The AAAI 2023 Workshop on Representation Learning for Responsible Human-Centric AI (R^2HCAI)

Analyzing Contextualized Knowledge Base Aware Representation Learning models for Word Sense Disambiguation

Mozhgan Saeidi[1] [2] **Evangelos Milios**[1] **Norbert Zeh**[1]

[1] Dalhousie University [2] Vector Institute
mozhgan.saeidi@vectorinstitute.ai, eem@dal.ca, nzeh@dal.ca

Abstract

Representation learning is crucial in solving most Natural Language Processing (NLP) problems, including Word Sense Disambiguation (WSD). The WSD task tries to find the best meaning in a knowledge base for a word with multiple meanings (ambiguous word). WSD methods choose this best meaning based on the context, i.e., the words around the ambiguous word in the input text document. Word representations may improve the effectiveness of the disambiguation models if they carry helpful information from the context and the knowledge base. In this paper, first, we provide an in-depth quantitative and qualitative analysis of existing transformer-based language models to understand their capabilities and potential limitations in encoding and recovering word senses. Second, we present a novel contextual-knowledge base aware sense representation method. The novelty in our representation is the integration of the knowledge base and the context. This representation lies in a space comparable to contextualized word vectors, thus allowing a word occurrence to be easily linked to its meaning by applying a simple nearest-neighbor approach. Finally, we compare our approach with state-of-the-art embedding methods for WSD.

Introduction

One important factor in Natural Language Processing (NLP) is representation learning, which plays a significant role in different tasks. Representation learning shows its importance by caring for information from different sources, like information from the context of the text (Navigli 2009; Saeidi et al. 2019). This hidden information is beneficial when solving some NLP tasks that rely on the context (Kosmajac, Taylor, and Saeidi 2020). One of these tasks is Word Sense Disambiguation (WSD). In this task, the input text includes words that have multiple possible meanings, and the goal is to find the best meaning based on the context. A word with multiple possible meanings is called an ambiguous word. The possible meanings are selected from a knowledge base (KB), like Wikipedia, known as predefined sense inventory. The context here refers to the input text in which an ambiguous word is.

The text ambiguity problem arises from the difficulty of picking the correct meaning for a word with multiple meanings. This task is easy for a human, especially when one considers the surrounding words, and the human reader identifies the correct meaning of each word based on the context in which the word is used. Computational methods try to mimic this approach (Ferreira, Pimentel, and Cristo 2018). Furthermore, these methods often represent their output by linking each word occurrence to an explicit representation of the chosen sense (West, Paranjape, and Leskovec 2015). We can divide the approaches to tackle this problem into the machine learning-based approach and the knowledge-based approach (Scarlini, Pasini, and Navigli 2020b). In the machine learning-based approach, systems are trained to perform the task (Saeidi et al. 2019). On the other hand, the knowledge-based approach requires external lexical resources such as Wikipedia, WordNet (Miller et al. 1990), a dictionary, or a thesaurus.

Some techniques to represent words as vectors (word embedding) include tf-idf (Ramos et al. 2003), and word2vec (Mikolov et al. 2013). Recently, embeddings based on pre-trained language models have attracted much interest. These recent models have shown promising results compared to classical embeddings for several NLP tasks. WSD is a task that achieves better results using the current pre-trained language models (Chronopoulou, Baziotis, and Potamianos 2019). These models include, e.g., ELMO (Peters et al. 2018a), BERT (Devlin et al. 2018), and XLNET (Yang et al. 2019), which encode several pieces of linguistic information in their word representations. These representations differ from static neural word embeddings (Pennington, Socher, and Manning 2014) in that they depend on the surrounding context of the word. This difference makes these vector representations especially interesting for the WSD task, where effective contextual representations can be highly beneficial for resolving lexical ambiguity. These representations enabled sense-annotated corpora to be exploited more efficiently (Loureiro and Jorge 2019). While recent works on word sense disambiguation using language models and contextualized embeddings show success, few studies analyze their effective behavior in lexical ambiguity regarding correct disambiguation. To our knowledge, the most recent work on this topic is by (Loureiro et al. 2021). This work considers the ability of some of the current contextualized models to support WSD.

One other important factor in the text ambiguity problem is the knowledge base. Knowledge bases are different in nature (Aleksandrova and Drouin 2020); 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 (Azad and Deepak 2019). Wikipedia is a hyperlink-based graph between encyclopedia entries. In WSD, with Wikipedia as the KB, we face the problem of link ambiguity between Wikipedia pages, meaning a phrase can usually be linked to more than one Wikipedia page in which the correct link depends on the context where it occurs. For example, the word "bar" can be linked to different articles, depending on whether it is used in a business or musical context.

This study overviews current text embedding approaches, focusing on the contextualized sense representation models. We consider a wide range of contextualized language models and evaluate their ability to capture lexical ambiguity in English. This evaluation is done from two perspectives. First, by analyzing the results of each layer of the language model, we discover which part of the model works better for disambiguation.Second, we identify which parts of speech are most commonly disambiguated incorrectly by different methods. Finally, we focus on the internal representation of the layers for modeling the errors of word categories. The goal is to find a useful word embedding representation for WSD and find the defects of these current approaches. Understanding the pros and cons of current representation learning methods helps us build a more effective embedding as our novel contribution. We use two different metrics to evaluate the performance of these embeddings in the context of disambiguation. We also provide an overview of the disambiguation methods and the most used ones in the literature. Our novel contribution provides a new representation of learning using the context of the input text and the context of the knowledge base. Using our vector representations, we applied the nearest neighbor heuristic algorithm to disambiguate ambiguous words. We finally compare the performance of our representations with the most current methods in solving the word sense disambiguation task and show our approach's efficiency.

Related Work

In this section, we first overview related works for the WSD task and then provide an overview of previous works toward analyzing pre-trained language models for lexical ambiguity. The WSD task is at the core of lexical semantics and has been tackled with various approaches. We divide these approaches into two categories of knowledge-based and supervised approaches (Navigli 2009).

Knowledge-Based Approaches

Knowledge-based methods use the semantic network structure, e.g., Wikipedia (Fogarolli 2009), WordNet (Miller et al. 1990), or BabelNet (Navigli and Ponzetto 2012), to find the correct meaning based on its context for each input word (Moro, Raganato, and Navigli 2014). In addition, these approaches employ algorithms on graphs to address the word ambiguity in texts (Agirre, de Lacalle, and Soroa 2014). Disambiguation based on Wikipedia has been demonstrated to be comparable in terms of coverage to domain-specific ontology (Weikum et al. 2020) since it has broad coverage, with documents about entities in various forms domains (Martinez-Rodriguez, Hogan, and Lopez-Arevalo 2020). 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 (Scarlini, Pasini, and Navigli 2020a). One of the latest works in this series is SensEmBERT (Scarlini, Pasini, and Navigli 2020a) which shows the power of language models combined with a vast amount of knowledge in a semantic network to produce latent semantic representations of nominal senses in multiple languages. ARES followed this model and created sense embeddings for the lexical meanings within a lexical knowledge base. These embeddings lie in a space comparable to that of contextualized word vectors (Scarlini, Pasini, and Navigli 2020b).

Supervised Approaches

Supervised approaches use sense-annotated data for their training. These approaches surpass the knowledge-based ones in all English data sets, even before introducing pre-trained language models. These approaches use neural architectures (Melamud, Goldberger, and Dagan 2016), or SVM models (Iacobacci, Pilehvar, and Navigli 2016), while still suffering from the need to create large manually-curated corpora (knowledge acquisition bottleneck), which reduces their usability to scale over unseen words (Gale, Church, and Yarowsky 1992). Automatic data augmentation approaches (Scarlini, Pasini, and Navigli 2019) developed methods to cover more words, senses, and languages.

Neural sequence models are trained for end-to-end WSD by (Raganato, Bovi, and Navigli 2017). They re-framed WSD as a translation task in that sequences of words are translated into sequences of senses. Later, some works showed the potential of contextual representation for WSD (Melamud, Goldberger, and Dagan 2016; Peters et al. 2018a). Sense embeddings initialization using glosses and adapted the skip-gram objective of word2vec is done by (Chen, Liu, and Sun 2014) to learn and improve the sense embeddings jointly with word embeddings. Later, by the appearance of NASARI vectors (Camacho-Collados and Pilehvar 2018), sense embeddings were created using structural knowledge from a large multilingual semantic network. These methods represent sense embeddings in the same space as the pre-trained word embeddings, while they suffer from fixed embedding spaces. Finally, the LMMS representation considers creating sense-level embeddings with complete coverage of WordNet. It shows the power of this representation for WSD by applying a simple Nearest Neighbors (k-NN) method (Loureiro and Jorge 2019). ARES used this 1-NN method with its representations and showed improved results in the WSD task.

Language Modelling Representation

Most NLP tasks now use semantic representations derived from language models. There are static word embeddings and

contextual embeddings.

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) (Mikolov et al. 2013). In both, the target word is at the center, and the context is considered a fixed-length window that slides over tokenized text. As a result, these models produce dense word representations. However, one limit for word embeddings means conflict around word types. This limitation affects the capability of these word embeddings for those sensitive to their context (Reisinger and Mooney 2010).

Contextual Word Embeddings The problem mentioned as a limitation for static word embeddings is solved in this type of embedding. The critical difference is that the contextual embeddings are sensitive to the context. Therefore, it allows the same word types to have different representations according to their context. The first work in contextual embeddings is ELMO (Peters et al. 2018a), followed by BERT (Devlin et al. 2018), as the state-of-the-art model. The critical feature of BERT, which makes it different, is the quality of its representations. Its results are task-specific fine-tuning of pre-trained neural language models. The recent representations in which we analyze their effectiveness are based on these two models (Peters et al. 2018b, 2019).

Transformer-based language models are pretty new in the NLP field, but there are a few works for analyzing these models and understanding the structure behind them (Liu et al. 2019; Loureiro et al. 2021). The transformer-based models have been shown to capture the syntax and be applicable for solving the NLP problems (Goldberg 2019; Saphra and Lopez 2018). (Jawahar, Sagot, and Seddah 2019) offers a phrasal representation analysis from BERT captured with the lower layers. It is also shown that transformer-based models encode well the human-like parse trees (Hewitt and Manning 2019). Quantitative analysis of contextualized word embeddings and sentence embedding models has demonstrated the effectiveness of the models' analysis of the semantic roles (Peters et al. 2018a; Conneau et al. 2018). The role of models for encoding sentence structure across a range of syntactic, semantic, local, and long-range phenomena is examined by (Tenney et al. 2019) and shows the strength of representations for syntactic phenomena. The entity type exploration and their relations are described in (Tenney et al. 2019). The effectiveness of LSTM language models has been shown (Linzen, Dupoux, and Goldberg 2016; Kuncoro et al. 2018), as well as understanding their internal representations for predicting words in a context (Van Schijndel, Mueller, and Linzen 2019). Furthermore, the LSTM predictions for a word in context provide the ability to retrieve substitutes, showing how well the language model has captured the information (Amrami and Goldberg 2018). Finally, for this LSTM-based contextualized embedding model, some analyses show how well these models distinguish between usages of words in context (Aina, Gulordava, and Boleda 2019; Soler et al. 2019).

In terms of a complete overview of neural network approaches and study of the BERT model, there are some complete recent surveys (Belinkov and Glass 2019; Rogers, Kovaleva, and Rumshisky 2020). The geometry of BERT is quantified in (Reif et al. 2019), which shows how this model cares about the neighboring tokens. A few studies try to use knowledge resources and extract semantic information to enhance the generalization of pre-trained language models like BERT (Peters et al. 2019; Levine et al. 2019). Characterizing the sense representation of BERT using cluster analysis has also been studied (Chronis and Erk 2020). The study on BERT's layers by (Reif et al. 2019) shows how this model performs for sense representations. The layer-wise performance of BERT when applied to the WSD task was studied in (Loureiro et al. 2021). The difference of our research is to quantitatively understand to what extent the pre-trained language models encode information for the lexical ambiguity in terms of different word types. We show these pre-trained contextualized sense embeddings' behavior when solving the ambiguousness of part of speech in the text.

Analysis of Language Models for WSD

For analyzing the current language models for WSD, we consider the analysis in two quantitative ways. First, to analyze the disambiguation performance by extracting embeddings from different layers of the language model. Second, to analyze the performance of each system in parts of speech.

Analysis by Layer Performance

In our quantitative analysis, we analyze the performance of the layer's representation in each model for each word in part of speech, i.e., nouns, verbs, adjectives, and adverbs. In this analysis, we are looking to find which layer produces a more effective representation for the WSD task. The results of a previous work (Reif et al. 2019) show the importance of the intermediate layers of BERT for sense representation, which was continued in (Loureiro et al. 2021) by separating layers. These recent contextualized embedding approaches use the sum of the corresponding representation from the final four layers of the BERT model by employing a 1NN strategy. The considered models, i.e., LMMS, SensEmBERT, and ARES, are BERT-based 1NN WSD methods, and they apply a pooling procedure to combine representations extracted from various layers of the model. Because of this reason, we focus on extracted representation results of each layer of BERT, i.e., which layer produces a more effective representation for the WSD task. Following the settings of previous work in this analysis (Loureiro et al. 2021), sense representation for each layer individually obtained for this analysis is learned from SemCor (Miller et al. 1993a). We show the performance of the layers in Figure 1. As the figure demonstrates, this experiment shows the effectiveness of the upper layers over the lower layers. Even for the verbs – which we show later that are the most challenging part of speech to disambiguate –, the layer that sums all the representations assigns the correct senses better than the individual layers. In more detail, the BERT representation achieves an overall F1 of 85.5 on adverbs, 79.7 on adjectives, 76.2 on nouns, and 62.9 on verbs. This analysis confirms the previous work and shows that the current convention of using the sum of the last four layers for sense representations is sensible, even if not optimal.

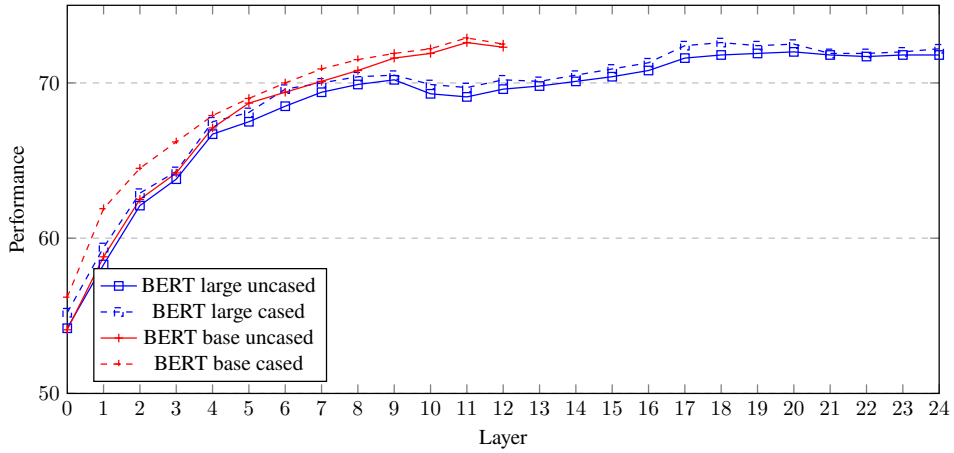

Figure 1: F-measure performance on the English WSD test set for representations extracted from each individual layer of BERT base and BERT large.

Analysis by Part-of-Speech

Another possible way to analyze the representation models in WSD is to measure the frequency of mis-disambiguation in different parts of speech (POS). As Table 9 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, which confirms the result of previous work (Loureiro et al. 2021), when the embedding model is BERT. The coverage of verb senses can explain this disambiguation performance difference between verbs and the other three parts of speeches 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 we applied WSD with different representations. For this aim, we first introduce an *error rate* formula, and second, we use *Confusion-Erro table*. In the confusion-error table, we show the rate of confusing one specific part of speech with other parts of speech that each embedding approach makes in the disambiguation task.

The error rate in each word type is similar to (Majidi and Crane 2014) that compares the errors of dependency parsers. The types include the ones we have in the standard dataset. For the error rates, first, we introduce the *Freq* as the frequency of the times the type has been disambiguated. The second item is *Error Frequency (EF)* as the model's performance in type disambiguation, the proportion of the times each type is disambiguated to the total number of disambiguated entities. Third, we introduce *Error Rate (ER)*, which is the number of times a type is mis-disambiguated to the frequency of that type, with formula 1, and we report this measure in our results. The last item is *Weighted Error Rate (WER)*, as formula 2. The main reason for introducing this variable is that more available word types provide more opportunities for learning them, so errors of that type should cost more. Therefore, we assigned a weight to the error rate by the error frequency of word appearance.

Type	Freq	EF	#Mis-D	ER	WER
Noun	4300	78.7	916	0.21	16.53
Verb	1652	67.3	540	0.33	22.21
Adj.	955	82.6	166	0.17	14.04
Adv.	346	87.1	45	0.13	11.32

Table 1: Error rate analysis of the 1-NN WSD evaluation framework with ARES representations on the All dataset, separated by type.

$$\text{Error Rate (e)} = \frac{\#\text{ type e mis-disambiguated}}{\text{Frequency of type e}} \quad (1)$$

$$WER\,(e) = EF\,(e) \times ER\,(e) \quad (2)$$

We report this analysis on BERT and ARES since BERT is the core language model used in all of the considered methods, and ARES shows the best results in Table 7, and compare these two models with our representations. We argue that errors made by these representations are of the same type. In Table 10, we observe the rate of errors in each type and the number of times each type is mis-disambiguated (Mis-D). The main observation extracted from Table 10 is that the embedding representations generated from ARES are not capturing the correct meaning of verb senses than other types. This result is similar when the embeddings are extracted from BERT, as Table 2 shows. While the ARES representation is a contextualized embedding, it fails 67.3% of times to address the ambiguity of verbs.

The other question about this quantitative analysis is whether the models made the same type of errors when they failed to disambiguate a word correctly. In the disambiguation task, some errors arose when the word type is mistaken; i.e.,.

Type	Freq	EF	#Mis-D	ER	WER
Noun	4300	76.2	1023	0.24	18.29
Verb	1652	62.9	613	0.37	23.27
Adj.	955	79.7	194	0.20	15.94
Adv.	346	85.5	50	0.14	11.97

Table 2: Error rate analysis of the 1-NN WSD evaluation framework with BERT representations on the All dataset, separated by type.

"close" is a noun in the sentence, but it is mistaken as a verb or adjective by the model. We show this error using a confusion-error table for all four parts of speech types, only reported for nouns because of space. The confusion-error of nouns disambiguation with other types in all datasets is shown in Table 3. For example, when BERT disambiguates nouns in this table, this model .75% of times is correct and considers the type as a noun. However, this model gets confused with the noun type with verb type for 0.11% of the time. The noun type with the adjective type gets confused 0.1% of the time. BERT model gets confused when disambiguating noun type instead of adverb for 0.04% of times. These results show that nouns get confused with verbs and adjectives for all models. After verbs, the adjective is confusing for almost all models when disambiguating the nouns. Among nouns, adjectives, and adverbs, the noun is the part of speech that gets confused instead of the verb in this task, almost for all the models. The results of the part of speech tag confusion analysis for verbs are shown in Table 4. ARES model is less confusing when disambiguating the type verb compared to other models.

We show how the models get confused when disambiguating type adverbs with other types in Table 5. This table demonstrates that LMMS and ARES both effectively disambiguate the adjectives to the right type. At the same time, SensEmBERT and BERT are two models that make mistakes when disambiguating the adjective that confuses it with nouns. The last type is an adverb, and the analysis of this type is shown in Table 6. These results indicate that ARES is the model with less confusion at disambiguating adverbs than other models. The SensEmBERT is the next model whose performance is near ARES regarding disambiguation confusion of adverbs. On the other hand, LMMS makes the mistake of considering verbs instead of adverbs. BERT is in the next place after LMMS of confusing verbs and adverbs.

Model	Noun	Verb	Adj.	Adv.
BERT	0.75	0.11	0.1	0.04
LMMS	0.80	0.1	0.07	0.03
SensEmBERT	0.79	0.1	0.09	0.02
ARES	0.81	0.09	0.07	0.03
C-KASE	0.84	0.085	0.06	0.01

Table 3: Confusion-Error table for Noun type by each model. This table shows how models are confused by the type of word at the time of disambiguation.

Model	Noun	Verb	Adj.	Adv.
BERT	0.26	0.64	0.06	0.04
LMMS	0.26	0.66	0.04	0.04
SensEmBERT	0.27	0.65	0.06	0.02
ARES	0.24	0.69	0.06	0.01

Table 4: Confusion table for Verb. This table shows how models are confused by the type of word at the time of disambiguation.

Model	Noun	Verb	Adj.	Adv.
BERT	0.1	0.06	0.82	0.02
LMMS	0.09	0.04	0.84	0.03
SensEmBERT	0.09	0.06	0.83	0.02
ARES	0.1	0.05	0.84	0.01

Table 5: Confusion table for Adjective.

New Representation Learning Model

This section presents Contextualized-Knowledge base Aware Sense Embedding (C-KASE), the novel contextualized-knowledge-based approach to creating sense representations. Based on the observations from the analysis of current embedding approaches, we develop this new representation model, reducing the cons of current models. C-KASE is created by combining semantic and textual information from the first paragraph of each sense's Wikipedia page and the paragraph of the input document text, which includes the senses. C-KASE uses the representation power of neural language models, i.e., BERT and SBERT. The other preliminaries creating C-KASE are Wikipedia, BabelNet, and Wordnet. C-KASE is based on three components; Context Retrieval, Word Embedding, and Sense Embedding, Figure 2.

The first component of C-KASE aims to collect contextual information from the knowledge base, which enhances the representations. For each ambiguous word in the input text, we create a set including candidate senses for the word from Wikipedia. This procedure aims to collect suitable contextual information from Wikipedia for each given concept in the semantic network. Then we exploit the mapping between synsets and Wikipedia pages available in BabelNet and its taxonomic structure to collect textual information 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:

$$R_s = \{s' | (s, s') \in E\} \quad (3)$$

E is the set including all hyponyms and hypernyms connections. In this work, for each page p_s, we consider the first

Model	Noun	Verb	Adj.	Adv.
BERT	0.05	0.08	0.01	0.86
LMMS	0.04	0.11	0.01	0.84
SensEmBERT	0.04	0.06	0.01	0.89
ARES	0.03	0.06	0.01	0.9

Table 6: Confusion table for Adverb.

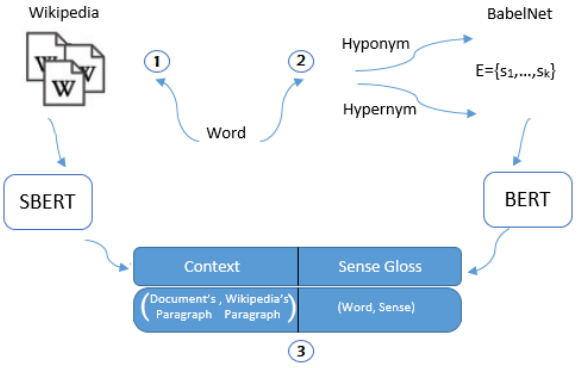

Figure 2: Demonstration of the C-KASE representation and its three components. Component 1) Collecting all Wikipedia pages for ambiguous words, Component 2) Using hypernymy and hyponymy relations to extract all synsets for ambiguous words from Babelnet in set E, Component 3) Concatenating (word,sense) representation for all senses in E from the second component with (Document's paragraph, Wikipedia's paragraph) representation from the first component as context.

opening paragraph of the page and compute its lexical vector by taking the mean of 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 (Pilehvar, Jurgens, and Navigli 2013), which is defined as follows:

$$WO(p_1, p_2) = (\sum_{w \in O} \frac{1}{r_w^{p_1} + r_w^{p_2}})(\sum_{i=1}^{|O|} \frac{1}{2i})^{-1} \quad (4)$$

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 (Pilehvar, Jurgens, and Navigli 2013). 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 $WO(p_{s_i}, p_{s'_i})$, which we set $k = 15$ in our experiments.

In the second component, we use BERT to extract the given ambiguous word from the input text. We extract its BERT representation for each ambiguous word (mention) of the input. Using the BabelNet relations of hyponymy and hypernymy, we extract all synsets of mention from BabelNet (set E). Use the link structure of BabelNet and Wikipedia; we collect all the Wikipedia pages for each sense. Finally, we use BERT representation for the second time to generate vector representation for senses. Each word is represented in the settings as the BERT dimension.

In this last component, we build the final representation of each mention. 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. Our unique representations combine the mention representation with sense representation, concatenating the two vector representations of $R(m)$ and $R(s_i)$. If mention m has k senses, C-KASE generates k different representations of $R(m, s_1)$, $R(m, s_2)$, ..., $R(m, s_k)$. Because of the dimension representation of $R(m)$ and each $R(s_i)$, these concatenated representation dimensions are doubled. The next novelty in our C-KASE representations is ranking the k senses of each mention based on their relevancy degree to the context. To this aim, we concatenate representations of the first step. In the first step, we took the representation of the input text paragraph, which contains the ambiguous mention, and show it by $R(PD)$, which stands for representation of the **P**aragraph of the input **D**ocument text. 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 concatenate these two representations as $R(PD, PW)$. The dimension of this concatenated representation is also equal to the word representation, making it possible to calculate their cosine similarities. To rank the senses' relevancy to the context, we use the cosine similarity as follows:

$$\text{Sim}(m, s_i) = Cosine(R(m, s_i), R(PD, PW)), \quad \text{for } i = 1, \ldots, k \quad (5)$$

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 word sense disambiguation.

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

WSD Experimental Setup

We present the settings of our evaluation of C-KASE in the English WSD task. This setup includes the benchmark, C-KASE setup for disambiguation task, and state-of-the-art WSD models as our comparison systems. To test each embedding on the WSD task, we employed the 1-NN algorithm and compared the disambiguated sense of each word with the ground truth annotations in the datasets. The nearest neighbors strategy is effective with pre-trained language models (Loureiro et al. 2021; Scarlini, Pasini, and Navigli 2020b).

Evaluation Benchmark

We use the English WSD test set framework, which is constructed by five standard evaluation benchmark datasets[1]. It is included Senseval-2 (Edmonds and Cotton 2001), Senseval-3 (Snyder and Palmer 2004), SemEval-07 (Pradhan et al. 2007), SemEval-13 (Navigli, Jurgens, and Vannella 2013), SemEval-15 (Moro and Navigli 2015) along with ALL, i.e.,

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

the concatenation of all the test sets (Raganato, Camacho-Collados, and Navigli 2017). All these datasets are WordNet-specific and mostly use SemCor (Miller et al. 1993b) as their training set. The unified benchmark provides 7253 test instances for 4363 sense types, which covers 3663 word types across four parts of speech: nouns, verbs, adjectives, and adverbs. Details of the number of instances in each part of speech of all five datasets are presented in Table 8. The sense inventory that is used in this work is WordNet (version3.0), the main sense inventory for the WSD task in English (Fellbaum 1998).

C-KASE Setup

In our experiments, we use BERT pre-trained cased model. Among all the configurations reported by(Devlin et al. 2018), we used the sum of the last four hidden layers as contextual embeddings of the words since, in our analysis, in the error modeling section, we show it has better performance. In order to be able to compare our system with supervised models, we build a supervised version of our C-KASE representations. This version combines the gloss and contextual information with the sense-annotated contexts in SemCor (Miller et al. 1993b), a corpus of 40K sentences where words have been manually annotated with a WordNet meaning.

We leveraged SemCor to build a representation of each sense therein. To this end, we followed (Peters et al. 2018a), given a mention-sense pair (m, s), we collected all the sentences $c_1, \ldots, c_n$ where m appears tagged with s. Then, we fed all the retrieved sentences into BERT and extracted the embeddings BERT$(c_1, m), \ldots,$ BERT(c_n, m). The final embedding of s was built by concatenating the average of its context and sense gloss vectors and its representation coming from SemCor, i.e., the average of BERT$(c_1, m), \ldots,$ BERT(c_n, m). Finally, we note that when a sense did not appear in SemCor, 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 using BERT and compared it against the embeddings of C-KASE 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 C-KASE are created by concatenating two BERT representations, i.e., context and sense gloss (see Section Sense Embedding); 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 current performing systems evaluated on the English WSD task. LMMS is one of these systems which generates sense embedding with complete coverage of Wordnet. It uses pre-trained ELMO and BERT models and the relations in a lexical knowledge base to create contextual embeddings (Loureiro and Jorge 2019). 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 (Scarlini, Pasini, and Navigli 2020a). The next 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 many words' meanings. ARES is the most recent contextualized word embedding system, to our knowledge. We also considered BERT a comparison system since it is at the core of all the considered methods. BERT has also shown good performance in most NLP tasks using pre-trained neural networks.

Results

The results of our evaluations on the WSD task are represented in this section. We show the effectiveness of C-KASE representation by comparing it with the existing state-of-the-art models on the standard WSD benchmarks. In Table 7, we report the results of C-KASE and compare them against the results obtained from other state-of-the-art approaches on all the nominal instances of the test sets in the framework of (Raganato, Camacho-Collados, and Navigli 2017). All performances are reported in terms of F1-measure, i.e., the harmonic mean of precision and recall. As we can see, C-KASE achieves the best results on the datasets compared to other previous contextualized approaches. It indicates that C-KASE is competitive with these previous models. These results show that the novel idea, like creating this C-KASE representation, has improved the lexical ambiguity. It is a good indicator of the dependency of the WSD task on the representation that is aware of the context and the information extracted from the reference knowledge base.

We also evaluate the effectiveness of our representation of parts of speeches. The parts of speech in the dataset are nouns, verbs, adjectives, and adverbs. Table 8 shows the number of instances in each category. In our second evaluation, we examined the effect of our representation against previous models on each word category. Table 9 represents the F-Measure performance of the 1-NN WSD of the contextualized word embeddings which we considered on All datasets split by parts of speech.

Conclusion

In this paper, we present C-KASE, a novel approach for creating sense embeddings considering the knowledge base and the context of the input document text. We showed that this context-rich representation is beneficial for lexical ambiguity in English. The first set of WSD experiments demonstrates the effectiveness of C-KASE representations compared to other state-of-the-art methods, despite relying only on English data. We further report the WSD performance of our embeddings on each of the four parts of speech; nouns, verbs, adjectives,

and adverbs. The results across different datasets show the high quality of our embeddings in English WSD while simultaneously relaxing the requirement for sense-annotated corpora. The second set of WSD experiments uses a novel measure of the WSD error of each representation model and quantifies how likely it is for each model to confuse parts of speech at the time of disambiguation. Among the different parts of speech, we observed that the verb is the most challenging type to disambiguate since its instances in the training dataset are very few. Future research includes: covering multiple languages; training on data with more verbs.

References

Agirre, E.; de Lacalle, O. L.; and Soroa, A. 2014. Random Walks for Knowledge-Based Word Sense Disambiguation. *Computational Linguistics*, 40(1): 57–84.

Aina, L.; Gulordava, K.; and Boleda, G. 2019. Putting words in context: LSTM language models and lexical ambiguity. *ACL*, 3342–3348.

Aleksandrova, D.; and Drouin, P. 2020. The multilingual automatic detection of biases in Wikipedia. *ACL*.

Amrami, A.; and Goldberg, Y. 2018. Word sense induction with neural biLM and symmetric patterns. *In Proceeding of EMNLP*, 4860–4867.

Azad, H. K.; and Deepak, A. 2019. A new approach for query expansion using Wikipedia and WordNet. *Information sciences*, 492: 147–163.

Belinkov, Y.; and Glass, J. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7: 49–72.

Camacho-Collados, J.; and Pilehvar, M. T. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Artificial Intelligence Research*, 63: 743–788.

Chen, X.; Liu, Z.; and Sun, M. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference EMNLP*, 1025–1035.

Chronis, G.; and Erk, K. 2020. When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference CNLL*, 227–244.

Chronopoulou, A.; Baziotis, C.; and Potamianos, A. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. *arXiv preprint arXiv:1902.10547*.

Conneau, A.; Kruszewski, G.; Lample, G.; Barrault, L.; and Baroni, M. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *ACL*, 2126–2136.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Edmonds, P.; and Cotton, S. 2001. SENSEVAL-2: Overview. In *SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 1–5. Toulouse, France: ACL.

Fellbaum, C. 1998. Towards a Representation of Idioms in WordNet. In *Coling-ACL*, 52–57.

Ferreira, R. S.; Pimentel, M. d. G.; and Cristo, M. 2018. A wikification prediction model based on the combination of latent, dyadic, and monadic features. *IST*, 69(3): 380–394.

Fogarolli, A. 2009. Word sense disambiguation based on wikipedia link structure. In *2009 IEEE International Conference on Semantic Computing*, 77–82. IEEE.

Gale, W. A.; Church, K. W.; and Yarowsky, D. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5): 415–439.

Goldberg, Y. 2019. Assessing BERT's syntactic abilities. *CoRR, abs/1901.05287*.

Hewitt, J.; and Manning, C. D. 2019. A structural probe for finding syntax in word representations. In *ACL*, 4129–4138.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2016. Embeddings for word sense disambiguation: An evaluation study. In *ACL*, 897–907.

Jawahar, G.; Sagot, B.; and Seddah, D. 2019. What does BERT learn about the structure of language? In *ACL*.

Kosmajac, D.; Taylor, S.; and Saeidi, M. 2020. 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*, 169–173.

Kuncoro, A.; Dyer, C.; Hale, J.; Yogatama, D.; Clark, S.; and Blunsom, P. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *ACL*, 1426–1436.

Levine, Y.; Lenz, B.; Dagan, O.; Ram, O.; Padnos, D.; Sharir, O.; Shalev-Shwartz, S.; Shashua, A.; and Shoham, Y. 2019. Sensebert: Driving some sense into bert. *ACL*.

Linzen, T.; Dupoux, E.; and Goldberg, Y. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4: 521–535.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loureiro, D.; and Jorge, A. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. *ACL*, 5682–5691.

Loureiro, D.; Rezaee, K.; Pilehvar, M. T.; and Camacho-Collados, J. 2021. Analysis and Evaluation of Language Models for Word Sense Disambiguation. *Computational Linguistics*, 1–55.

Majidi, S.; and Crane, G. 2014. Human and machine error analysis on dependency parsing of ancient Greek texts. In *IEEE/ACM Joint Conference on DL*, 221–224. IEEE.

Martinez-Rodriguez, J. L.; Hogan, A.; and Lopez-Arevalo, I. 2020. Information extraction meets the semantic web: a survey. *Semantic Web*, Preprint: 1–81.

Melamud, O.; Goldberger, J.; and Dagan, I. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *SIGNLL*, 51–61.

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

Table 7: F-Measure performance of WSD evaluation framework on the test sets nominal instances of the unified dataset.

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

Table 8: The Number of instances and ambiguity level of the concatenation of all five WSD datasets (Raganato, Camacho-Collados, and Navigli 2017).

Model	Nouns	Verbs	Adjectives	Adverbs
BERT	76.2	62.9	79.7	85.5
LMMS	78.2	64.1	81.3	82.9
ARES	78.7	67.3	82.6	87.1
C-KASE	79.6	69.6	85.2	89.3

Table 9: F-Measure performance of the 1-NN WSD of each embedding on All dataset split by parts of speech.

	ARES		BERT		C-KASE	
Type	#Mis-D	ER	#Mis-D	ER	#Mis-D	ER
Noun	916	0.21	1023	0.24	877	0.20
Verb	540	0.33	613	0.37	502	0.30
Adj.	166	0.17	194	0.20	141	0.14
Adv.	45	0.13	50	0.14	37	0.10

Table 10: Error rate analysis of the 1-NN WSD evaluation framework with ARES, BERT, and C-KASE representations on the All dataset, separated by type.

Mikolov, T.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of ICLR*, 4: 321–329.

Miller, G. A.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. J. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4): 235–244.

Miller, G. A.; Leacock, C.; Tengi, R.; and Bunker, R. T. 1993a. A Semantic Concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Miller, G. A.; Leacock, C.; Tengi, R.; and Bunker, R. T. 1993b. A semantic concordance. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Moro, A.; and Navigli, R. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *SEM*, 288–297. Denver, Colorado: ACL.

Moro, A.; Raganato, A.; and Navigli, R. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2: 231–244.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2): 1–69.

Navigli, R.; Jurgens, D.; and Vannella, D. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *SEM*. Atlanta, Georgia, USA: ACL.

Navigli, R.; and Ponzetto, S. P. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence*, 193: 217–250.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543. Qatar: EMNLP.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018a. Deep Contextualized Word Representations. *ACL*, 2227–2237.

Peters, M. E.; Logan IV, R. L.; Schwartz, R.; Joshi, V.; Singh, S.; and Smith, N. A. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Peters, M. E.; Neumann, M.; Zettlemoyer, L.; and Yih, W.-t. 2018b. Dissecting contextual word embeddings: Architecture and representation. *EMNLP*, 1499–1509.

Pilehvar, M. T.; Jurgens, D.; and Navigli, R. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *ACL*, 1341–1351.

Pradhan, S.; Loper, E.; Dligach, D.; and Palmer, M. 2007. SemEval-2007 Task-17: English Lexical Sample, SRL and All Words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 87–92. Prague, Czech Republic: Association for Computational Linguistics.

Raganato, A.; Bovi, C. D.; and Navigli, R. 2017. Neural sequence learning models for word sense disambiguation. In *EMNLP*, 1156–1167.

Raganato, A.; Camacho-Collados, J.; and Navigli, R. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *EACL*, 99–110.

Ramos, J.; et al. 2003. Using tf-idf to determine word relevance in document queries. In *ML*, volume 242, 29–48. Citeseer.

Reif, E.; Yuan, A.; Wattenberg, M.; Viegas, F. B.; Coenen, A.; Pearce, A.; and Kim, B. 2019. Visualizing and measuring the geometry of BERT. *Advances in Neural Information Processing Systems*, 32: 8594–8603.

Reisinger, J.; and Mooney, R. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: ACL*, 109–117.

Rogers, A.; Kovaleva, O.; and Rumshisky, A. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8: 842–866.

Saeidi, M.; Sousa, S. B. d. S.; Milios, E.; Zeh, N.; and Berton, L. 2019. Categorizing online harassment on Twitter. In *Joint European Conference on Machine Learning and KDD*, 283–297. Springer.

Saphra, N.; and Lopez, A. 2018. Understanding learning dynamics of language models with svcca. *NAACL*.

Scarlini, B.; Pasini, T.; and Navigli, R. 2019. Just "OneSeC" for producing multilingual sense-annotated data. In *ACL*, 699–709.

Scarlini, B.; Pasini, T.; and Navigli, R. 2020a. SensEmBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *AAAI*, volume 34, 8758–8765.

Scarlini, B.; Pasini, T.; and Navigli, R. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round Word Sense Disambiguation. In *EMNLP*, 3528–3539.

Snyder, B.; and Palmer, M. 2004. The English all-words task. In *SENSEVAL-3*, 41–43. Barcelona, Spain: Association for Computational Linguistics.

Soler, A. G.; Cocos, A.; Apidianaki, M.; and Callison-Burch, C. 2019. A comparison of context-sensitive models for lexical substitution. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, 271–282.

Tenney, I.; Xia, P.; Chen, B.; Wang, A.; Poliak, A.; McCoy, R. T.; Kim, N.; Van Durme, B.; Bowman, S. R.; Das, D.; et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *In International Conference on Learning Representations*.

Van Schijndel, M.; Mueller, A.; and Linzen, T. 2019. Quantity doesn't buy quality syntax with neural language models. *EMNLP-IJCNLP*, 5831–5837.

Weikum, G.; Dong, L.; Razniewski, S.; and Suchanek, F. 2020. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. *arXiv preprint arXiv:2009.11564*.

West, R.; Paranjape, A.; and Leskovec, J. 2015. Mining missing hyperlinks from human navigation traces: A case study of Wikipedia. In *Proceedings of the 24th international conference on World Wide Web*, 1242–1252.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Curran Associates, Inc.*, 32: 221–229.