the knowledge base.

2.4 BabelNet

is a multilingual semantic network, which comprises information coming from heterogeneous resources, such as WordNet, and Wikipedia [19]. It is organized into synsets, i.e., sets of synonyms that express a single concept, which, in their turn, are connected to each other by different types of relationships. One of Babelnet's features which is useful for our representation is *hypernym-hyponym* relations. In this relation, each concept is connected to other concepts via hypernym relation (for generalization) and via hyponym relation (for specification). *Semantically-related* relation is the other feature that we use that expresses a general notation of relatedness between concepts. The last feature of Babelent used in this work is *mapping to Wikipedia*, which maps its concepts to Wikipedia pages.

2.5 WordNet

is the most widely used lexical knowledge repository for English. It can be seen as a graph, with nodes representing concepts (synsets) and edges representing semantic relationships between them. Each synset has a set of synonyms, such as the lemmas spring, fountain, and natural spring in the synset, A natural flow of groundwater.

2.6 SemCor

is the typical manually-curated corpus for WSD, with about 220K words tagged with 25K distinct WordNet meanings, resulting in annotated contexts for around 15% of WordNet synsets.

2.7 BERT

is a Transformer-based language model for learning contextual representations of words in a text. The contextualized representation of BERT is the key factor that has changed the performance in many NLP tasks, such as text ambiguity. In our representations, we use BERT-base-cased to generate the vectors of each sense [5].

2.8 SBERT

is a modification of the pre-trained BERT network that uses siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. We use this sentence representation when generating the vector representations of sense sentences, both in the input text and in the knowledge base text.

2.9 Graph Convolutional Network

Graph Convolutional Networks (GCN) is a very powerful multilayer neural network architecture for machine learning on graphs [8]. GCN operates directly on a graph and induces embedding vectors of nodes based on the properties of their neighborhoods. In fact, they are so powerful that even a randomly initiated $2-$layer GCN can produce useful feature representations of nodes in networks[2]. Formally, consider a graph $G = (V, E)$, where $V(|V| = n)$ and E are sets of nodes and edges, respectively. Every node is assumed to be connected to itself, i.e., $(v, v) \in E$ for any v which the reason for this assumption is mentioned at the end of this paragraph. Let $X \in R^{n\times m}$ be a matrix containing all n nodes with their features, where m is the dimension of the feature vectors, each row $x_v \in R_m$ is the feature vector for v. We introduce an adjacency matrix A of G and its degree matrix D, where $D_{ii} = \sum_j A_{ij}$. Because of self-loops, the diagonal elements of A are all 1. We now have a graph, its adjacency matrix A, and a set of input feature X. After applying the propagation rule $f(X, A) = AX$ and $X = I$, the representation of each node (each row) is now a sum of its neighbor's features. In other words, the graph convolutional layer represents each node as an aggregate of its neighborhood. The reason for considering the self-loops in the graph is the aggregated representation of a node to include its own features.

For a one-layer GCN, the new k-dimensional node feature matrix $L^{(1)} \in R^{n\times k}$ is computed as:

$$L^{(1)} = \rho(\hat{A}XW_0) \tag{1}$$

[2] The notation we used for GCN in this paper are the same as notations in [39]

where $\hat{A}$ is $D^{-0.5}AD^{-0.5}$, the normalized symmetric adjacency matrix and $W_0 \in R^{m \times k}$ is the weight matrix. The ρ is the activation function (RELU); $\rho(x) = max(0, x)$. GCN can capture information only about immediate neighbors with one layer of convolution. When multiple GCN layers are stacked, information about larger neighborhoods are integrated;

$$L^{(j+1)} = \rho(\hat{A}L^{j}W_j) \tag{2}$$

which j is the layer number and $L^0 = X$. In other words, the size of the second dimension of the weight matrix determines the number of features at the next layer. The feature representations can be normalized by node degree by transforming the adjacency matrix A by multiplying it with the inverse degree matrix D. First we used the simple propagation rule $f(X, A) = D^{-1}AX$, while then improved it. The improved version is inspired by a recent work [8] that proposes a fast approximate spectral graph convolutions using a spectral propagation rule $f(X, A) = \sigma(D^{-0.5}\hat{A}D^{-0.5}XW)$. They showed this property is very useful, that connected nodes tend to be similar (e.g. have the same label).

3 Methodology

This section presents our novel embedding approach of creating sense representations of BabelNet senses. Our representation learning is created by combining semantic and textual information from the first paragraph of each sense's Wikipedia page and the input document paragraph, which includes the ambiguous word. our approach uses the representation power of neural language models, i.e., BERT and SBERT. We divide our approach into the following steps:

3.1 Context Retrieval

In this step, we collect suitable contextual information from Wikipedia for each given concept in the semantic network. Similar to [34], we exploit the mapping between synsets and Wikipedia pages available in BabelNet, as well as its taxonomic structure, to collect textual information that is relevant to a target synset s. For each synset s, we collect all the connected concepts to s through hyponym and hypernym connections of the BabelNet knowledge base. We show this set of related synsets to s by R_s which is:

$$R_s = \{s^{'} | (s, s^{'}) \in E\}$$

Similar to [34], we use E as the set includes all hyponyms and hypernyms connections. In this work, for each page p_s, we consider the first opening paragraph of the page and compute its lexical vector by summing the SBERT vector representation of the sentences in this first paragraph. These lexical representations are later used for the similarity score finding between p_s and $p_{s'}$, for each $s^{'} \in R_s$ by using the weighted overlap measure from [25], which is defined as follows:

$$WO(p_1, p_2) = (\sum_{w \in O} \frac{1}{r_w^{p_1} + r_w^{p_2}})(\sum_{i=1}^{|O|} \frac{1}{2i})^{-1}$$

where O is the set of overlapping dimensions of p_1 and p_2 and $r_w^{p_i}$ is the rank of the word w in the lexical vector of p_i. We preferred the weighted overlap over the more common cosine similarity as it has proven to perform better when comparing sparse vector representations [25]. Similar to [34], Once we have scored all the $(p_s, p_{s'})$ pairs, we create partitions of R_s, each comprising all the senses s' connected to s with the same relation r, where r can be one among: hypernymy, and hyponymy. We then retain from each partition only the top-k scored senses according to $WO(p_s, p_{s'_i})$, which we set $k = 15$ in our experiments.

3.2 Word Embedding

In the second step, we use BERT for the representation of the given concepts from the input text. For each ambiguous word–which we call this word by mention– of the input, we extract the BERT representation of the mention. Using the BabelNet relations of hyponymy and hypernymy, we extract all synsets of mention from BabelNet (set E). For each one of these senses, use the link structure of BabelNet and Wikipedia; we collect all the Wikipedia pages for each sense. We use BERT representation for the second time to generate vector representation for senses. In the settings, each word is represented as a 300-dimensional vector, as the BERT dimension.

3.3 Sense Embedding

In this step, we build the final representation of each concept. From the previous step, we took the representation of mention, $R(m)$, and the representation of each one of its senses. We show the representations of each k sense of m by $R(s_i)$ which i varies from 1 to k, based on their similarity scores. Our unique representations combine the mention representation with sense representation, averaging the vector representations of $R(m)$ and $R(s_i)$. If mention m has k senses, our model generates k different representations of $R(m, s_1)$, $R(m, s_2)$, ..., $R(m, s_k)$. Since the dimension representation of $R(m)$ and each $R(s_i)$ is 300, these averaged representation dimensions are 300. Next novelty in our representations is ranking the k senses of each mention based on their relevancy degree to the context. To this aim, we average the representations of the first step. In the first step, we took the representation of the input text paragraph, which contains the ambiguous mention, show it by $R(PD)$ which stands for representation of the **P**aragraph of the input **D**ocument. In the first step, we also took the representation of the first paragraph of the Wikipedia page, which represents it by $R(PW)$, which stands for representation of the first **P**aragraph of the **W**ikipedia page. Finally, we average these two representations as $R(PD, PW)$. In the $R(PD, PW)$, the context is constant for each sense since the input text as the context is constant for each possible sense of the ambiguous words. The dimension of this averaged representation is also equal to the word representation, so it makes it possible to calculate their cosine similarities. To rank the senses most related to the context, we use the cosine similarity as follows:

$$\text{Sim}(m, s_i) = Cosine(R(m, s_i), R(PD, PW)) \text{ , for i=1 , ... , k}$$

This ranking provides the most similar sense to the context for each mention. This novelty makes this representation more effective than the previous contextualized-based

embeddings, especially in the task of sense disambiguation. At the end of these three steps, each sense is associated with a vector that encodes both the contextual information and knowledge base semantic information from the extracted context of Wikipedia and its gloss.

We consider each mention of the document as one node of the graph, and a newly added node (redirect link) will connect with its nearest neighbor by using cosine similarity, which makes the edges of the graph. The cosine similarity between two nodes on the edges makes the weight matrix. The number of nodes in the text graph $|V|$ is the number of mentions. For each sense s, we use an integrated representation of its mention m with its own representation, i.e,. $R(m, s)$. We set the feature matrix X as extracted representation of BERT as input to GCN. The dimension of the feature matrix here is 300, as it is the averaged representation length of two BERT embeddings, one for the mention and the other for the sense.

As mentioned, formally, the weights of edge between node i and node j defines as:

$$W_{ij} = \text{cosine sim}(R(i), R(j)) = \frac{R(i).R(j)}{||R(i)||||R(j)||} \tag{3}$$

which $R(i)$ is our representation of node i.

After building the graph, we feed it into a simple $2-$layers GCN as [8], the second layer node (mention,sense) embeddings are fed into a softmax classifier:

$$Z = softmax(\hat{A}RELU(\hat{A}XW_0)W_1) \tag{4}$$

where

$$\hat{A} = D^{-0.5}AD^{-0.5}$$

and

$$softmax(x_i) = \frac{1}{Z}exp(x_i)$$

with $S = \sum_i exp(x_i)$. The loss function is the one defined in [39] as:

$$L = -\sum_{d \in Y}\sum_{f=1}^{F} Y_{df} ln Z_{df} \tag{5}$$

where Y_D is the set of mention indices that have labels and F is the dimension of the output feature. Y is the label indicator matrix. Similar to [39], the weight parameters W_0 and W_1 can be trained via gradient descent. The $\hat{A}XW_0$ contains the first layer (mention,sense) and embeddings, and $\hat{A}RELU(\hat{A}XW_0)W_1$ contains the second layer (mention,sense) and embeddings. This two-layer GCN performs message passing between nodes to two steps away, maximum. Therefore, the two-layer GCN allows the exchange of information between pairs of nodes. This GCN model on our experimental datasets shows better performance than a one-layer model and models with more than two layers. This shows the validity of our model, based on similar results in other recent works [8,9].

4 Experimental Setup

We present the settings of our evaluation of our representation in the English WSD task. This setup includes the benchmark, our representation setup for disambiguation task and state-of-the-art WSD models as our comparison systems.

Evaluation Benchmark We use the English WSD test set framework which is constructed by five standard evaluation benchmark datasets[3]. It is included of Senseval-2 [6], Senseval-3 [35], SemEval-07 [26], SemEval-13 [18], SemEval-15 [15] along with ALL, i.e., the concatenation of all the test sets [27].

Experiment Setup In our experiments, we use BERT pre-trained cased model. Similar to [34], among all the configurations reported by Devlin et al. (2019), we used the sum of the last four hidden layers as contextual embeddings of the words since they showed it has better performance. In order to be able to compare our system with supervised models, we build a supervised version of our representations. This version combines the gloss and contextual information with the sense-annotated contexts in SemCor [14], a corpus of 40K sentences where words have been manually annotated with a WordNet meaning. We leveraged SemCor for building a representation of each sense therein. To this end, we followed [22], given a mention-sense pair (m, s), we collected all the sentences $c_1, ..., c_n$ where m appears tagged with s. Then, we fed all the retrieved sentences into BERT and extracted the embeddings BERT(c_1, m), ... , BERT(c_n, m). The final embedding of s was built by the average of its context and sense gloss vectors and its representation coming from SemCor, i.e., the average of BERT(c_1, m), ... , BERT(c_n, m). We note that when a sense did not appear in SemCor, and we built its embedding by replacing the SemCor part of the vector with its sense gloss representation.

WSD Model For WSD modeling, we employed a 1-nearest neighbor approach– as previous methods in the literature– to test our representations on the WSD task. For each target word m in the test set, we computed its contextual embedding by means of BERT and compared it against the embeddings of our representation associated with the senses of m. Hence, we took as a prediction for the target word the sense corresponding to its nearest neighbor. We note that the embeddings produced by our representations are created by averaging two BERT representations, i.e., context and sense gloss (see Section 3.3), hence we repeated the BERT embedding of the target instance to match the number of dimensions.

Comparison Systems We compared our representation against the best recent performing systems evaluated on the English WSD task. LMMS is one of these systems which generates sense embedding with full coverage of Wordnet. It uses pre-trained ELMO and BERT models, as well as the relations in a lexical knowledge base to create contextual embeddings [10]. SensEmBERT is the next system that relies on different resources for building sense vectors. These resources include Wikipedia, BabelNet, NASARI lexical vectors, and BERT. It computes context-aware representations of BabelNet senses by combining the semantic and textual information derived from multilingual resources. This model uses the BabelNet mapping between WordNet senses and Wikipedia pages which drops the need for sense-annotated corpora [34]. The next

[3] http://lcl.uniroma1.it/wsdeval/

Table 1: F-Measure performance of WSD evaluation framework on the test sets of the unified dataset.

Model	Senseval-2	Senseval-3	Semeval-7	Semeval-13	Semeval-15	All
BERT	77.1±0.3	73.2±0.4	66.1±0.3	71.5±0.2	74.4±0.3	73.8±0.3
LMMS	76.1±0.6	75.5±0.2	68.2±0.4	75.2±0.3	77.1±0.4	75.3±0.2
SensEmBERT	72.4±0.1	69.8±0.2	60.1±0.4	78.8±0.1	75.1±0.2	72.6±0.3
ARES	78.2±0.3	77.2±0.1	71.1±0.2	77.2±0.2	83.1±0.2	77.8±0.1
our model	79.6±0.2	78.5±0.2	74.6±0.3	79.3±0.6	82.9±0.4	78.9±0.1

comparison system is ARES, a semi-supervised approach to produce sense embeddings for all the word senses in a language vocabulary. ARES compensates for the lack of manually annotated examples for a large portion of words' meanings. ARES is the most recent contextualized word embedding system, to our knowledge. In our comparisons, we also considered BERT as a comparison system since it is at the core of all the considered methods. BERT also has shown good performance in most NLP tasks by using pre-trained neural networks.

5 Results

The results of our evaluations on the WSD task are represented in this section. We show the effectiveness of our representation by comparing it with the existing state-of-the-art models on the standard WSD benchmarks. In Table 1 we report the results of our representation and compare it against the results obtained from other state-of-the-art approaches on all the nominal instances of the test sets in the framework of [27]. All performances are reported in terms of F1-measure, i.e., the harmonic mean of precision and recall. As we can see, our model achieves the best results on the datasets when compared to other precious contextualized approaches. It indicates that our representation is competitive with these previous models. These results show the novel idea in the nature of creating this new representation has improved the lexical ambiguity. It is a good indicator of the dependency of the WSD task to the representation that is aware of the context and the information extracted from the reference knowledge base.

Analysis by Part-of-Speech One other possible way to analyze the errors that arise in WSD with each embedding approach is to measure the frequency of mis-disambiguation in different parts of speech. The considered parts of speech are nouns, verbs, adjectives, and adverbs, as are the covered types in the datasets. The F-measure performance of the 1-NN WSD of each embedding on All dataset is shown in Table 3 which is categorized by parts of speech. As it shows, the type in which its disambiguation has been correct more than other types is adverbs. At the same time, verbs are the ones that are difficult to disambiguate because they have the lowest mis-disambiguation frequency across all language models. In each one of the models, disambiguating the nouns is more accurate than verbs, when the embedding model is BERT. The coverage of verb senses can

Table 2: The Number of instances and ambiguity level of the concatenation of all five WSD datasets [27].

	Nouns	Verbs	Adj.	Adv	All
#Entities	4300	1652	955	346	7253
Ambiguity	4.8	10.4	3.8	3.1	5.8

explain this disambiguation performance difference between verbs and the other three parts of speech in WordNet, significantly less than the coverage of noun senses. To be more specific with our quantitative POS analysis, we tried to find the type of words in all datasets with more errors when disambiguating with different representations. We evaluate the effectiveness of our representation on parts of speeches, in comparison with the recent methods. The parts of speech that we have in the dataset are nouns, verbs, adjectives, and adverbs. Table 2 shows the number of instances in each category. In our second evaluation, we examined the effect of our representation against previous ones on each word category. Table 3 represents the F-Measure performance of the 1-NN WSD of each one of the contextualized word embeddings which we considered on All datasets split by parts of speech.

Table 3: F-Measure performance of the 1-NN WSD of each embedding on the standard WSD dataset split by parts of speech. The dataset in this experiment is a concatenation of all five datasets, which is split by Part-of-Speech tags.

Model	Nouns	Verbs	Adjectives	Adverbs
BERT	76.2±0.2	62.9±0.5	79.7±0.2	85.5±0.5
LMMS	78.2±0.6	64.1±0.3	81.3±0.1	82.9±0.3
SensEmBERT	77.8±0.3	63.4±0.5	80.1±0.4	86.4±0.2
ARES	78.7±0.1	67.3±0.2	82.6±0.3	87.1±0.4
our model	79.6±0.2	69.6±0.1	85.2±0.1	89.3±0.5

6 Conclusion

In this paper, we consider the problem of text ambiguity and one of its important factors, the word representation. We evaluate the pros and cons of current state-of-the-art approaches for word embedding, and applied them in parts of speeches on the standard datasets. By observing the opportunities to improve a word embedding model,

we present a novel approach for creating word embeddings. In our model, we consider the knowledge base and the context of the input document text, when generating the representation. We showed that this context-rich representation is beneficial for lexical ambiguity in English. The results of experiments in the WSD task show the efficiency of our representations compared to other state-of-the-art methods, despite relying only on English data. We further tested our embeddings on the split data into four parts of speeches. As the results of our second experiment show, the effectiveness of the contextualized embeddings in WSD on verbs is not as good as on nous. This defect is because of the lack of instances in the dataset in each word category. As future work, one point to improve our representations in the text ambiguity task is by training the model with data including more verbs than the current one.

References

1. Agirre, E., de Lacalle, O.L., Soroa, A.: Random walks for knowledge-based word sense disambiguation. Computational Linguistics **40**(1), 57–84 (Mar 2014), https://direct.mit.edu/coli/article/40/1/57/145
2. Aleksandrova, D., Drouin, P., Lareau, F. c c.o., Venant, A.: The multilingual automatic detection of 'e nonc é s bias 'e s in wikip é dia. ACL (2020), https://www.aclweb.org/anthology/R19-1006.pdf
3. Azad, H.K., Deepak, A.: A new approach for query expansion using wikipedia and wordnet. Information sciences **492**, 147–163 (2019), https://www.sciencedirect.com/science/article/pii/S0020025519303263
4. Calvo, H., Rocha-Ramírez, A.P., Moreno-Armendáriz, M.A., Duchanoy, C.A.: Toward universal word sense disambiguation using deep neural networks. IEEE Access **7**, 60264–60275 (2019), https://ieeexplore.ieee.org/abstract/document/8706934
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In North American Association for Computational Linguistics (NAACL) (2018), https://www.aclweb.org/anthology/N19-1423/
6. Edmonds, P., Cotton, S.: SENSEVAL-2: Overview. In: Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems. pp. 1–5. Association for Computational Linguistics, Toulouse, France (Jul 2001), https://www.aclweb.org/anthology/S01-1001.pdf
7. Fogarolli, A.: Word sense disambiguation based on wikipedia link structure. In: 2009 IEEE International Conference on Semantic Computing. pp. 77–82. IEEE (2009), https://ieeexplore.ieee.org/stamp/stamp.jsp
8. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
9. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: AAAI. vol. 32, pp. 234–242. Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans (2018)
10. Loureiro, D., Jorge, A.: Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.Florence, Italy. p. 5682–5691 (2019), https://www.aclweb.org/anthology/P19-1569
11. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I.: Information extraction meets the semantic web: a survey. Semantic Web **Preprint**, 1–81 (2020), http://repositorio.uchile.cl/bitstream/handle/2250/174484/Information-extraction-meets-the-Semantic-Web.pdf?sequence=1

12. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. ICLR **4**, 321–329 (2013), https://arxiv.org/pdf/1301.3781.pdf
13. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to wordnet: An on-line lexical database. International journal of lexicography **3**(4), 235–244 (1990), https://watermark.silverchair.com/235.pdf
14. Miller, G.A., Leacock, C., Tengi, R., Bunker, R.T.: A semantic concordance. In: Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993 (1993), https://www.aclweb.org/anthology/H93-1061/
15. Moro, A., Navigli, R.: SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In: SEM. pp. 288–297. Association for Computational Linguistics, Denver, Colorado (Jun 2015), https://www.aclweb.org/anthology/S15-2049.pdf
16. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics **2**, 231–244 (2014), https://watermark.silverchair.com/tacl_a_00179.pdf
17. Navigli, R.: Word sense disambiguation: A survey. ACM computing surveys (CSUR) **41**(2), 1–69 (2009), https://dl.acm.org/doi/abs/10.1145/1459352.1459355
18. Navigli, R., Jurgens, D., Vannella, D.: SemEval-2013 task 12: Multilingual word sense disambiguation. In: SEM. Association for Computational Linguistics, Atlanta, Georgia, USA (Jun 2013), https://www.aclweb.org/anthology/S13-2040.pdf
19. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial intelligence **193**, 217–250 (2012), https://www.sciencedirect.com/science/article/pii/S0004370212000793
20. Pasini, T., Elia, F.M., Navigli, R.: Huge automatically extracted training sets for multilingual word sense disambiguation. arXiv preprint arXiv:1805.04685 (2018), https://arxiv.org/abs/1805.04685
21. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing. pp. 1532–1543. EMNLP, Qatar (2014), https://www.aclweb.org/anthology/D14-1162.pdf
22. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. Association for Computational Linguistics pp. 2227–2237 (2018), https://www.aclweb.org/anthology/N18-1202
23. Peters, M.E., Logan IV, R.L., Schwartz, R., Joshi, V., Singh, S., Smith, N.A.: Knowledge enhanced contextual word representations. arXiv preprint arXiv:1909.04164 (2019), https://arxiv.org/pdf/1909.04164.pdf
24. Peters, M.E., Neumann, M., Zettlemoyer, L., Yih, W.t.: Dissecting contextual word embeddings: Architecture and representation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing p. 1499–1509 (2018), https://www.aclweb.org/anthology/D18-1179/
25. Pilehvar, M.T., Jurgens, D., Navigli, R.: Align, disambiguate and walk: A unified approach for measuring semantic similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1341–1351 (2013), https://www.aclweb.org/anthology/P13-1132.pdf
26. Pradhan, S., Loper, E., Dligach, D., Palmer, M.: SemEval-2007 task-17: English lexical sample, SRL and all words. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). pp. 87–92. Association for Computational Linguistics, Prague, Czech Republic (Jun 2007), https://www.aclweb.org/anthology/S07-1016
27. Raganato, A., Camacho-Collados, J., Navigli, R.: Word sense disambiguation: A unified evaluation framework and empirical comparison. In: Proceedings of the 15th Conference of

the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. pp. 99–110 (2017), https://www.aclweb.org/anthology/E17-1010/
28. Reisinger, J., Mooney, R.: 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. pp. 109–117 (2010), https://www.aclweb.org/anthology/N10-1013.pdf
29. Saeidi, M., Kosmajac, D., Taylor, S.: Dnlp@ fintoc'20: Table of contents detection in financial documents. In: Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation. pp. 169–173 (2020)
30. Saeidi, M., Milios, E., Zeh, N.: Contextualized knowledge base sense embeddings in word sense disambiguation. In: International Conference on Document Analysis and Recognition. pp. 174–186. Springer (2021)
31. Saeidi, M., Milios, E., Zeh, N.: Graph representation learning in document wikification. In: International Conference on Document Analysis and Recognition. pp. 509–524. Springer (2021)
32. Saeidi, M., Sousa, S.B.d.S., Milios, E., Zeh, N., Berton, L.: Categorizing online harassment on twitter. In: Joint European Conference on Machine Learning and KDD. pp. 283–297. Springer (2019), https://link.springer.com/chapter/10.1007/978-3-030-43887-6_22
33. Scarlini, B., Pasini, T., Navigli, R.: Just "onesec" for producing multilingual sense-annotated data. In: Proceedings of ACL. pp. 699–709 (2019), https://www.aclweb.org/anthology/P19-1069.pdf
34. Scarlini, B., Pasini, T., Navigli, R.: Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 8758–8765 (2020), https://ojs.aaai.org//index.php/AAAI/article/view/6402
35. Snyder, B., Palmer, M.: The English all-words task. In: Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. pp. 41–43. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://www.aclweb.org/anthology/W04-0811
36. Weikum, G., Dong, L., Razniewski, S., Suchanek, F.: Machine knowledge: Creation and curation of comprehensive knowledge bases. arXiv preprint arXiv:2009.11564 (2020), https://arxiv.org/pdf/2009.11564.pdf
37. West, R., Paranjape, A., Leskovec, J.: Mining missing hyperlinks from human navigation traces: A case study of wikipedia. In: Proceedings of the 24th international conference on World Wide Web, pp. 1242–1252 (2015), https://dl.acm.org/doi/pdf/10.1145/2736277.2741666
38. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. Curran Associates, Inc. **32**, 221–229 (2019), https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf
39. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7370–7377. AAAI, Honolulu (2019)
40. Zhao, G., Wu, J., Wang, D., Li, T.: Entity disambiguation to wikipedia using collective ranking. Information Processing & Management **52**(6), 1247–1257 (2016), https://www.sciencedirect.com/science/article/pii/S0306457316301893