Analyzing Homonymy Disambiguation Capabilities of Pretrained Language Models

Lorenzo Proietti[1], **Stefano Perrella**[1], **Simone Tedeschi**[1,2], **Giulia Vulpis**[1],
Leonardo Lavalle[1], **Andrea Sanchietti**[1], **Andrea Ferrari**[1], **Roberto Navigli**[1]
[1] Sapienza University of Rome, [2] Babelscape
{lproietti, perrella, tedeschi, navigli}@diag.uniroma1.it
{vulpis.1807771, lavalle.1838492, sanchietti.1883210, ferrari.2086779}@studenti.uniroma1.it

Abstract

Word Sense Disambiguation (WSD) is a key task in Natural Language Processing (NLP), aiming to assign the correct meaning (sense) to a word in context. However, traditional WSD systems rely on WordNet as the underlying sense inventory, often differentiating meticulously between subtle nuances of word meanings, which may lead to excessive complexity and reduced practicality of WSD systems in today's NLP. Indeed, current Pretrained Language Models (PLMs) do seem to be able to perform disambiguation, but it is not clear to what extent, or to what level of granularity, they actually operate. In this paper, we address these points and, firstly, introduce a new large-scale resource that leverages homonymy relations to systematically cluster WordNet senses, effectively reducing the granularity of word senses to a very coarse-grained level; secondly, we use this resource to train Homonymy Disambiguation systems and investigate whether PLMs are inherently able to differentiate coarse-grained word senses. Our findings demonstrate that, while state-of-the-art models still struggle to choose the correct fine-grained meaning of a word in context, Homonymy Disambiguation systems are able to differentiate homonyms with up to 95% accuracy scores even without fine-tuning the underlying PLM. We release our data and code at https://github.com/SapienzaNLP/homonymy-wsd.

Keywords: Semantics, Word Sense Disambiguation, Corpus (Creation, Annotation, etc.)

1. Introduction

Word Sense Disambiguation (WSD) is a longstanding challenge in Natural Language Processing (NLP), whose objective is to associate words in context with their most suitable entries in a pre-defined sense inventory (Navigli, 2009; Bevilacqua et al., 2021). WSD can be beneficial for a wide range of NLP tasks, such as Machine Translation (Chan et al., 2007; Pu et al., 2018), Information Retrieval (Sanderson, 1994; Zhong and Ng, 2012), and Sentiment Analysis (Sumanth and Inkpen, 2015; Pamungkas and Putri, 2017), among others. At the same time, recent works suggest that PLMs are intrinsically able to capture various linguistic phenomena, including semantics (Amrami and Goldberg, 2018; Zhou et al., 2019; Loureiro et al., 2021), raising doubts concerning the benefits of WSD in today's NLP. However, analyzing the capabilities of PLMs in the context of WSD is complicated by the granular sense distinctions available in lexical resources like WordNet (Miller, 1994). While these resources are invaluable for capturing the nuanced meanings of words, their nature frequently poses unnecessary practical challenges. Indeed, word senses are often hard to differentiate even for experienced human annotators, with an estimated inter-annotator agreement lower than 80%, as measured on fine-grained sense inventories (Chklovski and Mihalcea, 2003; Snyder and Palmer, 2004; Palmer et al., 2007; Navigli et al., 2013; Moro and Navigli, 2015).

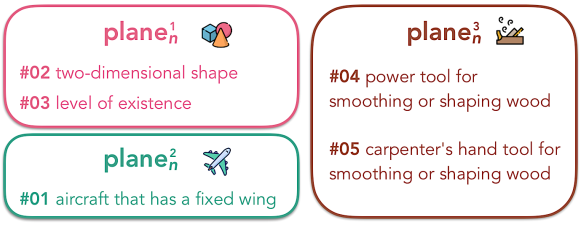

plane^1_n
#02 two-dimensional shape
#03 level of existence
plane^2_n
#01 aircraft that has a fixed wing
plane^3_n
#04 power tool for smoothing or shaping wood
#05 carpenter's hand tool for smoothing or shaping wood

Figure 1: The senses of the noun 'plane' from WordNet 3.0, grouped by its three homonyms.

To determine to what extent PLMs are able to perform disambiguation, we exploit the difference between homonymous and polysemous senses[1] of a word, and introduce a new resource for coarse-grained WSD. We separate homonymous senses in WordNet, and cluster polysemous ones, by manually aligning word senses with their corresponding sense definitions in the Oxford Dictionary of English (Soanes and Stevenson, 2003). As a result, our resource strikes a balance between the semantic richness of word meanings and their suitability for

[1] "Homonymy is the relation between unrelated senses that share a form, while polysemy is the relation between related senses that share a form." (Jurafsky and Martin, 2009). In the following, we refer to distinct, unrelated, and coarse-grained meanings as homonymous senses.

practical applications. Figure 1 shows an example of WordNet senses grouped into their homonymy clusters.

We then leverage our newly-introduced resource in order to study the capabilities of PLMs and WSD systems in the context of Homonymy Disambiguation, and put forward the following research questions:

- **(RQ1)** Is the capability to properly capture homonymy already acquired by PLMs during pretraining?
- **(RQ2)** Do fine-grained[2] WSD systems inherently distinguish homonymous senses? If this is not the case, can fine-grained WSD systems benefit from the predictions of Homonymy Disambiguation systems?

Finally, to encourage the development and study of Homonymy Disambiguation systems, we release our resource and models at `https://github.com/SapienzaNLP/homonymy-wsd`.

2. Related Work

We now describe other relevant approaches for the creation of coarse-grained sense inventories (Section 2.1). Subsequently, considering that our resource is created with the main purpose of enabling the analysis of the disambiguation capabilities of PLMs, we also review past methods for probing them on lexical-semantic tasks (Section 2.2).

2.1. Coarse-Grained Inventories

During the last three decades, numerous manual and automated techniques have been proposed for clustering senses contained in well-known sense inventories (Dolan, 1994; Pustejovsky and Buitelaar, 1998; Pedersen et al., 2004; Palmer et al., 2007; Navigli et al., 2007; Lacerra et al., 2020). Some early works aimed at uncovering sense similarities within dictionary entries (Dolan, 1994), while others introduced synset similarity measures based on properties in WordNet, including gloss-based heuristics (Banerjee and Pedersen, 2003), content-based metrics (Resnik, 1995; Lin, 1998a; Jiang and Conrath, 1997), and structure-based criteria (Mihalcea and Moldovan, 2001), among others. A large body of work also attempted to capture corpus-based estimates of word similarity (Pereira et al., 1993; Lin, 1998b; Chugur et al., 2002; Agirre, 2004; McCarthy, 2006). Nevertheless, the scarcity of extensive sense-tagged corpora hindered the efficacious use of these techniques in comparing various meanings of the same word.

Another set of methods involved associating senses with coarser inventories by means of either manually annotated or automatically generated mappings. For instance, an attempt to provide general sense distinctions for Senseval-1 (Kilgarriff and Palmer, 2000) involved mapping between WordNet and the Hector lexicon (Palmer et al., 2007). Similar initiatives included mappings between WordNet and PropBank (Palmer et al., 2004) as well as mappings to Levin classes (Levin, 1993). Navigli (2006), instead, introduced an automated strategy for mapping one sense inventory to another by exploiting similarities in gloss definitions and the structural relationships between the two inventories. This latter work was then used as the starting point for introducing the task of coarse-grained all-words WSD at SemEval-2007 (Navigli et al., 2007).

More recently, Vial et al. (2019) proposed a methodology that exploited semantic relationships within WordNet, specifically hypernymy and hyponymy, to compress the sense vocabulary. Their technique focused on automatically clustering closely related senses into broader categories, significantly reducing the granularity of WordNet. On a different note, Lacerra et al. (2020) presented CSI (Coarse Sense Inventory), which aligned 83K WordNet synsets with a set of 45 high-level semantic labels through a combination of manual and semi-automatic steps. While both Vial et al. (2019) and Lacerra et al. (2020) aimed at tackling the issue of fine-grained sense distinctions in WordNet, their methodologies and results differed significantly. Vial et al. (2019) employed semantic relationships within WordNet to compress its sense vocabulary, effectively reducing granularity and maintaining the framework of the original sense inventory. In contrast, the CSI introduced by Lacerra et al. (2020) abstracted these fine distinctions and organized WordNet senses into a newly-defined, domain-based inventory of 45 high-level semantic labels. More similar to our work, Maudslay and Teufel (2022) employed a fully-automatic approach to link WordNet sense definitions to the Oxford English Dictionary, in order to enrich WordNet with homonymy annotations.

However, despite the valuable contributions of the aforementioned works in simplifying the organization of word senses, none of them has resulted in a large-scale, coarse-grained inventory that can be reliably utilized in lexical-semantic tasks. Key challenges include limited coverage in some inventories or their need to be created through automatic steps. Additionally, with the exception of Maudslay and Teufel (2022), none of the previous studies exploited homonymy, despite the fact that homonymy represents a powerful linguistic tool for distinguishing systematically between word senses and reducing their granularity. Finally, even when WSD

[2] Considering Homonymy Disambiguation as coarse-grained WSD, we refer to standard WSD as fine-grained WSD.

systems are assessed using the most recent coarse inventories, they merely achieve up to 85.9% accuracy (Lacerra et al., 2020), and hence they are still far from achieving performance that might enable improvements in downstream tasks.

2.2. Probing Language Models

Various studies have shown that fulfilling the language modeling objective inherently forces the model to capture various linguistic phenomena. A highly-studied phenomenon is syntax, which has been investigated both for earlier LSTM-based models (Linzen et al., 2016; Kuncoro et al., 2018), as well as for the more recent Transformer-based ones (Goldberg, 2019; Hewitt and Manning, 2019; Saphra and Lopez, 2019; Jawahar et al., 2019; van Schijndel et al., 2019; Tenney et al., 2019).

In the context of semantics, and particularly with a focus on lexical ambiguity, most of the studies analyzing language models have chosen WSD and lexical substitution as their experimental benchmarks. Yuan et al. (2016b) advanced WSD by integrating an LSTM language model with semi-supervised learning, notably improving verb disambiguation. This innovative approach paved the way for subsequent studies, with Amrami and Goldberg (2018) further demonstrating the versatility of LSTM language models by applying them to the task of Word Sense Induction (Navigli, 2009; Yuan et al., 2016a; Amplayo et al., 2019). Specifically, they investigated whether the predictions of an LSTM for a word in context provided a useful way for retrieving substitutes, and demonstrated that this information is indeed captured in the language model. From a more analytical point of view, Aina et al. (2019) proposed a probe task based on lexical substitution in order to understand the internal representations of an LSTM language model for predicting words in context. As regards Transformer-based models, Zhou et al. (2019) proposed a model based on BERT to achieve state-of-the-art results in lexical substitution, showing that BERT is particularly suited to finding senses of a word in context. Even more relevant to our work, Reif et al. (2019) studied the BERT's embedding space and observed that, generally, when contextualized BERT embeddings for ambiguous words are visualized, clear clusters for different senses are identifiable. Finally, Loureiro et al. (2021) introduced CoarseWSD-20, a dataset including a selection of twenty ambiguous words of different kinds, and analyzed the two major language model settings (i.e. feature extraction and fine-tuning) for coarse-grained WSD.

Along the same lines, but using our newly-introduced resource, we investigate whether and to what extent PLMs are inherently capable of capturing and distinguishing between homonymous senses.

3. Resource Creation

WordNet[3] provides various types of relational information about word senses, but homonymy information is currently missing. Indeed, for a given (lemma, PoS)[4] pair l in WordNet, its candidate senses $s_1, s_2, \ldots, s_k$ all have separate, independent entries, no matter how narrow their semantic gap is.

3.1. Transferring Homonymy from ODE to WordNet

In order to transfer homonyms from the Oxford Dictionary of English (ODE) to WordNet, we ask three linguists to manually associate WordNet senses with Oxford homonyms based on their definitions. As a first step, we automatically extract all (lemma, PoS) pairs of ODE that have at least two homonymous senses. Then, each linguist associates the fine-grained senses of WordNet with the coarse-grained ones present in ODE[5], when possible. Indeed, it may happen that some sense in WordNet cannot be associated with any homonymy cluster in Oxford, or vice versa.

More formally, let ODE be the set of (lemma, PoS) pairs in the Oxford Dictionary of English, then $\mathrm{ODE}_h \subset \mathrm{ODE}$ is the set of (lemma, PoS) pairs with at least two homonymous senses. Then, considering a (lemma, PoS) pair l, $H_l = \{h_1, h_2, \ldots, h_n\}$ is the set of its homonymous senses, and therefore $\mathrm{ODE}_h = \{l \in \mathrm{ODE}\colon |H_l| > 1\}$. Moreover, WN being the set of all (lemma, PoS) pairs in WordNet, we define $\mathrm{WN}_h = \mathrm{WN} \cap \mathrm{ODE}_h$. For every $l \in \mathrm{WN}_h$, let $S_l = \{s_1, s_2, \ldots, s_k\}$ be the set of its candidate WordNet senses. Then, for each $l \in \mathrm{WN}_h$, our goal is to find a mapping f_l from set S_l to set H_l. Each mapping f_l is not necessarily injective or surjective, i.e. multiple senses in S_l can be mapped to the same homonymous sense in H_l, and not all elements of H_l necessarily have to be mapped from S_l. This arises from the fact that i) Oxford homonyms contain more than a single fine-grained sense, ii) not all Oxford homonyms are represented by WordNet senses.

For the manual annotation process mentioned above, we measure the Fleiss' kappa score (Fleiss, 1971; Fleiss et al., 2013) to be $\kappa = 0.79$, highlighting a substantial agreement among the annotators. Specifically, the inter-annotator agreement was computed as follows: we asked the three annotators to map 30% of the (lemma, PoS) in Word-

[3] We use WordNet 3.0 in this work.

[4] PoS refers to the Part-of-Speech tag.

[5] The Oxford Dictionary of English organizes fine-grained senses into homonym entries (as we did in Figure 1 for WordNet), effectively separating homonymous senses belonging to different words.

Net that have at least two homonymous senses in the Oxford Dictionary of English, i.e. 30% of the (lemma, PoS) in WN_h. From this sample, we measured the Fleiss' kappa score. The final mapping in this sample was computed using majority voting. For the remaining 70% of (lemma, PoS) in WN_h, the annotation was completed by the annotator who was the most times (98.57%) in agreement with the majority vote in the aforementioned 30% sample.

3.2. Clustering Remaining WordNet Senses

As a result of the manual annotation procedure, we obtain a mapping between WordNet senses and ODE homonymy clusters. However, this mapping does not cover the entire WordNet repository because, i) the manual mapping involves only those (lemma, PoS) of ODE that have at least two homonymous senses, ii) even for those (lemma, PoS) that have been mapped, some of their WordNet senses do not have a correspondence in ODE. Nonetheless, since we are interested in enriching the entire WordNet repository with homonymy information, we devise an automatic strategy for extending our resource.

Formally, each non-mapped WordNet sense belongs to one of the two disjoint sets U_1 and U_2:

- $U_1 = \bigcup_{l \in \mathrm{WN}_r} S_l$;
- $U_2 = \{s \in S_l \mid l \in \mathrm{WN}_h, s \notin \mathsf{dom}(f_l)\}$.

Where $\mathrm{WN}_r = \mathrm{WN} \setminus \mathrm{WN}_h$ is the set of WordNet (lemma, PoS) pairs not involved in the previously described manual annotation procedure. Hence, U_1 is the set containing all candidate senses of the WordNet (lemma, PoS) pairs not in ODE_h. Since we adopt the Oxford Dictionary of English as the authoritative inventory for homonyms, if a WordNet (lemma, PoS) pair is not in ODE_h, we make the assumption that it does not have any homonymous senses.[6] As a direct implication of this assumption, all candidate senses associated with such a (lemma, PoS) are automatically mapped to a single newly-created homonymy cluster. This approach ensures a consistent treatment of WordNet senses in relation to their homonymic categorization in the Oxford Dictionary of English. In this automatic step we extend the mapping to cover the remaining $|\mathrm{WN}_r| = 152,893$ (lemma, PoS) pairs, on top of the ones that are involved in homonymy relations in ODE ($|\mathrm{WN}_h| = 2394$). We highlight that, among the (lemma, PoS) pairs in WN_r, only 16.4% are polysemous in WordNet; specifically, $|\{l \in \mathrm{WN}_r \colon |S_l| > 1\}| = 25,138$.

The senses in U_2, instead, are those for which human annotators could not identify a matching homonymy cluster in ODE, with $|U_2| = 506$. For 250 of these senses the solution is straightforward: in fact, each of these is the only non-mapped sense of a (lemma, PoS) pair; therefore, since it does not belong to other homonymy clusters in ODE, we can create a new cluster that contains only this sense. For the remaining 256 senses, instead, we are not able to automatically determine whether they should be new singleton clusters, or whether some of them should be grouped in the same cluster. For this reason, we ask the annotator to inspect these senses and decide the composition of the new clusters.[7] As a result, our resource maps every (lemma, PoS) pair in WordNet to its set of homonymous senses, each of which is a cluster of WordNet senses. We show an excerpt of the newly-created resource in Table 1.

To sum up, we reduced the number of distinct senses[8] in WordNet from $206,941$ to $158,131$. This results in a sizeable reduction in the average polysemy degree of WordNet lemmas[9] (see Table 2). More in detail, when restricting the analysis to polysemous (lemma, PoS) only, the average polysemy degree drops considerably, meaning that many fine-grained senses could effectively be clustered based on the homonymy relation.

3.3. Mapping WSD Datasets

We use our new resource to tag the instances of standard WSD datasets with their coarse-grained senses. By doing so, we enable the use of these datasets to answer the research questions outlined in Section 1. The datasets we considered for this step are:

- **SemCor** (Miller et al., 1993), a large sense-annotated corpus for WSD.
- **WordNet Examples**, contextual examples associated with specific synsets in WordNet.
- **SemEval-2007** (Palmer et al., 2001), typically used as development set for WSD systems. It is part of the test set made available by Raganato et al. (2017a), called **ALL**.
- **ALL*NEW*** (Maru et al., 2022), a refined version of the ALL test set.

[6] Please refer to Appendix A for examples of such (lemma, PoS) pairs.

[7] This annotation step was carried out by the annotator who was the most times in agreement with the majority vote mentioned in Section 3.1.

[8] `https://wordnet.princeton.edu/documentation/wnstats7wn`

[9] We are intentionally extending the concept of polysemy degree to also account for homonymous senses. When dealing with homonymous senses, we define the average polysemy degree to be the average number of homonymous senses of a lemma.

(lemma, PoS)	Homonym	Synset	Definition
(soil, NOUN)	soil.n.h.01	soil.n.02 territory.n.03 land.n.02	the part of the earth's surface consisting of humus and disintegrated rock the geographical area under the jurisdiction of a sovereign state material in the top layer of the surface of the earth in which plants can grow (especially with reference to its quality or use)
	grime.n.h.01	dirt.n.02	the state of being covered with unclean things
(list, VERB)	list.v.h.01	list.v.01 list.v.02 number.v.03	give or make a list of; name individually; give the names of include in a list enumerate
	list.v.h.02	list.v.03 list.v.04	cause to lean to the side tilt to one side

Table 1: For each (lemma, PoS) pair, there are its homonyms in the form `lemma.PoS.h.id`, which are coarse-grained senses, each grouping together one or more WordNet senses.

	Before		After	
	w/	w/o	w/	w/o
Noun	1.24	2.79	1.02	2.39
Verb	2.17	3.57	1.02	2.09
Adjective	1.40	2.71	1.00	2.07
Adverb	1.25	2.50	1.00	N/A

Table 2: Average polysemy degree of Nouns, Verbs, Adjectives, and Adverbs in WordNet before and after our homonymy-based clustering. w/ means that monosemous lemmas are included in the computation, while w/o means that the polysemy degree is computed by taking into consideration only the polysemous lemmas. 'N/A' indicates the absence of polysemous lemmas.

An established practice in the WSD literature (Raganato et al., 2017b; Huang et al., 2019; Blevins and Zettlemoyer, 2020) is to use SemCor as training set, SemEval-2007 as development set, and ALL as test set. Following more recent works (Bevilacqua and Navigli, 2020; Conia and Navigli, 2021; Barba et al., 2021a,b,c), we also include the WordNet Examples dataset in our training data. However, after tagging the aforementioned datasets with coarse-grained sense annotations, we observe too few polysemous instances in the development and test sets, as shown in Table 3. Such distribution would hinder the effectiveness of our analysis of coarse-grained WSD systems. For this reason, we concatenate all these datasets and split them into new training, development, and test sets, ensuring a number of polysemous instances that better fits our purposes. Table 4 shows the number of such instances in the new data splits.[10]

4. Experiments

We employ the newly-created resource to answer our research questions. For **RQ1**, our goal is to establish the extent to which current PLMs are capable of disambiguating homonyms. As representatives of all existing PLMs we choose four of the most popular ones: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020) and DeBERTa-v3 (He et al., 2021).[11] BERT and RoBERTa are Transformer-based encoder models that were pretrained using the Masked Language Modelling objective task; Clark et al. (2020), instead, introduced a different pretraining objective in ELECTRA called Replaced Token Detection, which was also used for training DeBERTa-v3. Considering their different pretraining objectives, we expect their output representations to have different properties, which renders our analysis more robust and of greater generality.

Regarding **RQ2**, instead, we want to investigate the relationship between Homonymy Disambiguation and fine-grained WSD. As the former is a simpler version of the latter, we wonder whether a system trained to perform the more challenging task of WSD is able to distinguish homonyms on par with a system trained specifically for Homonymy Disambiguation, and, if this is not the case, whether a Homonymy Disambiguation system can be used effectively to improve WSD performance.

4.1. Probing PLMs for Homonymy Disambiguation

In order to investigate whether PLMs learn the ability to disambiguate homonymous senses already during their pretraining, we devise a method, which we refer to as *distance-based disambiguation*, to tackle the disambiguation of homonyms without any kind of training. Given a test instance[12] to disambiguate, we want to assign to it the homonymy cluster that contains the closest sense, in terms

[10] Please refer to Appendix B for more details about the data mentioned in Section 3.3.

[11] More in detail, the exact models are bert-large-cased, facebook/roberta-large, google/electra-large-discriminator and microsoft/deberta-v3-large from Huggingface.

[12] With the term instance we refer to a word in context.

	Total	FGA	HA
SemCor	226,036	187,911	7865
SE7	455	429	16
ALL$_{NEW}$	4917	4917	353
WN Examples	47,269	33,414	1375

Table 3: Number of instances in standard WSD datasets. The FGA (Fine-Grained Ambiguous) items are the instances with more than one candidate sense; the HA (Homonymy Ambiguous) items are those instances that have more than one candidate homonymy cluster.

	Total	FGA	HA
Train	253,276	205,810	6224
Dev	8195	6689	1120
Test	17,206	14,172	2265

Table 4: Number of instances in the new train/dev/test split. The FGA (Fine-Grained Ambiguous) items are the instances that have more than one candidate sense; the HA (Homonymy Ambiguous) items are those instances that have more than one candidate homonymy cluster.

	Cosine	Euclidean
BERT	**95.24**	94.75
RoBERTa	93.92	93.92
ELECTRA	89.98	89.82
DeBERTa	91.30	91.46

Table 5: Distance-based Homonymy Disambiguation accuracy when using cosine and Euclidean distances, respectively. The highest accuracy is in bold, and the top two accuracy values with the two distance measures are underlined.

of either cosine or Euclidean distance.[13] In order to compute the distance between test instances and candidate senses, we use PLMs to extract their vector representations. An instance is represented by the contextualized embedding of the word in context.[14] Given this, a sense can have multiple representations, one for each training instance tagged with that particular sense. We compute the distance between the test instance and each of the sense representations, and use the smallest one for the prediction. Unfortunately, we cannot extract vector representations for all senses, because for some of them there are no training instances. For this reason, in this experiment, we restrict the test data to contain only the instances whose candidate homonymy clusters contain at least one sense that has a vector representation. In addition, considering that we are interested in establishing whether the selected PLMs are able to distinguish between different homonymous senses, we also remove from our test set those instances that have a single candidate homonymy cluster, for which there would be only one possible prediction, adding no information to our analysis. As a result, the number of remaining test instances is 609.

We show in Table 5 the results of this experiment. Despite the difference in performance between different combinations of models and distance measures, in all experiments, we get an accuracy that is above 89%, with peaks of more than 95% (obtained by BERT). Interestingly, the Most Frequent Sense baseline[15] achieves a disambiguation accuracy score of 84.40, which is more than 10 points below our best distance-based Homonymy Disambiguation system. This suggests that, although PLMs are not explicitly trained to disambiguate homonyms during the pretraining phase, working with vast amounts of textual data frequently exposes them to situations where homonymous words are used. We hypothesize that, over time, the models learn the patterns and contexts, and acquire the capabilities to differentiate between different (macro) meanings of the same word. Consequently, the high results obtained by distance-based approaches indicate that PLMs are able to produce vector representations of words in context that are easily separable in their vector spaces, at least at a coarse-grained level.

For completeness, in Table 8, we provide some examples of correct and wrong predictions of the best distance-based Homonymy Disambiguation model (i.e. BERT).

4.2. Are WSD Systems Homonymy Disambiguators?

With the aim of investigating the relationship between Homonymy Disambiguation and WSD, we test the capabilities of a system trained for the latter task in dealing with homonyms. More specifically, we map the fine-grained senses predicted by a WSD system to their homonymy clusters,[16] so that

[13] We choose to use Euclidean distance together with cosine distance because the latter accounts only for the angle between two vectors, but the actual magnitude of a vector's components could also be of some relevance to the disambiguation task.

[14] The contextualized embeddings are extracted from the last hidden layer of the PLM. We use the first subword embedding when the word has been split by the tokenizer.

[15] For each (lemma, PoS), such baseline selects the most frequent sense among its candidates as measured in the training set.

[16] In our resource, for each (lemma, PoS) we have its set of candidate coarse-grained senses, which are groups of fine-grained senses obtained through clus-

we can measure its performance on the task of Homonymy Disambiguation. Furthermore, we compare its performance with a system that has been trained specifically for Homonymy Disambiguation. Both systems are based on a BERT model,[17] which is used to extract contextualized representations of words in context.

Our architecture is inspired by Conia and Navigli (2021). Specifically, we encode a word in context w with BERT, concatenate the hidden states of the last 4 layers of the encoder, and then apply batch normalization (Ioffe and Szegedy, 2015) to obtain $\boldsymbol{e}_w \in \mathbb{R}^d$.[18] Then, we use a two-layer fully-connected neural network to predict the sense of the word in context. More formally:

$$\begin{aligned}
\boldsymbol{e}_w &= \mathrm{BatchNorm}\left(\boldsymbol{l}_w^{-1} \oplus \boldsymbol{l}_w^{-2} \oplus \boldsymbol{l}_w^{-3} \oplus \boldsymbol{l}_w^{-4}\right) \\
\boldsymbol{h}_w &= \mathrm{Dropout}\left(\mathrm{Swish}(W_h \boldsymbol{e}_w + \boldsymbol{b}_h)\right) \\
\boldsymbol{o}_w &= \boldsymbol{W}_o \boldsymbol{h}_w + \boldsymbol{b}_o
\end{aligned}$$

where $\boldsymbol{l}_w^{-i}$ is the hidden state of the $i-th$ layer of the Transformer starting from its topmost layer, $\mathrm{BatchNorm}(\cdot)$ is the batch normalization operation, $\mathrm{Dropout}(\cdot)$ is the dropout regularization (Srivastava et al., 2014), and $\mathrm{Swish}(x) = x \cdot \mathrm{sigmoid}(x)$ is the Swish activation function (Ramachandran et al., 2017). During training, the pre-trained weights of BERT are kept frozen, in line with the approach taken by Conia and Navigli (2021). We train both models for at most 10 epochs, selecting the checkpoint corresponding to the highest accuracy obtained on the development set in Table 4.[19] We report the results of this experiment in Table 6.

As expected, the two systems have more or less the same performance in all three samples of the test set: considering that Homonymy Disambiguation is easier than WSD, it is reasonable to expect that a model that learns to distinguish fine-grained senses is also capable of implicitly distinguishing the coarser homonymous senses. In addition, looking at the accuracy value obtained in the test set sample containing only polysemous[20] instances, i.e. HA, and considering that the disambiguation model employed is a very simple baseline, we hypothesize that a state-of-the-art WSD system (Barba et al., 2021c) without any modification to the underlying architecture would reach near-perfect performances on the Homonymy Disambiguation task.

Test data	System	WSD	HD
Total	WSD	81.77	**99.23**
	HD	-	99.16
HA	WSD	73.91	**94.13**
	HD	-	93.64
HA$_\mathrm{p}$	WSD	74.06	**96.39**
	HD	-	96.06

Table 6: Performance of Word Sense Disambiguation (WSD) and Homonymy Disambiguation (HD) systems when evaluated on both tasks. The accuracy values are measured on different samples of the test set in Table 4: i) Total represents the entire test set, ii) HA (Homonymy Ambiguous) comprises only those instances that have more than one candidate homonymy cluster, and iii) HA$_\mathrm{p}$ comprises only test instances used in Section 4.1. The best results in HD for each test set sample are in bold.

Finally, as expected, training a Homonymy Disambiguation system allows us to obtain better results compared to a distance-based disambiguation system on the same test set sample used in Section 4.1, i.e. HA$_\mathrm{p}$, since with the former we obtain an accuracy score of 96.06, whereas the best distance-based disambiguation system achieves an accuracy score of 95.24 on that test set sample (see Table 5).

5. Analyses

Our new resource clusters the fine-grained senses of ambiguous words according to homonymy relations. In this section, we conduct a study of this clustering under the lens of the representations generated by BERT and DeBERTa. These PLMs were not only among the top performers in the task of distance-based Homonymy Disambiguation, as reported in Table 5, but also represent the two distinct pretraining strategies outlined in Section 4. Specifically, we leverage the representations of the last hidden layer of these PLMs. We are interested in getting a general sense of how *close* the representations of senses in the same homonymy cluster are, and, contextually, how *far* from each other those of different homonymous senses of the same (lemma, PoS) are. We use all available data, and therefore we concatenate the train, dev, and test splits mentioned in Section 3.3.

Preliminarily, let us define a homonymy cluster $C = \{s_1, s_2, \ldots, s_{K_C}\}$, where s_i is a sense, and K_C is the number of senses that belong to the cluster. A sense $s_i = \{x_1, \ldots, x_N\}$ is the set of vector representations[21] $\boldsymbol{x}_i$, where each $\boldsymbol{x}_i$ is the embedding

tering according to homonymy relations, as outlined in Section 3. Therefore, for every fine-grained sense of a (lemma, PoS) we are able to uniquely determine its homonymy cluster.

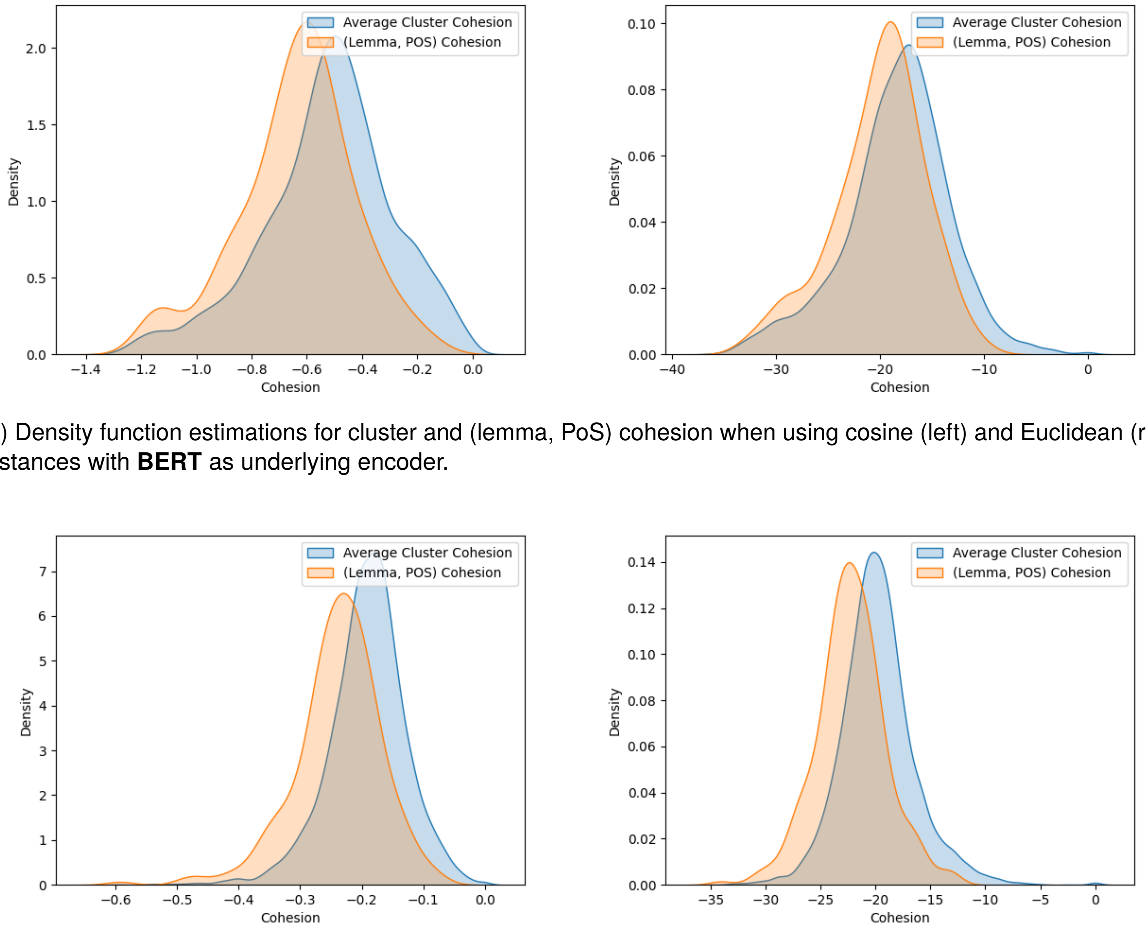
[17] We choose BERT because it is the best-resulting model for distance-based disambiguation in Section 4.1.

[18] Also in this case, we use the first sub-word embedding if the word has been split by the tokenizer.

[19] Please refer to Appendix C for all hyperparameters.

[20] With respect to the Homonymy Disambiguation task.

[21] A sense can be associated with 0, 1 or multiple vec-



(a) Density function estimations for cluster and (lemma, PoS) cohesion when using cosine (left) and Euclidean (right) distances with **BERT** as underlying encoder.

(b) Density function estimations for cluster and (lemma, PoS) cohesion when using cosine (left) and Euclidean (right) distances with **DeBERTa** as underlying encoder.

Figure 2: Density function estimations (Scott, 2012) for cluster and (lemma, PoS) cohesion.

#(lemma, PoS) = 151		$\mathrm{LC} > \mathrm{CC}_{\mathrm{AVG}}$	$\mathrm{LC} < \mathrm{CC}_{\mathrm{AVG}}$
BERT	cosine	28 (18.54%)	123 (81.46%)
	euclidean	36 (23.84%)	115 (76.16%)
DeBERTa	cosine	35 (23.18%)	116 (76.82%)
	euclidean	31 (20.53%)	120 (79.47%)

Table 7: Number of times that, for a given (lemma, PoS), its cohesion is greater than the average cohesion of its clusters, and vice versa.

of an instance tagged with s_i. We define two new measures of distance:

1. **Sense distance**, i.e. the average distance between all representations of two senses s_i and s_j:

$$\mathrm{SD}(s_i, s_j) = \frac{1}{|s_i||s_j|} \sum_{\boldsymbol{x}_v \in s_i, \boldsymbol{x}_q \in s_j} D(\boldsymbol{x}_v, \boldsymbol{x}_q).$$

Where $D(\boldsymbol{x}_v, \boldsymbol{x}_q)$ is a distance measure between vectors, such as the cosine and Euclidean distances.

2. **Cluster distance**, that is, the average distance between all senses of two clusters C_i and C_j:

$$\mathrm{CD}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{s_v \in C_i, s_q \in C_j} \mathrm{SD}(s_v, s_q).$$

We can now introduce the two metrics that we use to conduct our analysis:

1. **Cluster Cohesion** is the opposite of the average distance between all pairs of senses that participate in the same homonymy cluster C:

$$\mathrm{CC}(C) = -\frac{2}{|C|(|C|-1)} \sum_{s_i, s_j \in C, i<j} \mathrm{SD}(s_i, s_j);$$

2. **(lemma, PoS) Cohesion** is the negative of the average distance between all pairs of clusters in the candidates of a (lemma, PoS) l:

tor representations x_i, depending on the number of its instances.

Example	Prediction	Ground Truth	Cosine
If you can't tell, get help from your county agricultural agent or other local specialist. **Soil** type, drainage, or degree of slope can make the difference between good crops and poor ones.	**grime.n.h.01**	soil.n.h.01	0.3254
From the equilibrium sorption data which are available, it seems logical to expect that polyphosphate ions would be strongly sorbed on the surface of the dirt (especially clay **soils**) so as to give it a greatly increased negative charge.	**soil.n.h.01**	soil.n.h.01	0.1738
Erosion **listed** the old tree.	**list.v.h.01**	list.v.h.02	0.7638
The Office of Minerals Exploration (OME) of the U.S. Department of the Interior offers financial assistance to firms and individuals who desire to explore their properties or claims for 1 or more of the 32 mineral commodities **listed** in the OME regulations.	**list.v.h.01**	list.v.h.01	0.2644

Table 8: Examples of correct and wrong predictions of the best distance-based Homonymy Disambiguation model (i.e. BERT), together with the cosine distance between the instance and the closest sense. To give a clearer perspective of the model's behavior, we provide the relevant homonymy clusters, together with their corresponding fine-grained synsets and definitions, in Table 1.

$$\mathrm{LC}(L) = -\frac{2}{|L|(|L|-1)} \sum_{C_i, C_j \in L, i<j} \mathrm{CD}(C_i, C_j).$$

Where, given a (lemma, PoS) pair l, $L = \{C_1, \ldots, C_M\}$ is the set of its candidate homonymy clusters.

The range of values that the cohesion metrics can assume depends on whether we use the cosine or Euclidean distance to measure the distance between vector representations. More specifically, with cosine distance as the underlying distance measure, cohesion metrics range in $[-2, 0]$, while with Euclidean the range becomes $(-\infty, 0]$. In both cases, high cohesion implies that the vectors lie in a narrow subspace, while low cohesion indicates that they are more spread.

Leveraging these metrics, we can further validate the coherence of our resource introduced in Section 3, and investigate how the cohesion of individual clusters relates to the overall (lemma, PoS) cohesion, as well as the distribution of cluster and (lemma, PoS) cohesion measures. Specifically, we expect BERT and DeBERTa to encode senses of the same homonymy cluster into embedding vectors that are closer to each other, compared to those of senses belonging to other candidate clusters of the same (lemma, PoS) pair. More formally, given a (lemma, PoS) pair l, we expect its cohesion $\mathrm{LC}(L)$ to be smaller than the average cohesion of its candidate sense clusters, that is $\mathrm{CC}_{\mathrm{AVG}}(L) = \frac{1}{|L|} \sum_{C \in L} \mathrm{CC}(C)$. For each (lemma, PoS), we measure its cohesion, and compare it to the average cohesion of its clusters.

We show in Figure 2 the cohesion distributions when extracting embedding vectors with BERT and DeBERTa, and using both cosine and Euclidean distances. In each sub-figure, the average cluster cohesion distribution is shifted to the right of that of the (lemma, PoS) cohesion: this indicates that, by means of our homonymy-based clustering, we group together senses that, on average, are closer to each other. Moreover, Table 7 shows how many times the cohesion of a given (lemma, PoS) is greater than its average cluster cohesion, and vice versa. In most cases, $\mathrm{CC}_{\mathrm{AVG}}$ is greater than LC, confirming previous results with a quantitative measurement.

6. Conclusions

In this paper, we propose an approach for clustering WordNet senses based on homonymy relations. We manually map WordNet senses to their most appropriate coarse-grained senses in the Oxford Dictionary of English, and put forward an automatic technique for extending this mapping to every (lemma, PoS) pair in WordNet. As the outcome, we are able to release a comprehensive, high-quality resource, which enables a substantial reduction of the granularity of WordNet. Contextually, we study whether current PLMs can separate homonymous senses, probing their representations by means of cosine and Euclidean distance measures. Our experiments, while unavoidably limited in terms of (lemma, PoS) coverage, suggest that this is indeed the case, with accuracy scores as high as 95%.

Furthermore, we demonstrate that a straightforward baseline system, trained for the more challenging WSD task, achieves remarkable results in Homonymy Disambiguation, attaining an accuracy score exceeding 94% on the test set sample containing only instances that are polysemous in homonymy. This achievement hints at the potential for state-of-the-art WSD systems to achieve near-perfect accuracy scores in Homonymy Disambiguation, without any modifications to their architecture.

Finally, we qualitatively analyze the resulting clustering through the lens of the representations generated by BERT and DeBERTa, showing that our resource clusters together senses that are closer (in terms of cosine and Euclidean distances) to each other than to other senses of the same (lemma, PoS).

7. Acknowledgements

We gratefully acknowledge the support of the PNRR MUR project PE0000013-FAIR, and the CREATIVE project (CRoss-modal understanding and gEnerATIon of Visual and tExtual content), which is funded by the MUR Progetti di Rilevante Interesse Nazionale programme (PRIN 2020). This article is also based upon work from COST Action NexusLinguarum – "European network for Web-centered linguistic data science" (CA18209), supported by COST (European Cooperation in Science and Technology) `www.cost.eu`.

This work has been carried out while Lorenzo Proietti and Simone Tedeschi were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome.

8. Bibliographical References

Eneko Agirre. 2004. Clustering of word senses. *Proc. of the 2nd Global WordNet Conference, panel on figurative language. Brno, Czech Republic. http://www.fi.muni.cz/gwc2004/proc/index.html © Masaryk University Brno. ISBN: 80-210-3302-9*.

Laura Aina, Kristina Gulordava, and Gemma Boleda. 2019. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348, Florence, Italy. Association for Computational Linguistics.

Reinald Kim Amplayo, Seung-won Hwang, and Min Song. 2019. Autosense model for word sense induction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6212–6219.

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels, Belgium. Association for Computational Linguistics.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, page 805–810, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021a. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Edoardo Barba, Luigi Procopio, Caterina Lacerra, Tommaso Pasini, and Roberto Navigli. 2021b. Exemplification modeling: Can you give me an example, please? In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3779–3785. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021c. ConSeC: Word sense disambiguation as continuous sense comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1492–1503, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 33–40, Prague, Czech Republic. Association for Computational Linguistics.

Timothy Chklovski and Rada Mihalcea. 2003. Exploiting agreement and disagreement of human annotators for word sense disambiguation. In *Recent Advances in Natural Language Processing*.

Irina Chugur, Julio Gonzalo, and Felisa Verdejo. 2002. A study of polysemy and sense proximity in the senseval-2 test suite. In *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 32–39. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Simone Conia and Roberto Navigli. 2021. Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3269–3275, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan. 1994. Word sense ambiguation: Clustering related senses. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*.

JL Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378—382.

J.L. Fleiss, B. Levin, and M.C. Paik. 2013. *Statistical Methods for Rates and Proportions*. Wiley Series in Probability and Statistics. Wiley.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, pages 19–33, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

Adam Kilgarriff and Martha Palmer. 2000. Introduction to the special issue on senseval. *Computers and the Humanities*, 34:1–13.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Caterina Lacerra, Michele Bevilacqua, Tommaso Pasini, and Roberto Navigli. 2020. Csi: A coarse sense inventory for 85% word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8123–8130.

Beth Levin. 1993. English verb classes and alternations: A preliminary investigation.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Dekang Lin. 1998b. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2021. On the variance of the adaptive learning rate and beyond.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2021. Analysis and evaluation of language models for word sense disambiguation. *Computational Linguistics*, 47(2):387–443.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli. 2022. Nibbling at the hard core of Word Sense Disambiguation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4724–4737, Dublin, Ireland. Association for Computational Linguistics.

Rowan Maudslay and Simone Teufel. 2022. Homonymy information for English WordNet. In *Proceedings of Globalex Workshop on Linked Lexicography within the 13th Language Resources and Evaluation Conference*, pages 90–98, Marseille, France. European Language Resources Association.

Diana McCarthy. 2006. Relating WordNet senses for word sense disambiguation. In *Proceedings of the Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*.

Rada Mihalcea and Dan I. Moldovan. 2001. Automatic generation of a coarse grained wordnet. In *HTL 2001*.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Roberto Navigli. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 105–112, Sydney, Australia. Association for Computational Linguistics.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. SemEval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, Prague, Czech Republic. Association for Computational Linguistics.

Martha Palmer, Olga Babko-Malaya, and Hoa Trang Dang. 2004. Different sense granularities for different applications. In *Proceedings of the 2nd International Workshop on Scalable Natural Language Understanding (ScaNaLU 2004) at HLT-NAACL 2004*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24, Toulouse, France. Association for Computational Linguistics.

Endang Wahyu Pamungkas and Divi Galih Prasetyo Putri. 2017. Word sense disambiguation for lexicon-based sentiment analysis. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, ICMLC '17, page 442–446, New York, NY, USA. Association for Computing Machinery.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity - measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41, Boston, Massachusetts, USA. Association for Computational Linguistics.

Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, Ohio, USA. Association for Computational Linguistics.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

James Pustejovsky and Peter Paul Buitelaar. 1998. Corelex: systematic polysemy and underspecification.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. Searching for activation functions. *CoRR*, abs/1710.05941.

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 *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, page 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Mark Sanderson. 1994. Word sense disambiguation and information retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 142–151. Springer.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

David W. Scott. 2012. Multivariate density estimation and visualization.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Catherine Soanes and Angus Stevenson, editors. 2003. *Oxford Dictionary of English*. Oxford University Press.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Chiraag Sumanth and Diana Inkpen. 2015. How much does word sense disambiguation help in sentiment analysis of micropost data? In *Proceedings of the 6th Workshop on Computational*

Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 115–121, Lisboa, Portugal. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5831–5837, Hong Kong, China. Association for Computational Linguistics.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference*, pages 108–117, Wroclaw, Poland. Global Wordnet Association.

Dayu Yuan, Ryan Doherty, Julian Richardson, Colin Evans, and Eric Altendorf. 2016a. Word sense disambiguation with neural language models. *CoRR*, abs/1603.07012.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016b. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhi Zhong and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282, Jeju Island, Korea. Association for Computational Linguistics.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.

9. Language Resource References

Miller, George A. 1994. *WordNet: A Lexical Database for English*. ISLRN 379-473-059-273-1.

A. Resource Extension

Examples of (lemma, PoS) pairs in WN_r can be seen in Table 9.

B. Data

Additional statistics about the data used in this work are presented in Table 10.

C. Hyperparameters

For both trained disambiguation systems we employ the same set of hyperparameters, detailed in Table 11.

(lemma, PoS)	**Homonym**	**Synset**	**Definition**

(acidity, NOUN)	`acidity.n.h.01`	`acidity.n.02`
		`acidity.n.03`
		`sourness.n.02`
			the taste experience when something acidic is taken into the mouth
			pH values below 7
			the property of being acidic

(schematize, VERB)	`schematize.v.h.01`	`schematize.v.01` `schematize.v.02`	formulate in regular order; to reduce to a scheme or formula give conventional form to

Table 9: For each (lemma, PoS) pair, there are its homonyms in the form `lemma.PoS.h.id`, which are coarse-grained senses, each grouping together one or more WordNet senses.

	sents	lemmas	(lemma, PoS)
SemCor	36,298	20,399	22,436
SE7	120	327	330
ALL$_{NEW}$	951	1701	1810
WN Examples	47,269	22,482	24,437
Train	81,938	31,820	35,084
Dev	900	3576	3822
Test	1800	5720	6178

Table 10: Number of sentences, distinct lemmas, and distinct (lemma, PoS) pairs in each set.

Hyperparameter	**Value**
Optimizer	RAdam (Liu et al., 2021)
Learning Rate	1e-4
Batch Size	8
Accumulation Steps	2
Dropout	0.1
Dimension of $\boldsymbol{h}_w$	512

Table 11: Hyperparameters used for training the two disambiguation systems.