With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation

Bianca Scarlini, Tommaso Pasini and **Roberto Navigli**
Sapienza NLP Group
Department of Computer Science
Sapienza University of Rome
{scarlini,pasini,navigli}@di.uniroma1.it

Abstract

Contextualized word embeddings have been employed effectively across several tasks in Natural Language Processing, as they have proved to carry useful semantic information. However, it is still hard to link them to structured sources of knowledge. In this paper we present ARES (context-AwaRe Embeddings of Senses), a semi-supervised approach to producing sense embeddings for the lexical meanings within a lexical knowledge base that lie in a space that is comparable to that of contextualized word vectors. ARES representations enable a simple 1-Nearest-Neighbour algorithm to outperform state-of-the-art models, not only in the English Word Sense Disambiguation task, but also in the multilingual one, whilst training on sense-annotated data in English only. We further assess the quality of our embeddings in the Word-in-Context task, where, when used as an external source of knowledge, they consistently improve the performance of a neural model, leading it to compete with other more complex architectures. ARES embeddings for all WordNet concepts and the automatically-extracted contexts used for creating the sense representations are freely available at http://sensembert.org/ares.

1 Introduction

Contextualized word embeddings have proved to be highly beneficial to the majority of Natural Language Processing tasks (Wang et al., 2019). Indeed, language models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), etc., enable architectures built on top of them to attain performances that were previously out of reach (Wang et al., 2019). The main reason behind this great success is the fact that contextualized embeddings of words encode the semantics defined by their input context (Reif et al., 2019). Indeed, when tested in the Word-in-Context (WiC) task (Pilehvar and Camacho-Collados, 2019), i.e., a binary classification problem where a model has to classify whether a target word is used with the same meaning in two different sentences, contextualized word embeddings placed themselves as the best approaches across the board. Nevertheless, these latent representations do not provide any explicit information regarding the meaning expressed by the word in context, hence making it difficult to link texts to structured sources of knowledge such as lexical knowledge bases (LKB).

The task of associating a word in context with the most suitable meaning from a predefined sense inventory is better known as Word Sense Disambiguation (Navigli, 2009, WSD), and is usually tackled by two kinds of approach: knowledge-based and supervised ones. On the one hand, knowledge-based approaches (Scozzafava et al., 2020; Conia and Navigli, 2020) are able to scale across languages since they do not need sense-annotated corpora and rely only on the information within their underlying LKB. On the other hand, supervised models (Huang et al., 2019; Bevilacqua and Navigli, 2020) have proved to achieve state-of-the-art results on the English benchmarks by taking advantage of manually-annotated data for the task and machine learning algorithms. However, supervised approaches are mostly focused on English (Navigli, 2018; Pasini, 2020) and have only recently been applied to lower-resourced languages thanks to automatically-produced datasets (Scarlini et al., 2019; Barba et al., 2020; Pasini and Navigli, 2020). Another effective approach in this direction has been presented by Scarlini et al. (2020), who introduced SensEmBERT, a knowledge-based approach to building sense embeddings without relying on sense-annotated data. Since it is not tied to semantic annotations, SensEmBERT scales over different languages. However, it is limited to nominal concepts only and provides different

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3528–3539,
November 16–20, 2020.

representations for the same concepts across different languages, which hinders its applicability to cross-lingual tasks.

In this paper we present ARES (context-AwaRe Embeddings of Senses), a semi-supervised approach to producing sense embeddings for all the word senses in a language vocabulary. ARES makes up for the paucity of manually-annotated examples for a large portion of words' meanings by coupling the information within a knowledge base with the representational power of a pre-trained language model. This enables reliable representations to be built for those senses not appearing in manually-curated resources, while at the same time enriching the vectors for all the other concepts.

We tested our embeddings on the two tasks that measure a model's capabilities to encode word meanings, i.e., WSD and WiC. In both tasks, ARES representations prove to be of great benefit. In WSD, while employing a simple 1-Nearest-Neighbour (1-NN) algorithm, they attain state-of-the-art results on English, even beating dedicated architectures with long and expensive fine-tuning procedures. In WiC they lead a simple BERT-based model to perform in the same ballpark as other state-of-the-art alternatives which rely on more complex architectures. Furthermore, by taking advantage of pre-trained multilingual models we provide unified representations of meanings across languages, which, while using English data only, outperform their competitors and achieve the state of the art on all the languages available in the all-words multilingual WSD tasks, i.e., French, German, Italian and Spanish.

2 Related Work

Word Sense Disambiguation (WSD) is a core task in lexical semantics and has mainly been tackled by two kinds of approach: knowledge-based and supervised ones. Knowledge-based methods build upon lexical knowledge bases, such as WordNet (Miller et al., 1990) and BabelNet (Navigli and Ponzetto, 2012), and employ algorithms on graphs to address the word ambiguity in texts (Moro et al., 2014; Agirre et al., 2014; Tripodi and Navigli, 2019; Scozzafava et al., 2020). These approaches do not rely on semantically-tagged training data and are hence able to scale over all the languages supported by their underlying knowledge base. Nevertheless, they lag behind their supervised counterparts on English in terms of performance. Supervised approaches, by framing WSD as a classification task, have acquitted themselves as the state of the art in English (Hadiwinoto et al., 2019; Huang et al., 2019; Blevins and Zettlemoyer, 2020; Bevilacqua and Navigli, 2020; Bevilacqua et al., 2020), outperforming their knowledge-based competitors by several points.

Recently, Pilehvar and Camacho-Collados (2019) provided a new declination of WSD, formulating it as a binary classification problem where, given a target word and two contexts, a model has to predict whether the target word is used with the same meaning. This setting has the advantage of not drawing on sense inventories and provides an effective testbed for context-based word embeddings (Peters et al., 2019; Levine et al., 2020).

Contextualized sense representations have recently been employed to compute sense representations that can be applied directly to disambiguation. Some of the first approaches of this kind were proposed by Melamud et al. (2016) and Peters et al. (2018), who exploited the semantically-tagged sentences of SemCor (Miller et al., 1993) and neural language models to create embeddings for the senses in WordNet. Similarly, Loureiro and Jorge (2019, LMMS) computed sense embeddings using BERT (Devlin et al., 2019) and the relations in a lexical knowledge base in order to also provide vectors for those meanings that do not appear in SemCor. The most recent effort in this direction is SensEmBERT (Scarlini et al., 2020), which drops the need for sense-annotated corpora by exploiting the BabelNet mapping between WordNet senses and Wikipedia pages so as to collect contextual information for the senses in WordNet. Since it does not rely on manually-annotated data SensEmBERT can scale over different languages, being limited, however, to nominal senses only.

In this work we continue along this latter line of research and propose a novel method for producing sense embeddings which, by relying on English data only, also proves to be able to model meanings across languages. Rather than leveraging WordNet relations as LMMS does, ARES creates vector representations for all senses by automatically providing usage examples for the synsets within a knowledge base. In contrast to SensEmBERT, instead, ARES covers all the four WordNet POS tags, and, at the same time, disposes of the resources required by SensEmBERT, such as NASARI and the Wikipedia category graph.

3 Preliminaries

We now describe the resources that we use to build ARES embeddings.

WordNet (Miller et al., 1990) is the most used lexical knowledge base for English. It can be viewed as a graph where nodes are concepts, i.e., synsets, and edges are semantic relations between them. Each synset contains a set of synonyms, e.g., the synset defined as *A natural flow of ground water* comprises the lemmas *spring*, *fountain* and *natural spring*. We use the notation $\{l_1, \ldots, l_n\}$ (g) to refer to the concept with gloss g and expressed by the lemmas $l_1, \ldots, l_n$. We define a *sense* as a lemma-gloss pair, i.e., a meaning that is specific to a given lemma, e.g., *fountain*-(*A natural flow of ground water*) is a sense of {*spring, fountain, natural spring*} (*A natural flow of ground water*).

SyntagNet (Maru et al., 2019) is a repository containing approximately $88K$ lexical-semantic collocations, i.e., pairs of WordNet synsets that co-occur more frequently than would be expected.[1] For example, the concepts {*coach, bus, autobus*} (*A vehicle carrying many passengers*) and {*driver, motorist*} (*The operator of a motor vehicle*) appear in SyntagNet as they form a collocation.

UKB (Agirre et al., 2014) is a knowledge-based approach to WSD based on the Personalized PageRank algorithm (Haveliwala et al., 2002). We set WordNet as underlying knowledge base, disable the Most Frequent Sense backoff strategy and set the parameters according to Agirre et al. (2018).

SemCor (Miller et al., 1993) is the standard manually-curated corpus for WSD including more than $220K$ words tagged with $25K$ distinct WordNet meanings, hence providing annotated contexts for roughly 15% of the synsets in WordNet.

BERT (Devlin et al., 2019) is a deep neural architecture trained with the masked language model objective. Given a text, it provides contextual embeddings for the subtokens therein. We choose BERT because it has proven to capture the semantics of a word in context (Reif et al., 2019), while also being able to effectively generalize cross-lingually thanks to its multilingual representations (Pires et al., 2019).[2]

[1] http://syntagnet.org/

[2] We note that a comparison with other pre-trained language models is outside the scope of this paper and a more extensive evaluation is left as future work.

4 ARES

We now introduce ARES, a semi-supervised approach for creating sense embeddings that cover all the senses in a language vocabulary. Given as input a corpus C of raw sentences and a synset $s \in$ WordNet together with its lexicalizations L_s, ARES operates the following three steps:

1. **Context extraction,** which exploits the representation capabilities of BERT and the collocational information comprised in SyntagNet to extract a meaningful set of contexts where s is likely to appear (Section 4.1);

2. **Synset embedding,** which creates the embedding of the synset s by encoding the contextual information of the sentences gathered in the previous step (Section 4.2);

3. **Sense embedding,** which combines the sense-annotated contexts in SemCor, the definitional information of the glosses and our synset embeddings to create the final sense representation (Section 4.3).

4.1 Context Extraction

In this Section we describe our approach for automatically retrieving contexts for WordNet's synsets. First, as in Pasini et al. (2020), we utilize BERT and UKB to find contexts that are similar to each other and link them to a meaning in WordNet. Then, we enrich the set of contexts retrieved for a given synset s by exploiting the semantic collocations available in SyntagNet.

Similarity-Based Extraction Given a synset s and one of its lexicalizations l, we collect the occurrences of l in the input corpus C and compute their contextualized representations by means of BERT.[3] We then cluster the contextualized vectors of l's occurrences by using the *k-means* algorithm. We note that the sentences comprised within the same cluster define similar contexts for the target word, hence implying that l is very likely to be used with the same meaning across sentences. Therefore, we associate each cluster with one of l's meanings and a disambiguation score. To this end, we apply UKB (see Section 3) to the set of words that most characterize the given cluster, i.e., the top n most

[3] We discard all the sentences in which l is part of a larger span that is identified as a named entity.

Sentences for {*spring, fountain, natural spring*}
Springs that contain significant amounts of minerals are called 'mineral springs'.
The forcing of the **spring** to the surface can be the result of a confined aquifer.
Other **fountains** are the result of pressure from an underground source in the earth
Natural springs that contain significant amounts of minerals are called 'mineral springs'.
Other **natural springs** are the result of pressure from an underground source in the earth.

Table 1: Sentences retrieved for the synset {*spring, fountain, natural spring*} (upper part) and sentences where the target lemmas have been replaced with the missing ones (bottom part).

Sentences
He learned how to **play** the **guitar** at the age of eleven.
Michelle can **play** skillfully on **guitar** and piano.
The Ventures **played** Fender **guitars** for their live performances.

Table 2: Excerpt of sentences where the synsets {*play*} (*Play on an instrument*) and {*guitar*} (*A stringed instrument*) appear together.

frequent words[4] among its sentences.[5] Once each cluster has been disambiguated with one meaning of l, we retain only those clusters that are associated with s. Then, we associate each sentence with the disambiguation score provided by UKB for its cluster and sample t sentences according to their score, creating a set of contexts $\Phi_{l,s}$ for the lemma l in the synset s. We note that it might happen that none of the clusters of l is associated with s. This limits both the number and the diversity of contexts available for the target synset. To overcome this issue and increase coverage, we sample a set of ξ sentences from $\cup_{l' \in L_s} \Phi_{l',s}$ and replace the lexicalizations l' of s that appear therein with the lemma l. For example, let {*spring, fountain, natural spring*} (*A natural flow of ground water*) be the input synset, and the sentences in Table 1 (top) be the contexts retrieved thanks to the clustering and disambiguation steps, we replace some occurrences of *spring* and *fountain* with *natural spring*, as shown in the bottom part of the Table.

Collocation-Aware Extraction We now enrich the set $\Phi_{l,s}$ by leveraging the semantic collocations available in SyntagNet (see Section 3) for the synset s. To this end, we first retrieve from SyntagNet all the synsets s' that collocate with s, and then extract all the sentences in C where any of the lemmas l and l' of s and s', respectively, appear within a small window w. Finally, we disambiguate each occurrence of l with its synset s. For example, given the concepts {*play*} (*Play on an instrument*) and {*guitar*} (*A stringed instrument*) which are in collocation in SyntagNet, we search for all the occurrences of *play* and *guitar* in the sentences of the input corpus and retain only those where the two words appear within a window of size 3. In Table 2 we show an excerpt of the sentences extracted for the two aforementioned synsets. Each occurrence of *play* in those sentences is hence disambiguated with {*play*} (*Play on an instrument*).

At the end of this step, the synset s is associated with the set of sentences $\Phi_s = \cup_{l \in L_s} \Phi_{l,s}$ where any of the lemmas of s is disambiguated with s.

4.2 Synset Embedding

In this step we exploit the contexts retrieved for a target synset s in order to compute its latent representation.

First, we create the set $\hat{L}_s$ containing the lexicalizations of the synsets that are collocated with s in SyntagNet. For example, given the synset s = {*spring, fountain, natural spring*} (*A natural flow of ground water*), we consider the lexicalizations of its related concepts in SyntagNet, i.e., *flow* and *flowing* from the synset {*flow, flowing*} (*The motion characteristic of fluids*) and create $\hat{L}_s = \{flow, flowing\}$.

Then, we leverage the contexts in $\Phi_{l,s}$ and the lemmas in both L_s and $\hat{L}_s$ to compute the vector representation v_s for the synset s as follows:

$$v_{\mathrm{C}}^s = \frac{\sum_{l \in L_s} E(\Phi_{l,s}) + \sum_{l' \in \hat{L}_s} E(\hat{\Phi}_{l',s})}{Z} \quad (1)$$

$$E(\Phi_{\lambda,s}) = \sum_{\sigma \in \Phi_{\lambda,s}} \textsc{Bert}\,(\lambda, \sigma) \quad (2)$$

where $\hat{\Phi}_{l',s}$ is a subset of $\Phi_{l,s}$ containing all the sentences where the lemma $l' \in \hat{L}_s$ appears in collocation with $l \in L_s$, Z is $\sum_{l \in L} |\Phi_{l,s}| + \sum_{l' \in \hat{L}} |\hat{\Phi}_{l',s}|$, and $\textsc{Bert}(\lambda, \sigma)$ is the contextualized embedding for the lemma λ in the context σ.

At the end of this step, the synset s is associated with a vector v_{C}^s created as shown above.

4.3 Sense Embedding

In this final step, we first create sense-level representations by leveraging the contexts in SemCor

[4] We discard from this calculation the non-content words and the stopwords.

[5] We use UKB as it can directly take as input the Bag-of-Words representations of the clusters.

	Synsets	Sentences	Annotations	Avg sentences per synset	Synsets with 1 example
Cluster	71,025	9,274,698	10,575,541	148	1096
SyntagNet	19,706	1,324,863	2,649,726	134	763
ALL	77,195	10,599,561	13,225,267	141	1318

Table 3: Statistics of the contexts extracted by the similarity-based (Cluster) and collocation-aware (SyntagNet) extraction step in Section 4.1.

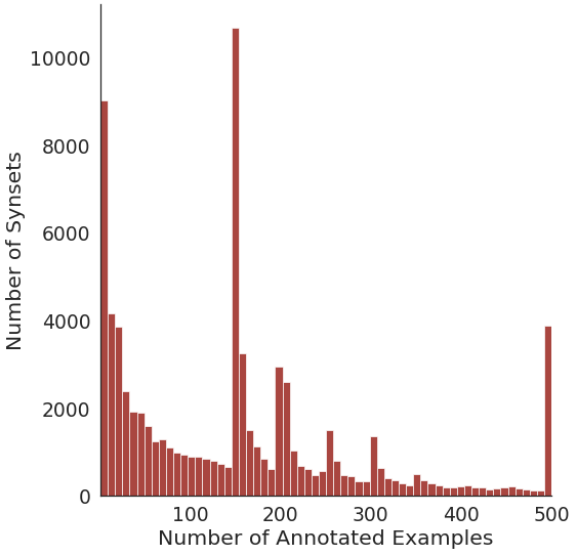

Figure 1: Histogram showing the number of synsets (y axis) by the number of annotated examples (x axis, bucket-based).

and the WordNet glosses, and then enrich them with our synset embeddings.

For each sense θ of s we create its embedding from its contextual occurrences within SemCor and its definition in WordNet. As for the SemCor part, we apply Peters et al. (2018)'s method to compute its representation v^{θ}_{SC}, i.e., we average the BERT embeddings of all the words in SemCor tagged with θ. As regards the sense gloss part, instead, we follow Loureiro and Jorge (2019) and prepend to the gloss of s both the lemma of θ and all the lexicalizations of s, and compute the sense gloss embedding v^{θ}_{G} by averaging the BERT representations of the words therein. For example, given the *spring* sense of the synset {*spring, fountain, natural spring*} (*A natural flow of ground water*), its sense gloss embedding is the average of the BERT representations of the following enriched gloss: *"spring - spring, fountain, natural spring - A natural flow of ground water"*.

We compute the representation ARES^{θ} for the sense θ of the synset s as follows:

$$\text{ARES}^{\theta} = v^{\theta}_{\text{SC}} \parallel v^{\theta}_{\text{G}} \odot v^{s}_{\text{C}}$$

where $\odot$ represents the mean between two vectors, and $\parallel$ their concatenation. If a sense does not occur in SemCor, we replace v^{θ}_{SC} with v^{θ}_{G} and apply the above formula. We recall from Section 3 that SemCor covers only 15% of WordNet's senses, nevertheless ARES is able to generalize over all the senses in WordNet thanks to the glosses and its automatically-retrieved contexts.

5 Statistics

In Table 3 we report the statistics of the sentences extracted as in Section 4.1. As one can see, our automatically-extracted annotations cover 65% of WordNet synsets (77,195 out of 117,659), providing at least one annotated example for 56,022 synsets that are not covered by SemCor. The total number of distinct tagged sentences is more than 10M for a total of 13M annotations. On average, most synsets have around 150 annotated examples, as shown in Figure 1.

6 WSD Experimental Setup

We now report the setup of the evaluation we conducted on the English and multilingual WSD tasks.

Evaluation Datasets We carried out the evaluation on the English all-words WSD framework by Raganato et al. (2017),[6] comprising five standard test sets, namely, Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-07 (Pradhan et al., 2007), SemEval-13 (Navigli et al., 2013), SemEval-15 (Moro and Navigli, 2015) along with ALL, i.e., the concatenation of all the test sets. As concerns the multilingual evaluation, we considered the latest versions of the two multilingual all-words WSD datasets of

[6] http://nlp.uniroma1.it/wsdeval/

SemEval-13 (Navigli et al., 2013) and SemEval-15 (Moro and Navigli, 2015), containing test sets for French, German, Italian and Spanish.[7] We report all results in terms of the F1 score, i.e., the harmonic mean of the precision and recall.[8]

ARES Configuration We used Wikipedia as input corpus since it is the largest general-domain resource currently available. Regarding the context extraction step (see Section 4.1), we set the number of clusters k for a lexeme l as the number of its senses in WordNet. We varied the number of words n to give as input to UKB between 5 and 25 with a 5 step and selected the value $n = 5$ by manually assessing the quality of a sample of the clusters' disambiguation output. As for the number of sentences t and ξ, we ranged them between 50 and 300 with a 50 step[9] and selected the values that maximized the performance in terms of F1 of ARES on SemEval-07,[10] i.e., $t = 150$ and $\xi = 50$. As regards the window size w, we followed Maru et al. (2019) and set $w = 3$.

Concerning BERT representations, we used the BERT large-cased model for English. To scale across languages, instead, we made use of BERT base-multilingual-cased (mBERT) so as to build unified representations that are shared across languages, i.e., ARES_{m}. For our multilingual representations, we focused on synset embeddings rather than sense ones. In fact, senses are language-specific as they are tied to one of the lemmas of the synset. Hence, we built ARES_{m} synset embeddings by averaging the representations of their English senses. We note that, while the pre-trained model differs between the two representations, the sentences used to create the embeddings are the same as the ones used for English. Following Loureiro and Jorge (2019), we took as BERT representation the sum of the last four hidden layers.

WSD Setup To test ARES on the WSD task, we employed the 1-NN algorithm. To this end, we computed the BERT representation of each word w in the test sentences and compared it with the embeddings corresponding to the senses of w in WordNet. Since ARES vectors are made of the concatenation of two BERT representations (Section 4.3), we repeated the embedding of w in order to match the shape of ARES vectors. Thus, we took as prediction the sense that maximizes the similarity with w's representation. For languages other than English, we considered as candidate synsets for a lemma those associated with it in BabelNet 4.0, i.e., a multilingual knowledge base providing lexicalizations of concepts in different languages.

Comparison systems We compared ARES with both knowledge-based and supervised approaches on English. As knowledge-based systems, we considered UKB with SyntagNet's relations (Scozzafava et al., 2020, $\text{UKB}_{\text{+Syn}}$), and SensEmBERT (Scarlini et al., 2020), along with its supervised version, i.e., $\text{SensEmBERT}_{\text{sup}}$. SensEmBERT and $\text{SensEmBERT}_{\text{sup}}$ cover only nominal senses, so we used the Most Frequent Sense (MFS) backoff strategy, i.e., predicting the most frequent sense of a lemma in WordNet, for tagging instances with other POS tags.

Among supervised systems, we tested against $\text{EWISE}_{\text{ConvE}}$ (Kumar et al., 2019), KnowBERT (Peters et al., 2019), the vocabulary compression model by Vial et al. (2019, BERT_{hyp}),[11] GlossBERT (Huang et al., 2019) and the approach proposed by Hadiwinoto et al. (2019, $\text{BERT}_{\text{GLU+LW}}$). Moreover, we compared against Loureiro and Jorge (2019, LMMS) and Peters et al. (2018)'s method using BERT (BERT k-NN). We also report the performance of these two latter approaches by using mBERT instead of BERT large, i.e., $\text{LMMS}_{\text{mBERT}}$ and mBERT k-NN. All supervised systems under comparison use SemCor only as training corpus.

We performed additional comparisons by using Peters et al. (2018)'s method with BERT on SemCor+OMSTI (Taghipour and Ng, 2015, $\text{SemCor+OMSTI}_{\text{BERT}}$), a semi-automatically generated extension of SemCor, and OneSeC (Scarlini et al., 2019, $\text{OneSeC}_{\text{BERT}}$), an automatically-tagged corpus.[12]

On the multilingual WSD tasks we compared against SensEmBERT and UKB augmented with SyntagNet's relations (Scozzafava et al., 2020, $\text{UKB}_{\text{+Syn}}$). We also trained a baseline on English data only, i.e., SemCor, and we tested it in all the

[7] https://github.com/SapienzaNLP/mwsd-datasets

[8] We used the scoring script in the Raganato et al. (2017)'s framework to compute all performances.

[9] All hyperparameters search spaces were manually chosen.

[10] We chose SemEval-07 as it is the standard development set used in the literature (Raganato et al., 2017).

[11] We excluded from the comparison both the ensemble and the model trained on SemCor and the WordNet disambiguated glosses reported by Vial et al. (2019) as it would not allow a fair comparison with the other systems under evaluation.

[12] OneSeC covers only nominal senses, so we resorted to the MFS strategy for instances with other POS tags.

	Model	Test Sets					Concatenation of All Test Sets				
		Senseval-2	Senseval-3	SemEval-07	SemEval-13	SemEval-15	Nouns	Verbs	Adj	Adv	ALL
KB	MFS	65.6	66.0	54.5	63.8	67.1	67.7	50.3	74.3	80.9	65.2
	OneSeC $_{\text{BERT}}$ (2019)	64.0	58.7	49.9	62.8	69.9	62.8	50.3	74.3	80.9	62.3
	UKB$_{+\text{Syn}}$ (2020)	71.2	71.6	59.6	72.4	75.6	-	-	-	-	71.5
	SensEmBERT (2020)	70.8	65.4	58.0	74.8	75.0	75.9	50.3	74.3	80.9	70.1
Supervised	BERT$_{\text{hyp}}$ (2019)	-	-	-	-	-	-	-	-	-	75.6
	EWISE$_{\text{ConvE}}$ (2019)	67.3	73.8	71.1	69.4	74.5	74.0	60.2	78.0	82.1	71.8
	KnowBert (2019)	-	-	-	-	-	-	-	-	-	75.1
	BERT$_{\text{GLU+LW}}$ (2019)	75.5	73.4	68.5	71.0	76.2	-	-	-	-	74.0
	GlossBERT (2019)	77.7	75.2	**76.1**	72.5	80.4	79.8	67.1	79.6	**87.4**	77.0
Sup$_{cont}$	SemCor+OMSTI$_{\text{BERT}}$ (2015)	74.0	70.6	63.1	72.4	75.0	74.8	60.1	77.5	83.2	72.2
	mBERT k-NN + MFS (2019)	72.7	70.1	62.4	69.0	72.0	73.2	57.9	75.9	82.1	70.5
	BERT k-NN + MFS (2019)	77.0	73.5	66.0	71.6	74.5	75.7	63.3	79.8	85.8	73.9
	LMMS$_{\text{mBERT}}$ (2019)	68.5	64.0	57.6	68.1	66.1	70.3	50.2	73.1	74.6	66.3
	LMMS (2019)	76.3	75.6	68.1	75.1	77.0	78.0	64.0	**80.7**	83.5	75.4
	SensEmBERT$_{\text{sup}}$ (2020)	72.2	69.9	60.2	**78.7**	75.0	80.5	50.3	74.3	80.9	72.8
Ours	ARES$_{\text{m}}$	74.8	71.5	64.8	72.7	77.0	75.9	62.3	76.8	81.2	73.2
	ARES	**78.0**	**77.1**	71.0	77.3	**83.2**	**80.6**	**68.3**	80.5	83.5	**77.9**

Table 4: F1 on the test sets of the all-words English WSD framework. *KB*: knowledge-based approaches; *Sup$_{cont}$*: supervised models exploiting contextual representations. Statistically-significant difference computed on the recall attained on the ALL dataset between ARES and GlossBERT is underlined (χ^2 with $p < 0.05$).

Model	ALL$_{\text{LFS}}$	ALL$_{\text{LFW}}$
LMMS	61.6	74.8
GlossBERT	62.0	75.6
ARES	**65.2**	**81.1**

Table 5: Results in terms of F1 on the ALL$_{\text{LFS}}$ and ALL$_{\text{LFW}}$ datasets.

other languages. To this end, we used mBERT with frozen weights followed by a linear layer with *swish* activation and an unbiased softmax classifier on top.[13] In addition, we report the performance of LMMS$_{\text{mBERT}}$ and mBERT k-NN on the multilingual datasets.

7 WSD Results

We now report the results of the evaluation we carried out on the English and multilingual WSD tasks, along with an ablation study of ARES components.

7.1 English all-words WSD

In Table 4 we report the results attained by the systems under comparison on the all-words English WSD datasets. Our direct competitors, i.e., SensEmBERT$_{\text{sup}}$ and LMMS, score, respectively, 5.1 and 2.5 F1 points lower than ARES. This comparison shows the effectiveness of different approaches in coping with the paucity of sense-annotated data for WSD. On the one hand, the SensEmBERT approach is effective in modeling nominal meanings, however, it cannot scale over other POS tags due to the limitations of its underlying resources. On the other hand, LMMS shows that the WordNet topology can be exploited to propagate the latent representations of frequent meanings towards those not appearing in sense-annotated corpora. Nevertheless, these less frequent senses do not have a specific characterization and thus their representations are less refined, as we also show in Section 7.2. Our approach overcomes both these limitations, being able to create better-characterizing representations across senses with different POS tags. This leads ARES to outperform the state of the art at the time of writing, i.e., GlossBERT, by almost 1 point on ALL by simply employing a 1-NN algorithm, and hence requiring no expensive fine-tuning procedure.

7.2 WSD on Infrequent Words and Senses

To test the ability of ARES and its competitors to scale over rare words and senses, we extracted two new test sets from ALL: i) ALL$_{\text{LFS}}$, which includes the 1139 instances in ALL associated with a sense not in SemCor; ii) ALL$_{\text{LFW}}$, which includes the 222 instances in ALL associated with a non-monosemous word not tagged in SemCor. As shown in Table 5, ARES proves to be the best system across the board, achieving the highest result on both datasets. This shows that the contexts extracted by ARES help balance the quality of meanings' representations across senses with different frequencies, without disadvantaging rare senses in favor of the more frequent ones. In contrast, both LMMS and GlossBERT are more

[13] See Appendix A.2 for training details.

Model	ALL
$\text{Cluster}_{\text{cont}}$	70.5
Syn_{cont}	60.1
$\text{Cluster}_{\text{cont}} \odot \text{Syn}_{\text{cont}}$	71.1
SemCor	69.2
SemCor $\Vert$ Gloss	75.7
SemCor $\Vert$ A_{CS}	77.1
SemCor $\Vert$ ($\text{A}_{\text{CS}} \odot$ Gloss)	77.9

Table 6: Ablation in terms of F1 of the different components of ARES on the ALL dataset. $\Vert$ indicates the concatenation while $\odot$ the average.

Model	SemEval-13				SemEval-15				AVG
	IT	ES	FR	DE	IT		ES		
					N	ALL	N	ALL	
mBERT	74.8	74.6	80.3	79.0	63.8	69.1	60.9	64.7	73.8
$\text{UKB}_{+\text{SyntagNet}}$	72.1*	74.1*	70.3*	76.4*	**68.2***	69.0*	64.3*	63.4*	70.9*
SensEmBERT	69.8*	73.4*	77.8*	79.2*	68.1*	-	68.1*	-	-
mBERT k-NN	68.6	69.3	75.4	73.8	59.1	64.6	56.3	61.6	68.8
$\text{LMMS}_{\text{mBERT}}$	68.0	66.3	76.2	78.3	61.2	62.5	63.0	60.1	68.5
ARES_{m}	**77.0**	**75.3**	**81.2**	**79.6**	68.0	**71.4**	**68.6**	**70.1**	**75.7**

Table 7: F1 on the WSD tasks's languages (SemEval-13 and SemEval-15) and the macro F1 score computed across all languages. Statistically-significant difference between ARES_{m} and mBERT's recalls is underlined (χ^2 with $p < 0.05$). *: Recomputed on the latest version of the datasets.

biased towards those representations in SemCor, hence losing ground on both datasets with a gap of 3.6 and 3.2 points, respectively, on ALL_{LFS} when compared to ARES. This latter, instead, by taking advantage of its automatically-retrieved contexts, scales better over rare words and senses, and outperforms its competitors on both datasets with the highest result of 81.1 on ALL_{LFW}.

7.3 Ablation Study

We now measure the impact that each part of our vectors has on the final results by means of an ablation study on the ALL dataset. The upper side of Table 6 compares the two kinds of contexts that we automatically retrieve (Section 4.1). As one can see, the $\text{Cluster}_{\text{cont}}$ alone, i.e., the sentences retrieved by means of the similarity-based step, already attains good results. When combined with the contexts extracted thanks to SyntagNet, i.e., Syn_{cont}, it gains 0.6 extra points. In the lower part of the Table, we show different combinations of the vectors built from SemCor, our contexts and the WordNet glosses. We indicate with A_{CS} and Gloss the vectors built from our extracted contexts (see Equation 4.2) and the sense gloss (see Section 4.3), respectively. SemCor alone attains 69.2 points, 1.3 points less than $\text{Cluster}_{\text{cont}}$. This is because SemCor does not provide examples for all WordNet meanings, therefore having a lower recall. When combining SemCor with WordNet's glosses (SemCor $\Vert$ Gloss) and A_{CS} (SemCor $\Vert$ A_{CS}), we have a 6.5 and 7.9 improvement, respectively. Finally, when combining the three components, we obtain our best score of 77.9 F1 points on ALL.

7.4 Multilingual all-words WSD

Finally, we investigate the ability of ARES_{m} to scale across languages by testing it on the multilingual WSD datasets of SemEval-13 and SemEval-15.[14] As shown in Table 7, ARES_{m} is the best system across the board, achieving state-of-the-art results on all languages of both datasets but the Italian nominal instances of SemEval-15. On average, ARES scores almost 2.0 F1 points higher compared to the second best performing system, i.e., mBERT. When compared to $\text{LMMS}_{\text{mBERT}}$, ARES achieves 7.0 F1 points higher on average. This may be due to the fact that our automatically-retrieved sentences provide a better contextualization of meanings than the propagation technique employed by LMMS, hence allowing our embeddings to scale effectively across languages. Finally, we surpass SensEmBERT and attain state-of-the-art performance on all languages of the multilingual all-words WSD tasks while at the same time keeping the quality on nouns high.

The evaluation carried out shows how beneficial our embeddings are to the English and the multilingual WSD tasks. ARES, in fact, proves to carry high-quality semantic information within its representations, which enables it to generalize over both words and languages, and achieve state-of-the-art results in all the tested settings.

8 WiC Experimental Setup

In this Section we further inspect the properties of our embeddings by measuring the improvements they bring to the Word-in-Context (WiC) task.[15]

Evaluation Dataset We tested on the Word-in-Context task (Pilehvar and Camacho-Collados, 2019, WiC),[16] i.e., a binary classification problem where, given a target word w and two contexts c_1 and c_2, the task is to determine if w occurs with the

[14] We also report the results on only the nominal instances of SemEval-15 to be comparable with SensEmBERT.

[15] https://super.gluebenchmark.com/

[16] Version 1.1 of SuperGLUE (Wang et al., 2019).

Model	Accuracy	Trainable Parameters
BERT$_{\text{LARGE}}$ (2019)	69.6	340 M
RoBERTa (2019)	69.9	355 M
KnowBert $_{\text{W+W}}$ (2019)	70.9	523 M
SenseBERT$_{\text{LARGE}}$ (2020)	72.1	380 M
T5-Large (2019)	69.3	770 M
T5-3B (2019)	72.1	3000 M
T5-11B (2019)	**76.1**	11000 M
BERT$_{\text{ARES}}$	72.2	342 M

Table 8: Results in terms of accuracy on the WiC test set and number of trainable parameters of each model.

same meaning in c_1 and c_2. We report the results in terms of accuracy, i.e., the number of correct answers over the total number of predictions.

WiC Model We integrated our embeddings as features in the English BERT large-cased model, i.e., BERT$_{\text{ARES}}$, during finetuning. Following Wang et al. (2019) we concatenated the two input sentences c_1 and c_2 with the [SEP] token and fed them to BERT with a logistic regression classifier on top. The last layer took as input the [CLS] embedding and the two representations of the target word w in c_1 and c_2. As additional features, we considered the senses s_1 and s_2 of w in c_1 and c_2, respectively, that we predicted by means of ARES as in Section 6. Then, we applied a dense layer – which we trained during finetuning – to the ARES embeddings of s_1 and s_2 and reduced their dimensionality to 1024. Finally, we concatenated the input of the classifier with these two new representations.[17]

Comparison Systems We compared ARES against the best performing models on the WiC task. We considered three pre-trained language models fine-tuned on WiC, i.e., BERT$_{\text{LARGE}}$ (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2019), and two language models which leverage external knowledge while pre-training, i.e., KnowBert (Peters et al., 2019) and SenseBERT$_{\text{LARGE}}$ (Levine et al., 2020).

9 WiC Results

In Table 8 we report the results of the systems under comparison on the WiC test set. BERT$_{\text{ARES}}$ attains 2.6 points more than its base model, i.e., BERT$_{\text{LARGE}}$, while exploiting ARES embeddings in a straightforward manner at finetuning. Moreover, BERT$_{\text{ARES}}$ performs better than or on a par with its closest competitors, i.e., KnowBert, SenseBERT$_{\text{LARGE}}$ and T5 (Large, and 3B), which, instead, rely on more complex architectures, specific pre-training phases and between 3000 M and 40 M more parameters. T5-11B is the only model achieving better results than BERT$_{\text{ARES}}$, mainly due to the large difference in terms of trainable weights (with T5-11B being 30 times bigger.)

[17] See Appendix A.3 for training details.

10 Conclusion

In this paper we presented ARES, a semi-supervised approach for producing embeddings of senses in English and across different languages. ARES can couple the information within sense-annotated corpora with that automatically created by means of a cluster-based algorithm so as to produce high-quality latent representations for the concepts within a lexical knowledge base. Our experiments showed that despite relying on English data only ARES outperforms all its alternatives. It achieves state-of-the-art results on both English and multilingual WSD benchmarks, leveraging BERT large and mBERT, respectively, as underlying pre-trained language models. We further tested our embeddings in the WiC task where they lead a baseline neural model to outperform its closest competitors that rely on larger architectures or dedicated pre-training routines. Our embeddings computed with BERT large and mBERT and the automatically-extracted contexts are available at `http://sensembert.org/ares`.

As future work, we plan to exploit the information brought by our embeddings to other downstream tasks, such as multilingual Semantic Role Labeling (Di Fabio et al., 2019; Conia et al., 2020) and cross-lingual Semantic Parsing (Blloshmi et al., 2020).

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 under the European Union's Horizon 2020 research and innovation programme.

This work was supported in part by the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of the Sapienza University of Rome.

References

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

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2018. The risk of sub-optimal use of Open Source NLP Software: UKB is inadvertently state-of-the-art in knowledge-based WSD. In *Proc. of ACL*, pages 29–33.

Edoardo Barba, Luigi Procopio, Niccolò Campolungo, Tommaso Pasini, and Roberto Navigli. 2020. MuLaN: Multilingual Label propagatioN for Word Sense Disambiguation. In *Proc. of IJCAI*, pages 3837–3844.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or: "How we went beyond word sense inventories and learned to gloss". In *Proc. of EMNLP*.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. In *Proc. of ACL*, pages 2854–2864.

Terra Blevins and Luke Zettlemoyer. 2020. Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. In *Proc. of ACL*, pages 1006–1017.

Rexhina Blloshmi, Rocco Tripodi, and Roberto Navigli. 2020. XL-AMR: Enabling Cross-Lingual AMR parsing with transfer learning techniques. In *Proc. of EMNLP*.

Simone Conia, Fabrizio Brignone, Davide Zanfardino, and Roberto Navigli. 2020. InVeRo: Making semantic role labeling accessible with intelligible verbs and roles. In *Proc. of EMNLP*.

Simone Conia and Roberto Navigli. 2020. Conception: Multilingually-enhanced, human-readable concept vector representations. In *Proc. of COLING*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, pages 4171–4186.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a Novel Large-Scale Verbal Semantic Resource and Its Application to Semantic Role Labeling. In *Proc. of EMNLP-IJCNLP*, pages 627–637.

Philip Edmonds and Scott Cotton. 2001. Senseval-2: overview. In *Proc. of SENSEVAL*, pages 1–5.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. In *Proc. of EMNLP*, pages 5296–5305.

Taher H. Haveliwala, Aristides Gionis Dan Klein, and Piotr Indyk. 2002. Evaluating strategies for similarity search on the web. In *Proc. of WWW*, pages 432–442.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. In *Proc. of EMNLP*, pages 3500–3505.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proc. of ACL*, pages 5670–5681.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving Some Sense into BERT. In *Proc. of ACL*, pages 4656–4667.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke S. Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proc. of ACL*, pages 5682–5691.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging Supervised Word Sense Disambiguation with Lexical-Semantic Combinations. In *Proc. of EMNLP*, pages 3525–3531.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. Context2vec: Learning generic context embedding with bidirectional LSTM. In *Proc. of CoNLL*, pages 51–61.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, pages 235–244.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proc. of the Workshop on Human Language Technology*, pages 303–308.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proc. of SemEval-2015*, pages 288–297.

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

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, pages 1–69.

Roberto Navigli. 2018. Natural Language Understanding: Instructions for (present and future) use. In *Proc. of IJCAI 2018*, pages 5697–5702.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual Word Sense Disambiguation. In *Proc. of SemEval-2013*, pages 222–231.

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

Tommaso Pasini. 2020. The Knowledge Acquisition Bottleneck Problem in Multilingual Word Sense Disambiguation. In *Proc. of IJCAI*, pages 4936–4942.

Tommaso Pasini and Roberto Navigli. 2020. Train-o-matic: Supervised word sense disambiguation with no (manual) effort. *AIJ*, 279.

Tommaso Pasini, Federico Scozzafava, and Bianca Scarlini. 2020. CluBERT: A Cluster-Based Approach for Learning Sense Distributions in Multiple Languages. In *Proc. of ACL*, pages 4008–4018.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*, pages 2227–2237.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proc. of EMNLP*, pages 43–54.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proc. of NAACL*, pages 1267–1273.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proc. of ACL*, pages 4996–5001.

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proc. of SemEval-2007*, pages 87–92.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proc. of EACL*, pages 99–110.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Proc. of NeurIPS*, pages 8592–8600.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2019. Just "OneSeC" for Producing Multilingual Sense-Annotated Data. In *Proc. of ACL*, pages 699–709.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *Proc. of AAAI*, pages 8758–8765.

Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized PageRank with syntagmatic information for multilingual Word Sense Disambiguation. In *Proc. of ACL*, pages 37–46.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proc. of Senseval 3*, pages 41–43.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proc. of CoNLL*, pages 338–344.

Rocco Tripodi and Roberto Navigli. 2019. Game theory meets embeddings: a unified framework for word sense disambiguation. In *Proc. of EMNLP*, pages 88–99.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation. In *Proc. of Global Wordnet Conference*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Proc. of NeurIPS*, pages 3266–3280.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proc. of NeurIPS*, pages 5754–5764.

A Supplementary Materials

A.1 Computing Infrastractures

All the experiments were performed using a x86-64 architecture with 64 GBs of RAM and a GeForce GTX 1080 Ti.

A.2 mBERT Baseline Training Details

The model was trained with Adam (Kingma and Ba, 2015) optimizer on SemCor for 50 epochs, and tuned on the SemEval-07 dataset. The learning rate was set to $2 \cdot 10^{-5}$ and gradient clipping to 1.0. The training was stopped earlier in the case that the loss ceased decreasing for 3 consequent epochs on the development set. We encoded each word by taking the sum of its hidden representations of the last four layers of the BERT base-multilingual-cased pre-trained model.

A.3 WiC Finetuning Details

We trained our BERT-based model with the `jiant`'s library.[18] As for the hyperparameters, we used the ones reported by Devlin et al. (2019), which are the standard configuration in the `jiant`'s framework. We finetuned the BERT large-cased pretrained model for 4 epochs with batch size equal to 4, learning rate $1 \cdot 10^{-4}$ and Adam as optimizer (Kingma and Ba, 2015). The dropout probability was set to 0.1 on every layer. The average runtime of the model was 30 minutes, including the validation on the development set at the end of each epoch. The accuracy we achieved on the development set was 73.7.

The accuracy on the test set was computed by uploading the predictions of our model on the SuperGLUE website (Wang et al., 2019).[19]

[18] https://github.com/nyu-mll/jiant
[19] https://super.gluebenchmark.com