Analysis and Evaluation of Language Models for Word Sense Disambiguation

Daniel Loureiro*
LIAAD - INESC TEC
Department of Computer Science - FCUP
University of Porto
dloureiro@fc.up.pt

Kiamehr Rezaee*
Department of Computer Engineering
Iran University of Science and Technology
k_rezaee@comp.iust.ac.ir

Mohammad Taher Pilehvar
Tehran Institute for Advanced Studies
mp792@cam.ac.uk

Jose Camacho-Collados
School of Computer Science and Informatics
Cardiff University
camachocolladosj@cardiff.ac.uk

Transformer-based language models have taken many fields in NLP by storm. BERT and its derivatives dominate most of the existing evaluation benchmarks, including those for Word Sense Disambiguation (WSD), thanks to their ability in capturing context-sensitive semantic nuances. However, there is still little knowledge about their capabilities and potential limitations in encoding and recovering word senses. In this article, we provide an in-depth quantitative and qualitative analysis of the celebrated BERT model with respect to lexical ambiguity. One of the main conclusions of our analysis is that BERT can accurately capture high-level sense distinctions, even when a limited number of examples is available for each word sense. Our analysis also reveals that in some cases language models come close to solving coarse-grained noun disambiguation under ideal conditions in terms of availability of training data and computing resources. However, this scenario rarely occurs in real-world settings and, hence, many practical

*These authors contributed equally to this work.

Submission received: 19 August 2020; revised version received: 15 February 2021; accepted for publication: 4 March 2021.

https://doi.org/10.1162/COLI_a_00405

challenges remain even in the coarse-grained setting. We also perform an in-depth comparison of the two main language model-based WSD strategies, namely, fine-tuning and feature extraction, finding that the latter approach is more robust with respect to sense bias and it can better exploit limited available training data. In fact, the simple feature extraction strategy of averaging contextualized embeddings proves robust even using only three training sentences per word sense, with minimal improvements obtained by increasing the size of this training data.

1. Introduction

In the past decade, word embeddings have undoubtedly been one of the major points of attention in research on lexical semantics. The introduction of Word2vec (Mikolov et al. 2013b), as one of the pioneering word *embedding* models, generated a massive wave in the field of lexical semantics, the impact of which is still being felt today. However, static word embeddings (such as Word2vec) suffer from the limitation of being *fixed* or context insensitive, that is, the word is associated with the same representation in all contexts, disregarding the fact that different contexts can trigger various meanings of the word, which might be even semantically unrelated. Sense representations were an attempt at addressing the meaning conflation deficiency of word embeddings (Reisinger and Mooney 2010; Camacho-Collados and Pilehvar 2018). Despite computing distinct representations for different senses of a word, hence addressing this deficiency of word embeddings, sense representations are not directly integrable into downstream NLP models. The integration usually requires additional steps, including a (non-optimal) disambiguation of the input text, which make sense embeddings fall short of fully addressing the problem.

The more recent *contextualized* embeddings (Peters et al. 2018a; Devlin et al. 2019) are able to simultaneously address both these limitations. Trained with language modeling objectives, contextualized models can compute *dynamic* meaning representations for words in context that highly correlate with humans' word sense knowledge (Nair, Srinivasan, and Meylan 2020). Moreover, contextualized embeddings provide a seamless integration into various NLP models, with minimal changes involved. Even better, given the extent of semantic and syntactic knowledge they capture, contextualized models get close to the one system for all tasks settings. Surprisingly, fine-tuning the same model on various target tasks often results in comparable or even higher performance when compared with sophisticated state-of-the-art task-specific models (Peters, Ruder, and Smith 2019). This has been shown for a wide range of NLP applications and tasks, including Word Sense Disambiguation (WSD), for which they have provided a significant performance boost, especially after the introduction of Transformer-based language models like BERT (Loureiro and Jorge 2019a; Vial, Lecouteux, and Schwab 2019; Wiedemann et al. 2019).

Despite their massive success, there has been limited work on the analysis of recent language models and on explaining the reasons behind their effectiveness in lexical semantics. Most analytical studies focus on syntax (Hewitt and Manning 2019; Saphra and Lopez 2019) or explore the behavior of self-attention heads (Clark et al. 2019) or layers (Tenney, Das, and Pavlick 2019), but there has been little work on investigating the potential of language models and their limitations in capturing other linguistic aspects, such as lexical ambiguity. Moreover, the currently popular language understanding evaluation benchmarks—for example, GLUE (Wang et al. 2018) and SuperGLUE (Wang et al. 2019)—mostly involve sentence-level representation, which does not shed much

light on the semantic properties of these models for individual words.[1] To our knowledge, there has so far been no in-depth analysis of the abilities of contextualized models in capturing the ambiguity property of words.

In this article, we carry out a comprehensive analysis to investigate how pretrained language models capture lexical ambiguity in the English language. Specifically, we scrutinize the two major language model-based WSD strategies (i.e., feature extraction and fine-tuning) under various disambiguation scenarios and experimental configurations. The main contributions of this work can be summarized as follows: (1) we provide an extensive quantitative evaluation of pretrained language models in standard WSD benchmarks; (2) we develop a new data set, CoarseWSD-20, which is particularly suited for the qualitative analysis of WSD systems; and (3) with the help of this data set, we perform an in-depth qualitative analysis and test the limitations of BERT on coarse-grained WSD. Data and code to reproduce all our experiments is available at `https://github.com/danlou/bert-disambiguation`.

The remainder of the article is organized as follows. In Section 2, we delineate the literature on probing pretrained language models and on analyzing the potential of representation models in capturing lexical ambiguity. We also describe in the same section the existing benchmarks for evaluating WSD. Section 3 presents an overview of WSD and its conventional paradigms. We then describe in the same section the two major approaches to utilizing language models for WSD, namely, nearest-neighbor feature extraction and fine-tuning. We also provide a quantitative comparison of some of the most prominent WSD approaches in each paradigm in various disambiguation scenarios, including fine- and coarse-grained settings. This quantitative analysis is followed by an analysis of models' performance per word categories (parts of speech) and for various layer-wise representations (in the case of language model-based techniques). Section 4 introduces CoarseWSD-20, the WSD data set we have constructed to facilitate our in-depth qualitative analysis. In Section 5 we evaluate the two major BERT-based WSD strategies on the benchmark. To highlight the improvement attributable to contextualized embeddings, we also provide results of a linear classifier based on pretrained FastText static word embeddings. Based on these experiments, we carry out an analysis on the impact of fine-tuning and also compare the two strategies with respect to robustness across domains and bias toward the most frequent sense. Section 6 reports our observations upon further scrutinizing the two strategies on a wide variety of settings such as few-shot learning and different training distributions. Section 7 summarizes the main results from the previous sections and discusses the main takeaways. Finally, Section 8 presents the concluding remarks and potential areas for future work.

2. Related Work

Recently, there have been several attempts at analyzing pretrained language models. In Section 2.1 we provide a general overview of the relevant works, and Section 2.2 covers those related to lexical ambiguity. Finally, in Section 2.3 we outline existing evaluation benchmarks for WSD, including CoarseWSD-20, which is the disambiguation data set we have constructed for our qualitative analysis.

1 WiC (Pilehvar and Camacho-Collados 2019) is the only SuperGLUE task where systems need to model the semantics of words in context (extended to several more languages in XL-WiC [Raganato et al. 2020]). In the Appendix we provide results for this task.

2.1 Analysis of Pretrained Language Models

Despite their young age, pretrained language models, in particular, those based on Transformers, have now dominated the evaluation benchmarks for most NLP tasks (Devlin et al. 2019; Liu et al. 2019b). However, there has been limited work on understanding behind the scenes of these models.

Various studies have shown that fulfilling the language modeling objective inherently forces the model to capture various linguistic phenomena. A relatively highly studied phenomenon is syntax, which is investigated both for earlier LSTM-based models (Linzen, Dupoux, and Goldberg 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, Sagot, and Seddah 2019; van Schijndel, Mueller, and Linzen 2019; Tenney et al. 2019). A recent work in this context is the **probe** proposed by Hewitt and Manning (2019), which enabled them to show that Transformer-based models encode human-like parse trees to a very good extent. In terms of semantics, fewer studies exist, including the probing study of Ettinger (2020) on semantic roles, and that of Tenney, Das, and Pavlick (2019), which also investigates entity types and relations. The closest analysis to ours is that of Peters et al. (2018b), which provides a deep analysis of contextualized word embeddings, both from the representation point of view and per architectural choices. In the same spirit, Conneau et al. (2018) proposed a number of linguistic probing tasks to analyze sentence embedding models. Perhaps more related to the topic of this article, Shwartz and Dagan (2019) showed how contextualized embeddings are able to capture non-literal usages of words in the context of lexical composition. For a complete overview of existing probe and analysis methods, the survey of Belinkov and Glass (2019) provides a synthesis of analysis studies on neural network methods. The more recent survey of Rogers, Kovaleva, and Rumshisky (2020) is a similar synthesis but targeted at BERT and its derivatives.

Despite all this analytical work, the investigation of neural language models from the perspective of ambiguity (and, in particular, lexical ambiguity) has been surprisingly neglected. In the following we discuss studies that aimed at shedding some light on this important linguistic phenomenon.

2.2 Lexical Ambiguity and Language Models

Given its importance, lexical ambiguity has for long been an area of investigation in vector space model representations (Schütze 1993; Reisinger and Mooney 2010; Camacho-Collados and Pilehvar 2018). In a recent study on word embeddings, Yaghoobzadeh et al. (2019) showed that Word2vec (Mikolov et al. 2013a) can effectively capture different coarse-grained senses if they are all frequent enough and evenly distributed. In this work we try to extend this conclusion to a language model-based representation and to the more realistic scenario of disambiguating words in context, rather than probing them in isolation for if they capture specific senses (as was the case in that work).

Most of the works analyzing language models and lexical ambiguity have opted for lexical substitution as their experimental benchmark. Amrami and Goldberg (2018) showed that an LSTM language model can be effectively applied to the task of word sense induction. In particular, they analyzed how the predictions of an LSTM for a word in context provided a useful way to retrieve substitutes, proving that this information is indeed captured in the language model. From a more analytical point of view, Aina, Gulordava, and Boleda (2019) proposed a probe task based on lexical substitution to understand the internal representations of an LSTM language model for predicting

words in context. Similarly, Soler et al. (2019) provided an analysis of LSTM-based contextualized embeddings in distinguishing between usages of words in context. As for 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 find senses of a word in context. While lexical substitution has been shown to be an interesting proxy for WSD, we provide a direct and in-depth analysis of the explicit capabilities of recent language models in encoding lexical ambiguity, both quantitatively and qualitatively.

Another related work to ours is the analysis of Reif et al. (2019) on quantifying the geometry of BERT. The authors observed that, generally, when contextualized BERT embeddings for ambiguous words are visualized, clear clusters for different senses are identifiable. They also devised an experiment to highlight a potential failure with BERT (or presumably other attention-based models): It does not necessarily respect semantic boundaries when attending to neighboring tokens. In our qualitative analysis in Section 6.4 we further explore this. Additionally, Reif et al. (2019) present evidence supporting the specialization of representations from intermediate layers of BERT for sense representation, which we further confirm with layer-wise WSD evaluation in Section 3.4.5. Despite these interesting observations, their paper mostly focuses on the syntactic properties of BERT, similarly to most other studies in the domain (see Section 2.1).

Finally, a few works have attempted to induce semantic priors coming from knowledge resources like WordNet to improve the generalization of pretrained language models like BERT (Levine et al. 2020; Peters et al. 2019). Other works have investigated BERT's emergent semantic space using clustering analyses (Yenicelik, Schmidt, and Kilcher 2020; Chronis and Erk 2020), seeking to characterize how distinct sense-specific representations occupy this space.

Our work differs in that we are trying to understand to what extent pretrained language models already encode this semantic knowledge and, in particular, what are their implicit practical disambiguation capabilities.

2.3 Evaluation Benchmarks

The most common evaluation benchmarks for WSD are based on fine-grained resources, with WordNet (Fellbaum 1998) being the de facto sense inventory. For example, the unified all-words WSD benchmark of Raganato, Camacho-Collados, and Navigli (2017) is composed of five data sets from Senseval/SemEval tasks: Senseval-2 (Edmonds and Cotton 2001, SE02), Senseval-3 (Mihalcea, Chklovski, and Kilgarriff 2004, SE03), SemEval-2007 (Agirre, Màrquez, and Wicentowski 2007, SE07), SemEval-2013 (Navigli, Jurgens, and Vannella 2013, SE13), and SemEval-2015 (Moro and Navigli 2015, SE15). Vial, Lecouteux, and Schwab (2018) extended this framework with other manually and automatically constructed data sets.[2] All these data sets are WordNet-specific and mostly use SemCor (Miller et al. 1993) as their training set. Despite being the largest WordNet-based sense-annotated data set, SemCor does not cover many senses occurring in the test sets, besides providing a limited number of examples per sense. Although scarcity in the training data is certainly a realistic setting, in this article we are interested in analyzing the limits of language models with and without training data, also for senses not included in WordNet, and run a qualitative analysis.

2 Pasini and Camacho-Collados (2020) provide an overview of existing sense-annotated corpora for WordNet and other resources.

To this end, in addition to running an evaluation in standard benchmarks, for this article we constructed a coarse-grained WSD data set, called CoarseWSD-20. CoarseWSD-20 includes a selection of 20 ambiguous words of different nature (see Section 4 for more details on CoarseWSD-20) where we run a qualitative analysis on various aspects of sense-specific information encoded in language models. Perhaps the closest data sets to CoarseWSD-20 are those of Lexical Sample WSD (Edmonds and Cotton 2001; Mihalcea, Chklovski, and Kilgarriff 2004; Pradhan et al. 2007). These data sets usually target dozens of ambiguous words and list specific examples for their different senses. However, these examples are usually fine-grained, limited in number,[3] and are limited to concepts (i.e., no entities such as *Java* are included). The CoarseWSD-20 data set is similar in spirit, but has larger training sets extracted from Wikipedia. Constructing the data set based on the sense inventory of Wikipedia brings the additional advantage of having both entities and concepts as targets, and a direct mapping to Wikipedia pages, which is the most common resource for entity linking (Ling, Singh, and Weld 2015; Usbeck et al. 2015), along with similar inter-connected resources such as DBpedia.

Another related data set to CoarseWSD-20 is WIKI-PSE (Yaghoobzadeh et al. 2019). Similarly to ours, WIKI-PSE is constructed based on Wikipedia, but with a different purpose. WIKI-PSE clusters all Wikipedia concepts and entities into eight general "semantic classes." This is an extreme coarsening of the sense inventory that may not fully reflect the variety of human-interpretable senses that a word has. Instead, for CoarseWSD-20, sense coarsening is performed at the word level, which preserves sense-specific information. For example, the word *bank* in WIKI-PSE is mainly identified as a *location* only, conflating the financial institution and river meanings of the word, whereas CoarseWSD-20 distinguishes between the two senses of *bank*. Moreover, our data set is additionally post-processed in a semi-automatic manner (an automatic preprocessing, followed by a manual check for problematic cases), which helps remove errors from the Wikipedia dump.

3. Word Sense Disambiguation: An Overview

Our analysis is focused on the task of word sense disambiguation. WSD is a core module of human cognition and a long-standing task in NLP. Formally, given a word in context, the task of WSD consists of selecting the intended meaning (sense) from a predefined set of senses for that word defined by a sense inventory (Navigli 2009). For example consider the word *star* in the following context:

- Sirius is the brightest *star* in Earth's night.

The task of a WSD system is to identify that the usage of *star* in this context refers to its astronomical meaning (as opposed to celebrity or star shape, among others). The context could be a document, a sentence, or any other information-carrying piece of text that can provide a hint on the intended semantic usage,[4] probably as small as a word, for example, "*dwarf* star."[5]

3 For instance, the data set of Pradhan et al. (2007), which is the most recent and the largest among the three mentioned lexical sample data sets, provides an average of 320/50 training/test instances for each of the 35 nouns in the data set. In contrast, CoarseWSD-20 includes considerably larger data sets for all words (1,160 and 510 sentences on average for each word in the training and test sets, respectively).

4 For this analysis we focus on sentence-level WSD, because it is the most standard practice in the literature.

5 A *dwarf star* is a relatively small star with low luminosity, such as the Sun.

WSD is described as an AI-hard[6] problem (Mallery 1988). In a comprehensive survey of WSD, Navigli (2009) discusses some of the reasons behind its difficulty, including heavy reliance on knowledge, difficulty in distinguishing fine-grained sense distinctions, and lack of application to real-world tasks. On WordNet-style sense inventories, the human-level performance (which is usually quoted as glass ceiling) is estimated to be 80% in the fine-grained setting (Gale, Church, and Yarowsky 1992a) and 90% for the coarse-grained one (Palmer, Dang, and Fellbaum 2007). This performance gap can be mainly attributed to the fine-grained semantic distinctions in WordNet that are sometimes even difficult for humans to distinguish. For instance, the noun *star* has 8 senses in WordNet 3.1, two of which refer to the astronomical sense (celestial body) with the minor semantic difference of if the star is visible from Earth at night. In fact, it is argued that sense distinctions in WordNet are too fine-grained for many NLP applications (Hovy, Navigli, and Ponzetto 2013). CoarseWSD-20 addresses this issue by devising sense distinctions that are easily interpretable by humans, essentially pushing the human performance on the task.

Similarly to many other tasks in NLP, WSD has gone under significant change after the introduction of Transformer-based language models, which are now dominating most WSD benchmarks. In the following we first present a background on existing sense inventories, with a focus on WordNet (Section 3.1), and then describe the state of the art in both the conventional paradigm (Section 3.2) and the more recent paradigm based on (Transformer-based) language models (Section 3.3). We then carry out a quantitative evaluation of some of the most prominent WSD approaches in each paradigm in various disambiguation scenarios, including fine- and coarse-grained settings (Section 3.4). This quantitative analysis is followed by an analysis of layer-wise representations (Section 3.4.5) and performance per word categories (parts of speech, Section 3.4.6).

3.1 Sense Inventories

Given that WSD is usually tied with sense inventories, we briefly describe existing sense inventories that are also used in our experiments. The main sense inventory for WSD research in English is the Princeton WordNet (Fellbaum 1998). The basic constituents of this expert-made lexical resource are **synsets**, which are sets of synonymous words that represent unique concepts. A word can belong to multiple synsets denoting to its different meanings. Version 3.0 of the resource, which is used in our experiments, covers 147,306 words and 117,659 synsets.[7] WordNet is also available for languages other than English through the Open Multilingual WordNet project (Bond and Foster 2013) and related efforts.

Other common-sense inventories are Wikipedia and BabelNet. The former is generally used for Entity Linking or **Wikification** (Mihalcea and Csomai 2007), in which the Wikipedia pages are considered as concept or entities to be linked in context. On the other hand, BabelNet (Navigli and Ponzetto 2012) is a merger of WordNet, Wikipedia, and several other lexical resources, such as Wiktionary and OmegaWiki. One of the key

6 By analogy to NP-completeness, the most difficult problems are referred to as AI-complete, implying that solving them is equivalent to solving the central artificial intelligence problem.

7 There are several other variants of WordNet available, either the newer v3.1, which is slightly different from the former version, or other non-Princeton versions that improve coverage, such as WordNet 2020 (McCrae et al. 2020) or CROWN (Jurgens and Pilehvar 2015). We opted for v3.0 given that it is the widely used inventory according to which most existing benchmarks are annotated.

features of this resource is its multilinguality, highlighted by the 500 languages covered in its most recent release (version 5.0).

3.2 WSD Paradigms

WSD approaches are traditionally categorized as **knowledge-based** and **supervised**. The latter makes use of sense-annotated data for its training whereas the former exploits sense inventories, such as WordNet, for the encoded knowledge, such as sense glosses (Lesk 1986; Banerjee and Pedersen 2003; Basile, Caputo, and Semeraro 2014), semantic relations (Agirre, de Lacalle, and Soroa 2014; Moro, Raganato, and Navigli 2014), or sense distributions (Chaplot and Salakhutdinov 2018). Supervised WSD has been shown to clearly outperform the knowledge-based counterparts, even before the introduction of pretrained language models (Raganato, Camacho-Collados, and Navigli 2017). Large pretrained language models have further provided improvements, with BERT-based models currently approaching human-level performance (Loureiro and Jorge 2019a; Vial, Lecouteux, and Schwab 2019; Huang et al. 2019; Bevilacqua and Navigli 2020; Blevins and Zettlemoyer 2020). A third category of WSD techniques, called **hybrid**, has recently attracted more attention. In this approach, the model benefits from both sense-annotated instances and knowledge encoded in sense inventories.[8] Most of the recent state-of-the-art approaches can be put in this category.

3.3 Language Models for WSD

In the context of Machine Translation (MT), a language model is a statistical model that estimates the probability of a sequence of words in a given language. Recently, the scope of LMs has gone far beyond MT and generation tasks. This is partly due to the introduction of Transformers (Vaswani et al. 2017), attention-based neural architectures that have proven immense potential in capturing complex and nuanced linguistic knowledge. In fact, despite their recency, Transformer-based LMs dominate most language understanding benchmarks, such as GLUE (Wang et al. 2018) and SuperGLUE (Wang et al. 2019).

There are currently two popular varieties of Transformer-based Language Models (LMs), differentiated most significantly by their choice of language modeling objective. There are causal (or left-to-right) models, epitomized by GPT-3 (Brown et al. 2020), where the objective is to predict the next word, given the past sequence of words. Alternatively, there are masked models, where the objective is to predict a masked (i.e., hidden) word given its surrounding words, traditionally known as the Cloze task (Taylor 1953), of which the most prominent example is BERT. Benchmark results reported in Devlin et al. (2019) and Brown et al. (2020) show that masked LMs are preferred for semantic tasks, whereas causal LMs are more suitable for language generation tasks. As a potential explanation for the success of BERT-based models, Voita, Sennrich, and Titov (2019) present empirical evidence suggesting that the masked LM objective induces models to produce more generalized representations in intermediate layers.

In our experiments, we opted for the BERT (Devlin et al. 2019) and ALBERT (Lan et al. 2020) models given their prominence and popularity. Nonetheless, our empirical

8 Note that knowledge-based WSD systems might benefit from sense frequency information obtained from sense-annotated data, such as SemCor. Given that such models do not incorporate sense-annotated instances, we do not categorize them as hybrid.

analysis could be applied to other pretrained language models as well (e.g., Liu et al. 2019b; Raffel et al. 2020). Our experiments focus on two dominant WSD approaches based on language models: (1) Nearest Neighbors classifiers based on features extracted from the model (Section 3.3.1), and (2) fine-tuning of the model for WSD classification (Section 3.3.2). In the following we describe the two strategies.

3.3.1 Feature Extraction. Neural LMs have been utilized for WSD, even before the introduction of Transformers, when LSTMs were the first choice for encoding sequences (Melamud, Goldberger, and Dagan 2016; Yuan et al. 2016; Peters et al. 2018a). In this context, LMs were often used to encode the context of a target word, or in other words, generate a contextual embedding for that word. Allowing for various sense-inducing contexts to produce different word representations, these contextual embeddings proved more suitable for lexical ambiguity than conventional word embeddings (e.g., Word2vec).

Consequently, Melamud, Goldberger, and Dagan (2016), Yuan et al. (2016), and Peters et al. (2018a) independently demonstrated that, given sense-annotated corpora (e.g., SemCor), it is possible to compute an embedding for a specific word sense as the average of its contextual embeddings. Sense embeddings computed in this manner serve as the basis for a series of WSD systems. The underlying approach is straightforward: Match the contextual embedding of the word to be disambiguated against its corresponding pre-computed sense embeddings. The matching is usually done using a simple k Nearest Neighbors (NN) (often with $k = 1$) classifier; hence, we refer to this feature extraction approach as **1NN** in our experiments. A simple 1NN approach based on LSTM contextual embeddings proved effective enough to rival the performance of other systems using task-specific training, such as Raganato, Delli Bovi, and Navigli (2017), despite using no WSD specific modeling objectives. Loureiro and Jorge (2019a, LMMS) and Wiedemann et al. (2019) independently showed that the same approach using contextual embeddings from BERT could in fact surpass the performance of those task-specific alternatives. Loureiro and Jorge (2019a) also explored a propagation method using WordNet to produce sense embeddings for senses not present in training data (LMMS_{1024}) and a variant that introduced information from glosses into the same embedding space (LMMS_{2048}). Similar methods have been also introduced for larger lexical resources such as BabelNet, with similar conclusions (Scarlini, Pasini, and Navigli 2020a, SensEmBERT).

There are other methods based on feature extraction that do not use 1NN for making predictions. Vial, Lecouteux, and Schwab (2019, Sense Compression) used contextual embeddings from BERT as input for additional Transformer encoder layers with a softmax classifier on top. Blevins and Zettlemoyer (2020) also experimented with a baseline using the final states of a BERT model with a linear classifier on top. Finally, the solution by Bevilacqua and Navigli (2020) relied on an ensemble of sense embeddings from LMMS and SensEmBERT, along with additional resources, to train a high performance WSD classifier.

3.3.2 Fine-Tuning. Another common approach to benefiting from contextualized language models in downstream tasks is fine-tuning. For each target task, it is possible to simply plug in the task-specific inputs and outputs into pretrained models, such as BERT, and fine-tune all or part of the parameters end-to-end. This procedure adjusts the model's parameters according to the objectives of the target task, for example, the classification task in WSD. One of the main drawbacks of this type of supervised model is their need for building a model for each word, which is unrealistic in practice for

all-words WSD. However, there are several successful WSD approaches in this category that overcome this limitation in different ways. GlossBERT (Huang et al. 2019) uses sense definitions to fine-tune the language model for the disambiguation task, similarly to a text classification task. KnowBERT (Peters et al. 2019) fine-tunes BERT for entity linking exploiting knowledge bases (WordNet and Wikipedia) as well as sense definitions. BEM (Blevins and Zettlemoyer 2020) proposes a bi-encoder method that learns to represent sense embeddings leveraging sense definitions while performing the optimization jointly with the underlying BERT model.

3.4 Evaluation in Standard Benchmarks

In our first experiment, we perform a quantitative evaluation on the unified WSD evaluation framework (Section 3.4.3), which verifies the extent to which a model can distinguish between different senses of a word as defined by WordNet's inventory.

3.4.1 BERT Models. For this task we use a NN strategy (1NN henceforth) that has been shown to be effective with pretrained language models, both for LSTMs and more recently for BERT (see Section 3.3.1). In particular, we used the cased base and large variants of BERT released by Devlin et al. (2019), as well as the xxlarge (v2) variant of ALBERT (Lan et al. 2020), via the Transformers framework (v2.5.1) (Wolf et al. 2020). Following LMMS, we also average sub-word embeddings and represent contextual embeddings as the sum of the corresponding representations from the final four layers. However, here we do not apply the LMMS propagation method aimed at fully representing the sense inventory, resorting to the conventional MFS fallback for lemmas unseen during training.

3.4.2 Comparison Systems. In addition to BERT and ALBERT, we include results for 1NN systems that exploit precomputed sense embeddings, namely, Context2vec (Melamud, Goldberger, and Dagan 2016) and ELMo (Peters et al. 2018a). Moreover, we include results for hybrid systems, namely, supervised models that also make use of additional knowledge sources (cf. Section 3.2), particularly semantic relations and textual definitions in WordNet. Besides the models already discussed in Sections 3.3.1 and 3.3.2, we also report results from additional hybrid models. Raganato, Delli Bovi, and Navigli (2017, Seq2Seq) trained a neural BiLSTM sequence model with losses specific not only to specific senses from SemCor but also part-of-speech tags and WordNet supersenses. EWISE (Kumar et al. 2019), which inspired EWISER (Bevilacqua and Navigli 2020), also uses a BiLSTM to learn contextual representations that can be matched against sense embeddings learned from both sense definitions and semantic relations.

For completeness we also add some of the best linear supervised baselines, namely, IMS (Zhong and Ng 2010) and IMS with embeddings (Zhong and Ng 2010; Iacobacci, Pilehvar, and Navigli 2016, IMS+emb), which are Support Vector Machine (SVM) classifiers based on several manually curated features. Finally, we report results for knowledge-based systems (KB) that mainly rely on WordNet: Lesk_{ext}+emb (Basile, Caputo, and Semeraro 2014), Babelfy (Moro, Raganato, and Navigli 2014), UKB (Agirre, López de Lacalle, and Soroa 2018), and TM (Chaplot and Salakhutdinov 2018). More recently, SyntagRank (Scozzafava et al. 2020) showed best KB results by combining WordNet with the SyntagNet (Maru et al. 2019) database of syntagmatic relations. However, as discussed in Section 3.2, we categorize these as knowledge-based because they do not directly incorporate sense-annotated instances as their source of knowledge.

Table 1
F-Measure performance on the unified WSD evaluation framework (Raganato, Camacho-Collados, and Navigli 2017) for three classes of WSD models (i.e., knowledge-based [KB], supervised, and hybrid), and for two sense specification settings (i.e., fine-grained [FN] and coarse-grained [CS]. Results marked with * make use of SE07/SE15 as development set. Systems marked with † rely on external resources other than WordNet. The results from complete rows were computed by ourselves given the system outputs, while those from incomplete rows were taken from the original papers.

	Type	System	SE2		SE3		SE07		SE13		SE15		ALL	
			FN	CS	FN	CS	FN	CS	FN	CS	FN	CS	FN	CS
KB		$Lesk_{ext}$+emb	63.0	74.9	63.7	75.5	56.7	71.6	66.2	77.4	64.6	73.9	63.7	75.3
		Babelfy†	67.0	78.4	63.5	77.5	51.6	68.8	66.4	77.0	70.3	79.1	65.5	77.3
		TM	69.0	—	66.9	—	55.6	—	65.3	—	69.6	—	66.9	—
		UKB	68.8	81.2	66.1	78.1	53.0	70.8	68.8	79.1	70.3	77.4	67.3	78.7
		SyntagRank	**71.6**	—	**72.0**	—	**59.3**	—	**72.2**	—	**75.8**	—	**71.7**	—
Supervised	SVM	IMS	70.9	81.5	69.3	80.8	61.3	74.3	65.3	77.4	69.5	75.7	68.4	79.1
	SVM	IMS+emb	72.2	82.8	70.4	81.5	62.6	75.8	65.9	76.9	71.5	76.7	69.6	79.8
	1NN	Context2vec	71.8	82.6	69.1	80.5	61.3	74.5	65.6	78.0	71.9	76.6	69.0	79.7
	1NN	ELMo	71.6	82.8	69.6	80.9	62.2	74.7	66.2	77.7	71.3	77.0	69.0	79.6
	1NN	BERT-Base	75.5	84.9	71.5	81.4	65.1	78.9	69.8	82.1	73.4	78.1	72.2	82.0
	1NN	BERT-Large	76.3	84.8	**73.2**	**82.9**	66.2	80.0	71.7	83.1	74.1	**79.1**	73.5	82.8
	1NN	ALBERT-XXL	**76.6**	**85.6**	73.1	82.6	**67.3**	**80.1**	**71.8**	**83.5**	**74.3**	78.3	**73.7**	**83.0**
Hybrid		Seq2Seq Att+Lex+PoS	70.1	—	68.5	—	63.1*	—	66.5	—	69.2	—	68.6*	—
		Sense Compr. Ens.	79.7	—	77.8	—	73.4	—	78.7	—	82.6	—	**79.0**	—
		LMMS 1024	75.4	—	74.0	—	66.4	—	72.7	—	75.3	—	73.8	—
		LMMS 2048	76.3	84.5	75.6	85.1	68.1	81.3	75.1	86.4	77.0	80.8	75.4	84.4
		EWISE	73.8	—	71.1	—	67.3*	—	69.4	—	74.5	—	71.8*	—
		KnowBert† WN+WK	76.4	85.6	76.0	85.1	71.4	82.6	73.1	83.8	75.4	80.2	75.1	84.1
		GlossBERT	77.7	—	75.2	—	72.5*	—	76.1	—	80.4	—	77.0*	—
		BEM	79.4	—	77.4	—	74.5*	—	79.7	—	**81.7**	—	79.0*	—
		EWISER†	**80.8**	—	**79.0**	—	**75.2**	—	**80.7**	—	81.8*	—	80.1*	—
	—	*MFS Baseline*	65.6	77.4	66.0	77.8	54.5	70.6	63.8	74.8	67.1	75.3	64.8	76.2

3.4.3 Data Sets: Unified WSD Benchmark. Introduced by Raganato, Camacho-Collados, and Navigli (2017) as an attempt to construct a standard evaluation framework for WSD, the unified benchmark comprises five data sets from Senseval/SemEval workshops (see Section 2.3).[9] The framework provides 7,253 test instances for 4,363 sense types. In total, around 3,663 word types are covered with an average polysemy of 6.2 and across four parts of speech: nouns, verbs, adjectives, and adverbs.

Note that the data sets are originally designed for the fine-grained WSD setting. Nonetheless, in addition to the fine-grained setting, we provide results on the coarse-grained versions of the same test sets. To this end, we merged those senses that belonged to the same domain according to CSI (Coarse Sense Inventory) domain labels from Lacerra et al. (2020).[10] With this coarsening, we can provide more meaningful comparisons and draw interpretable conclusions. Finally, we followed the standard procedure and trained all models on SemCor (Miller et al. 1993).

3.4.4 Results. Table 1 shows the results[11] of all comparison systems on the unified WSD framework, both for fine-grained (FN) and coarse-grained (CS) versions. The $LMMS_{2048}$

9 Data set downloaded from `http://lcl.uniroma1.it/wsdeval/`.
10 CSI domains downloaded from `http://lcl.uniroma1.it/csi`.
11 SensEmBERT not included because it is only applicable to the noun portions of these test sets.

hybrid model, which is based on the 1NN BERT classifier, is the best-performer based solely on feature extraction. The latest fine-tuning hybrid solutions, particularly BEM and EWISER, show overall best performance, making the case for leveraging glosses and semantic relations to optimize pretrained weights for the WSD task. Generally, all BERT-based models achieve fine-grained results that are in the same ballpark as human average inter-annotator agreements for fine-grained WSD, which ranges from 64% and 80% in the three earlier data sets of this benchmark (Navigli 2009). In the more interpretable coarse-grained setting, LMMS achieves a score of 84.4%, similar to the other BERT-based models, which surpass 80%. The remaining supervised models perform roughly equal, marginally below 80% and clearly underperformed by BERT-based models.

3.4.5 Layer Performance. Current BERT-based 1NN WSD methods (see Section 3.3.1), such as LMMS and SensEmBERT, apply a **pooling** procedure to combine representations extracted from various layers of the model. The convention is to sum the embeddings from the last four layers, following the Named Entity Recognition experiments reported by Devlin et al. (2019). It is generally understood that lower layers are closer to their static representations (i.e., initialization) and, conversely, upper layers better match the modeling objectives (Tenney, Das, and Pavlick 2019). Still, Reif et al. (2019) have shown that this relation is not monotonic when it comes to sense representations from BERT. Additional probing studies have also pointed to irregular progression of context-specificity and token identity across the layers (Ethayarajh 2019; Voita, Sennrich, and Titov 2019), two important pre-requisites for sense representation.

Given our focus on measuring BERT's adeptness for WSD, and the known variability in layer specialization, we performed an analysis to reveal which layers produce representations that are most effective for WSD. This analysis involved obtaining sense representations learned from SemCor for each layer individually using the process described in Section 3.3.1.

Figure 1 shows the performance of each layer using a restricted version of the MASC corpus (Ide et al. 2008) as a validation set where only annotations for senses that occur in SemCor are considered. Any sentence that contained annotations for senses not occurring in SemCor was removed, restricting this validation set to 14,645 annotations out of 113,518. We restrict the MASC corpus so that our analysis is not affected by strategies for inferring senses (e.g., Network Propagation) or fallbacks (e.g., Most Frequent Sense). This restricted version of MASC is based on the release introduced in Vial, Lecouteux, and Schwab (2018), which mapped annotations to Princeton WordNet (3.0).

Similarly to Reif et al. (2019), we find that lower layers are not as effective for disambiguation as upper layers. However, our experiment specifically targets WSD and its results suggest a different distribution of the best performing layers than those reported by Reif et al. (2019). Nevertheless, this analysis shows that the current convention of using the sum of the last four layers for sense representations is sensible, even if not optimal.

Several model probing works have revealed that the scalar mixing method introduced by Peters et al. (2018a) allows for combining information from all layers with improved performance on lexico-semantic tasks (Liu et al. 2019a; Tenney et al. 2019; de Vries, van Cranenburgh, and Nissim 2020). However, scalar mixing essentially involves training a learned probe, which can limit attempts at analyzing the inherent semantic space represented by NLMs (Mickus et al. 2020).

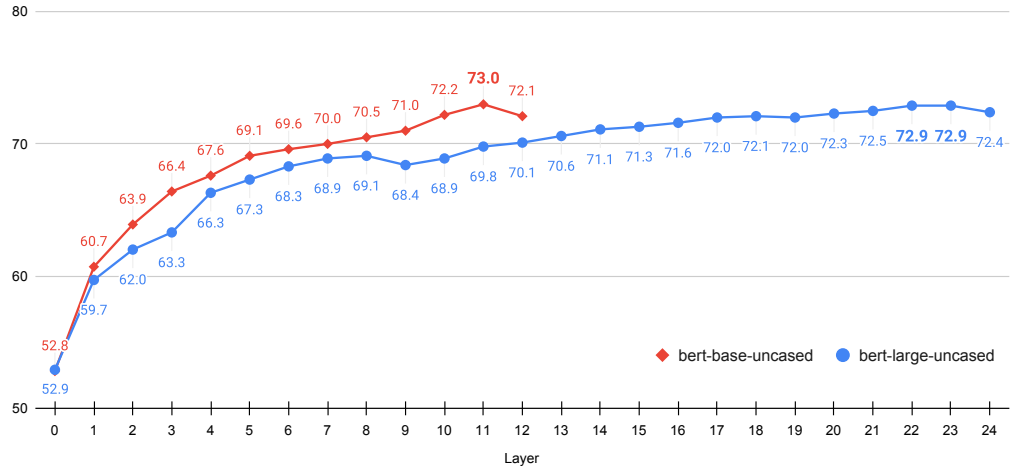

Figure 1
F-measure performance on a restricted version of the MASC corpus (Ide et al. 2008) for representations derived from individual layers of the two BERT models used in our experiments.

Table 2
F-Measure performance in the concatenation of all data sets of the unified WSD evaluation framework (Raganato, Camacho-Collados, and Navigli 2017), split by part of speech. As in Table 1 systems marked with † make use of external resources other than WordNet.

	Type	System	Nouns		Verbs		Adjectives		Adverbs	
			FN	CS	FN	CS	FN	CS	FN	CS
KB		UKB	71.2	80.5	50.7	69.2	75.0	82.7	77.7	91.3
		Lesk$_{ext}$+emb	69.8	79.0	51.2	69.2	51.7	62.4	80.6	92.8
		Babelfy†	68.6	78.9	49.9	67.6	73.2	82.1	79.8	91.6
Supervised	**1NN**	Context2vec	71.0	80.5	57.6	72.9	75.2	83.1	82.7	92.5
		ELMo	70.9	80.0	57.3	73.5	77.4	85.4	82.4	92.8
		BERT-Base	74.0	83.0	61.7	75.3	77.7	84.9	**85.8**	93.9
		BERT-Large	75.1	83.7	63.2	76.6	79.5	85.4	85.3	**94.2**
	SVM	IMS	70.4	79.4	56.1	72.5	75.6	84.1	82.9	93.1
		IMS+emb	71.9	80.5	56.9	73.1	75.9	83.8	84.7	93.4
Hybrid		LMMS$_{2048}$	**78.0**	**86.2**	64.0	76.5	**80.7**	**86.7**	83.5	92.8
		KnowBert† WN+WK	77.0	85.0	**66.4**	**78.8**	78.3	86.1	84.7	93.9
	–	*MFS Baseline*	67.6	77.0	49.6	67.2	73.1	82.0	80.5	92.9

3.4.6 Analysis by part of speech. Table 2 shows the results of BERT and the comparison systems by part of speech.[12] The results clearly show that verbs are substantially more difficult to model, which corroborates the findings of Raganato, Camacho-Collados, and Navigli (2017), while adverbs are the least problematic in terms of disambiguation. For example, in the fine-grained setting, BERT-Large achieves an overall F1 of 75.1% on nouns vs. 63.2% on verbs (85.3% on adverbs). The same trend is observed for other

12 For this table we only included systems for which we received access to their system outputs.

models, including hybrid ones. This may also be related to the electrophysiological evidence suggesting that humans process nouns and verbs differently (Federmeier et al. 2000). Another more concrete reason for this gap is due to the fine granularity of verb senses in WordNet. For instance, the verb *run* has 41 sense entries in WordNet, twelve of which denote some kind of motion.

The coarsening of sense inventory does help in bridging this gap, with the best models performing in the 75% ballpark. Nonetheless, the lower performance is again found in verb instances, with noun, adjective, and adverb performance being above 80% on the BERT-based models (above 90% in the case of adverbs). One problem with the existing coarsening methods is that they usually exploit domain-level information, whereas in some cases verbs do not belong to clear domains. For our example verb *run*, some of the twelve senses denoting motion are clustered into different domains, which eases the task for automatic models due to having fewer number of classes. However, one could argue that this clustering is artificial as all senses of the verb belong to the same domain.

Indeed, while the sense clustering provided by CSI (Lacerra et al. 2020) covers all PoS categories, it extends BabelDomains (Camacho-Collados and Navigli 2017), a domain clustering resource that covers mainly nouns. Although out of scope for this article, in the future it would be interesting to investigate verb-specific clustering methods (e.g., Peterson and Palmer 2018).

In the remainder of this article we focus on noun ambiguity, and check the extent to which language models can solve coarse-grained WSD in ideal settings. In Section 7, we extend the discussion about sense granularity in WSD.

4. CoarseWSD-20 Data Set

Standard WSD benchmarks mostly rely on WordNet. This makes the evaluations carried out on these data sets and the conclusions drawn from them specific to this resource only. Moreover, sense distinctions in WordNet are generally known to be too fine-grained (see more details about the fine granularity of WordNet in the discussion of Section 7) and annotations are scarce given the knowledge-acquisition bottleneck (Gale, Church, and Yarowsky 1992a; Pasini 2020). This prevents us from testing the limits of language models in WSD, which is one of the main motivations of this article.

To this end, we devise a new data set, CoarseWSD-20 henceforth, in an attempt to solve the aforementioned limitations. CoarseWSD-20 aims to provide a benchmark for the qualitative analysis of certain types of easily interpretable sense distinctions. Our data set also serves as a tool for testing the limits of WSD models in ideal training scenarios (i.e., with plenty of training data available per word).

In the following we describe the procedure we followed to construct CoarseWSD-20 (Section 4.1). Then, we present an estimation of the human performance (Section 4.2) and outline some relevant statistics (Section 4.3). Finally, we discuss the out-of-domain test set we built as a benchmark for experiments in Section 5.3.

4.1 Data Set Construction

CoarseWSD-20 targets noun ambiguity[13] for which, thanks to Wikipedia, data is more easily available. The data set focuses on the coarse-grained disambiguation setting,

13 There are arguably more types of ambiguity, including word categories (e.g., *play* as a noun or as a verb). Nevertheless, this type of ambiguity can be solved to a good extent by using state-of-the-art PoS taggers, which are able to achieve performances above 97% for English in general settings (Akbik, Blythe, and Vollgraf 2018).

which is more interpretable by humans (Lacerra et al. 2020). To this end, 20 words[14] and their corresponding senses were selected by a group of two expert computational linguists in order to provide a diverse data set. Wikipedia[15] was used as reference inventory and corpus. In this case, each Wikipedia page corresponds to an unambiguous sense. Sentences where a given Wikipedia page is referred to via a hyperlink are considered to be its corresponding sense-annotated sentences. The process to select 20 ambiguous words and their corresponding sense-annotated sentences was as follows:

1. A larger set of a few hundred ambiguous words that had a minimum of 30 occurrences[16] (i.e., sentences where one of their senses is referred to via a hyperlink) was selected.
2. Two experts selected 20 words based on a variety of criteria: type of ambiguity (e.g., spanning across domains or not), polysemy, overall frequency, distribution of instances across senses of the word, and interpretability. This process was performed semi-automatically, as initially the experts filtered words and senses manually providing a reduced set of words and associated senses. The main goal of this filtering was to discard those senses that were not easily interpretable or distinguishable by humans.

Once these 20 words were selected, we tokenized and lowercased the English Wikipedia and extracted all sentences that contained them and their selected senses as hyperlinks. All sentences were then semi-automatically verified so as to remove duplicate and noisy sentences. Finally, for each word we created a single data set based on a standard 60/40 train/test split.

4.2 Human Performance Estimation

As explained earlier this WSD data set was designed to be simple for humans to annotate. In other words, the senses considered for CoarseWSD-20 are easily interpretable. As a sanity check, we performed a disambiguation exercise with 1,000 instances randomly sampled from the test set (50 for each word). Four annotators[17] were asked to disambiguate a given target word in context using the CoarseWSD-20 sense inventory. Each annotator completed the task for five words. In the following section we provide details of the results of this annotation exercise, as well as general statistics of CoarseWSD-20.

4.3 Statistics

Table 3 shows the list of words, their associated senses, and the frequency of each word sense in CoarseWSD-20, along with the ratio of the first sense with respect to

14 The main justification to select 20 words (and no more) was the extent of experiments and the computation required to run a deep qualitative analysis (see Section 5.1). A larger number of words would have prevented us from running the analyses at the depth we envisaged: 20 provided a good trade-off between having a heterogeneous set of words and a deep qualitative analysis.

15 We used the Wikipedia dump of May 2016.

16 This threshold was selected for the goal of testing the language models under close-to-ideal conditions. A real setting should also include senses with even lower frequency, the so-called *long tail* (Ilievski, Vossen, and Schlobach 2018; Blevins and Zettlemoyer 2020), which would clearly harm automatic models.

17 All annotators were fluent English speakers and understood the predefined senses for their assigned words.

Table 3
Target words and their associated senses, represented by their Wikipedia page title, with their overall associated frequency in CoarseWSD-20 (train/test). *F2R* denotes the ratio of instances for first sense to the rest, while *Ent.* is the normalized entropy of sense distribution. Moreover, the *Human* performance is reported in terms of accuracy.

Word	F2R	Ent.	Hum	Senses	Frequency
apple	1.6	0.96	100	apple_inc	1,466/634
				apple	892/398
arm	2.8	0.83	100	arm_architecture	311/121
				arm	112/43
bank	23.1	0.28	98	bank	1,061/433
				bank_(geography)	46/22
bass	2.9	0.67	90	bass_guitar	2,356/1,005
				bass_(voice_type)	609/298
				double_bass	208/88
bow	1.0	0.87	98	bow_ship	266/117
				bow_and_arrow	185/72
				bow_(music)	72/26
chair	1.4	0.91	98	chairman	156/88
				chair	115/42
club	0.9	0.85	86	club	186/108
				nightclub	148/73
				club_(weapon)	54/21
crane	1.3	0.99	98	crane_(machine)	211/81
				crane_(bird)	161/76
deck	8.4	0.37	96	deck_(ship)	152/92
				deck_(building)	18/7
digit	2.2	0.74	100	numerical_digit	47/33
				digit_(anatomy)	21/9
hood	1.6	0.88	98	hood_(comics)	105/47
				hood_(vehicle)	42/13
				hood_(headgear)	24/22
java	1.4	0.96	100	java	2,641/1,180
				java_(progr._lang.)	1,863/749
mole	0.4	0.93	98	mole_(animal)	148/77
				mole_(espionage)	120/44
				mole_(unit)	108/42
				mole_sauce	53/23
				mole_(architecture)	51/20
pitcher	355.7	0.04	100	pitcher	6,403/2,806
				pitcher_(container)	18/13
pound	6.2	0.48	100	pound_mass	160/87
				pound_(currency)	26/10
seal	0.5	0.87	100	pinniped	305/131
				seal_(musician)	267/106
				seal_(emblem)	265/114
				seal_(mechanical)	38/12
spring	0.9	0.91	100	spring_(hidrology)	516/236
				spring_(season)	389/148
				spring_(device)	159/73
square	1.1	0.83	96	square	264/103
				square_(company)	167/62
				town_square	56/29
				square_number	21/13
trunk	1.3	0.85	100	trunk_(botany)	93/47
				trunk_(automobile)	36/16
				trunk_(anatomy)	35/14
yard	5.3	0.62	100	yard	121/61
				yard_(sailing)	23/11

Table 4
Statistics of the out of domain data set. The two rightmost columns show the number of instances for each of the seven words and their distribution across senses.

	Polysemy	Normalized entropy	No. of instances	Sense distribution
bank	2	0.87	48	34/14
chair	2	0.47	40	4/36
pitcher	2	0.52	17	15/2
pound	2	0.43	46	42/4
spring	3	0.63	31	3/24/4
square	3	0.49	26	22/2/2
club	2	0.39	13	12/1

the rest (F2E), normalized entropy[18] (Ent.), and an estimation of the human accuracy (see Section 4.2). The number of senses per word varies from 2 to 5 (11 words with two associated senses, 6 with three, 2 with four, and 1 with five) while the overall frequency ranges from 110 instances (68 for training) for *digit* to 9,240 (6,421 for training) for *pitcher*. As for the human performance, we can see how annotators did not have special difficulty in assigning the right sense for each word in context. Annotators achieve an accuracy of over 96% in all cases except for a couple of senses with slightly finer-grained distinctions such as *club* and *bass*.

Normalized entropy ranges from 0.04 to 0.99 (higher entropy shows more balanced sense distribution). While some words contain a roughly balanced distribution of senses (e.g., *crane* or *java*), other words' distribution are highly skewed (see normalized entropy values, e.g., for *pitcher* or *bank*).

Finally, in the Appendix we include more information for each of the senses available in CoarseWSD-20, including definitions and an example sentence from the data set.

4.4 Out of Domain Test Set

The CoarseWSD-20 data set was constructed exclusively based on Wikipedia. Therefore, the variety of language present in the data set might be limited. To verify the robustness of WSD models in a different setting, we constructed an out-of-domain test set from existing WordNet-based data sets.

To construct this test set, we leveraged BabelNet mappings from Wikipedia to WordNet (Navigli and Ponzetto 2012) to link the Wikipedia-based CoarseWSD-20 to WordNet senses. After a manual verification of all senses, we retrieved all sentences containing one of the target words in either SemCor (Miller et al. 1993) or any of the Senseval/SemEval evaluation data sets from Raganato, Camacho-Collados, and Navigli (2017). Finally, we only kept those target words for which all the associated senses were present in the WordNet-based sense annotated corpora and occurred at least 10 times. This resulted in a test set with seven target words (i.e., bank, chair, pitcher, pound, spring, square, and club). Table 4 shows the relevant statistics of this out-of-domain test set.

18 Computed as $\sum f_i \, log(f_i)$ normalized by $log(n)$ where n is the number of senses.

5. Evaluation

In this section we report on our quantitative evaluation in the coarse-grained WSD setting on CoarseWSD-20. We describe the experimental setting in Section 5.1 and then present the main results on CoarseWSD-20 (Section 5.2) and the out-of-domain test set (Section 5.3).

5.1 Experimental Setting

CoarseWSD-20 consists of 20 separate sets, each containing sentences for different senses of the corresponding target word. Therefore, the evaluation can be framed as a standard classification task for each word.

Given the classification nature of the CoarseWSD-20 data sets, we can perform experiments with our 1NN BERT system and compare it with a standard fine-tuned BERT model (see Section 3.3 for more details on the LM-based WSD approaches). Note that fine-tuning for individual target words results in many models (one per word). Therefore, this setup would not be computationally feasible in a general WSD setting, as the number of models would approach the vocabulary size. However, in our experiments we are interested in verifying the limits of BERT, without any other confounds or model-specific restrictions.

To ensure that our conclusions are generalizable, we also report 1NN and fine-tuning results using ALBERT. In spite of substantial operational differences, BERT and ALBERT have the most similar training objectives and tokenization methods out of several other prominent Transformer-based models (Yang et al. 2019; Liu et al. 2019b), thus being the most directly comparable. Given the similar performance between BERT-Large and ALBERT-XXLarge on the main CoarseWSD-20 data set, we proceed with further experiments using only BERT.

We also include two FastText linear classifiers (Joulin et al. 2017) as baselines: FTX-B (base model without pretrained embeddings) and FTX-C (using pretrained embeddings from Common Crawl). We chose FastText as the baseline given its efficiency and competitive results for sentence classification.

Configuration. Our experiments with BERT and ALBERT used the Transformers framework (v2.5.1) developed by Wolf et al. (2020), and we used the uncased pretrained base and large models released by Devlin et al. (2019) for BERT, and the xxlarge (v2) models released by Lan et al. (2020) for ALBERT. We use the uncased variants of Transformers models to match the casing in CoarseWSD-20 (except for ALBERT, which is only available in cased variants). Following previous feature extraction works (including our experiment in Section 3.4.1), with CoarseWSD-20 we also average sub-word representations and use the sum of the last four layers when extracting contextual embeddings. For fine-tuning experiments, we used a concatenation of the average embedding of target word's sub-words with the embedding of the [CLS] token, and fed them to a classifier. We used the same default hyper-parameter configuration for all the experiments. Given the fluctuation of results with fine-tuning, all the experiments are based on the average of three independent runs. Our experiments with FastText used the official package[19] (v0.9.1), with FastText-Base corresponding to the default supervised classification pipeline using randomly-initialized vectors, and FastText-Crawl corresponding to the

19 `https://fasttext.cc/`.

same pipeline but starting with pretrained 300-dimensional vectors based on Common Crawl. Following Joulin et al. (2017), classification with FastText is performed using multinominal logistic regression and averaged sub-word representations.

Evaluation Measures. In a classification setting, the performance of a model is measured by various metrics, among which precision, recall, and F-score are the most popular. Let TP_i (true-positive) and FP_i (false-positive) be the number of instances correctly / incorrectly classified as class c_i, respectively. Also, let TN_i (true-negative) and FN_i (false-negative) be the number of instances correctly / incorrectly classified as class c_j for any $j \neq i$. Therefore, for class c_i, precision P_i and recall R_i are defined as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (1) \qquad R_i = \frac{TP_i}{TP_i + FN_i} \quad (2)$$

In other words, precision is the fraction of relevant instances among the retrieved instances, and recall is the fraction of the total number of relevant instances that were actually retrieved. The F-score F_i for class c_i is then defined as the harmonic mean of its precision and recall values:

$$F_i = \frac{2}{P_i^{-1} + R_i^{-1}} = 2\frac{P_i.R_i}{P_i + R_i} \quad (3)$$

In order to have a single value to measure the overall performance of the model, we can take the weighted average of these computed values over all the classes, which is referred to as average micro, if the weights are set to be the number of instances for each class, and macro if the weights are set to be equal. For our experiments we mainly report *Macro-F1* and *Micro-F1*.

Number of Experiments. To provide an idea of the experiments run on (including the analysis in Section 6), in the following we detail the number of computations required. We evaluated six models, each of them trained and tested separately for each word (there are twenty of them). The same models are also trained with balanced data sets (Section 6.2.1). In total, 240 models trained and tested for the main results (excluding multiple runs). Then, the computationally more demanding models (BERT-Large) are also evaluated on the out-of-domain test set, and trained with different training data sizes (Section 6.2.2) and with fixed number of examples (Section 6.3). In the latter case, BERT-base and FastText models are also considered (sometimes with multiple runs). As a rough estimate, all the experiments took over 1,500 hours on a Tesla K80 GPU. These experiments do not include the experiments run in the standard benchmarks (Section 3.4) and all the extra analyses and prior experimental tests that did not make it into the article.

5.2 Results

Word-specific results for different configurations of BERT and ALBERT as well as the FastText baseline are shown in Table 5. In general, results are high for all Transformer-based models, over 90% in most cases. This reinforces the potential of language models for WSD, both in its light-weight 1NN and in the fine-tuning settings. Although BERT-Large slightly improves over BERT-Base, the performance of the former is very

Table 5
Micro-F1 (top) and macro-F1 (bottom) performance on the full CoarseWSD-20 data set for eight different models: FastText-Base (FTX-B) and -Crawl (FTX-C), 1NN and fine-tuned BERT-Base (BRT-B), -Large (BRT-L), and ALBERT-XXL (ALBRT). An estimation of the human performance (see Section 4.2 for more details) and the most frequent sense (MFS) baseline are also reported for each word. Rows in each table are sorted by the entropy of sense distribution (see Table 3), in descending order. Table cells are highlighted (from red to green) for better interpretability.

Word	Human	MFS	Static emb.		1NN			Fine-tune		
			FTX-B	FTX-C	BRT-B	BRT-L	ALBRT	BRT-B	BRT-L	ALBRT
Micro-F1 (Accuracy)										
crane	*98.0*	51.6	91.7	94.9	93.6	96.8	98.1	97.5	98.1	96.8
java	*100.0*	61.2	98.8	99.4	99.6	99.6	99.6	99.7	99.7	99.5
apple	*100*	61.4	96.5	98.4	99.0	99.2	99.4	99.6	99.6	99.3
mole	*98.0*	37.4	87.4	93.2	97.1	98.5	98.1	98.9	98.9	98.5
spring	*100*	51.6	91.9	94.5	97.4	97.8	99.3	98.0	98.3	98.2
chair	*98.0*	67.7	81.5	88.5	96.2	96.2	95.4	96.7	96.2	94.1
hood	*98.0*	57.3	80.5	89.0	98.8	100	98.8	98.0	99.6	98.8
seal	*100*	36.1	88.7	95.0	96.4	98.1	97.5	99.0	99.0	98.3
bow	*98.0*	54.4	89.8	95.8	96.3	95.3	96.7	97.5	98.5	97.7
club	*86.0*	53.5	79.2	80.7	81.2	85.1	82.7	85.2	84.7	84.3
trunk	*100*	61.0	84.4	90.9	96.1	98.7	98.7	97.8	98.3	99.1
square	*96.0*	49.8	87.0	90.3	95.2	96.1	94.2	95.8	95.7	96.5
arm	*100*	73.8	94.5	98.2	99.4	99.4	99.4	99.4	99.4	99.6
digit	*100*	78.6	92.9	100.0	100.0	100.0	100.0	99.2	100.0	100.0
bass	*90.0*	72.3	93.9	94.2	80.7	84.5	85.5	95.5	95.8	95.7
yard	*100*	84.7	86.1	94.4	76.4	88.9	93.1	98.6	99.5	99.5
pound	*100*	89.7	87.6	87.6	86.6	89.7	95.9	94.9	94.9	96.6
deck	*96.0*	92.9	91.9	93.9	89.9	91.9	94.9	96.6	95.3	97.0
bank	*98.0*	95.2	96.9	98.0	99.6	99.8	99.8	99.6	99.3	99.3
pitcher	*100*	99.5	99.6	99.7	99.9	99.9	100.0	100.0	100.0	99.8
AVG		66.5	90.0	93.8	94.0	95.8	96.4	97.4	97.5	97.4
Macro-F1										
crane	–	34.0	91.7	94.8	93.5	96.7	98.1	97.5	98.1	96.8
java	–	38.0	98.7	99.4	99.7	99.6	99.6	99.7	99.7	99.5
apple	–	38.1	96.2	98.1	99.0	99.1	99.3	99.6	99.6	99.3
mole	–	10.9	84.4	91.0	97.6	99.0	98.4	98.9	99.2	98.8
spring	–	22.7	91.1	94.9	97.4	97.8	99.2	97.8	98.1	98.2
chair	–	40.4	79.5	86.5	94.7	94.7	94.7	96.1	95.5	93.3
hood	–	24.3	70.5	83.2	98.5	100.0	98.5	97.8	99.6	98.3
seal	–	13.3	72.7	92.6	97.3	98.5	98.1	98.9	98.6	97.9
bow	–	23.5	83.3	93.7	97.0	95.7	97.3	97.5	98.6	96.8
club	–	23.2	73.2	80.5	84.6	88.7	87.1	84.3	84.1	84.0
trunk	–	25.3	76.0	85.9	97.9	99.3	99.3	97.6	98.0	99.0
square	–	16.6	67.7	76.3	92.5	94.7	89.7	92.2	91.4	93.5
arm	–	42.5	92.5	98.0	99.6	99.6	99.6	99.2	99.2	99.5
digit	–	44.0	83.3	100.0	100.0	100.0	100.0	98.8	100.0	100.0
bass	–	28.0	80.2	81.3	79.1	84.0	87.1	87.5	87.6	86.9
yard	–	45.9	54.5	81.8	86.1	93.4	95.9	97.2	99.1	99.1
pound	–	47.3	48.9	53.3	92.5	94.3	97.7	84.4	83.9	90.4
deck	–	48.2	56.1	57.1	88.0	95.7	84.1	83.4	78.0	85.2
bank	–	48.8	68.2	79.5	95.5	97.7	97.7	97.9	95.6	96.3
pitcher	–	49.9	61.5	69.2	99.9	100.0	100.0	97.3	97.3	89.2
AVG	–	33.2	76.5	84.9	94.5	96.4	96.1	95.2	95.1	95.1

similar to that of ALBERT-XXL across different configurations, despite having different architectures, number of parameters, and training objectives. Overall, performance variations in different models are similar to those for the human baseline. For instance, words such as *java* and *digit* seem easy for both humans and models to disambiguate, whereas words such as *bass* and *club* are challenging perhaps because of their more fine-grained distinctions.[20] As a perhaps surprising result, having more training instances does not necessarily lead to better performance, indicated by the very low Pearson correlation (0.2 or lower) of the number of training instances with results in all BERT configurations. Also, higher polysemy is not a strong indicator of lower performance (see Table 4.3 for statistics of the 20 words, including polysemy), as one would expect from a classification task with a higher number of classes (near zero average correlation across settings). In the following we also discuss other relevant points with respect to Most Frequent Sense (MFS) bias and fine-tuning.

MFS Bias. As expected, macro-F1 results degrade for the purely supervised classification models (FastText and fine-tuned BERT), indicating the inherent sense biases captured by the model that lead to lowered performance for the obscure senses (see the work by Postma et al. (2016) for a more thorough analysis on this issue). However, BERT proves to be much more robust with this respect whereas FastText suffers heavily (highlighted in the macro setting).

Impact of Fine-Tuning. On average, fine-tuning improves the performance for BERT-Large by 1.6 points in terms of micro-F1 (from 95.8% to 97.5%) but decreases on macro-F1 (from 96.4% to 95.1%). While BERT-Base significantly correlates with BERT-Large in the 1NN setting (Pearson correlation above 0.9 for both micro and macro), it has a relatively low correlation with the fine-tuned BERT-Base (0.60 on micro-F1 and 0.75 on macro-F1). The same trend is observed for BERT-Large, where the correlation between fine-tuning and 1NN is 0.71 and 0.63 on micro-F1 and macro-F1, respectively. The operating principles behind both approaches are significantly different, which may explain this relatively low correlation. While fine-tuning is optimizing a loss function during training, the 1NN approach is simply memorizing states. By optimizing losses, fine-tuning is more susceptible to overfit on the MFS. In contrast, by memorizing states, 1NN models sense independently and disregard sense distributions entirely. These differences can explain the main discrepancies between the two strategies, reflected for both micro and macro scores (macro-F1 penalizes models that are not as good for less frequent senses). The differences between 1NN and fine-tuned models will be analyzed in more detail in our analysis section (Section 6).

In our error analysis we will show, among other things, that there are some cases that are difficult even for humans to disambiguate, for example, the intended meaning of *apple* (fruit vs. company) or *club* (nightclub vs. association) in the following contexts taken from the test set: "it also likes apple" and "she was discovered in a club by the record producer peter harris."

20 Given that the human performance is estimated based on a small subset of the test set, and given the skewed distribution of sense frequencies, macro-F1 values can be highly sensitive to less-frequent senses (which might even have no instance in the subset); hence, we do not report macro-F1 for human performance.

5.3 Out of Domain

To verify the robustness of BERT and to see if the conclusions can be extended to other settings, we carried out a set of cross-domain evaluations in which the same BERT models (trained on CoarseWSD-20) were evaluated on the out-of-domain data set described in Section 4.4.

Table 6 shows the results. The performance trend is largely in line with that presented in Table 5, with some cases even having higher performance in this out-of-domain test set. Despite the relatively limited size of this test set, these results seem to corroborate previous findings and highlight the generalization capability of language models to perform WSD in different contexts. The fine-tuned version of BERT clearly achieves the highest micro-F1 scores, in line with previous experiments. Perhaps more surprisingly, BERT-Base 1NN achieves the best macro-F1 performance, also highlighting its competitiveness with respect to BERT-Large in this setting. As explained before, the 1NN strategy seems less prone to biases than the fine-tuned model, and this experiment shows the same conclusion extends to domain specificity as well, therefore the higher figures according to the macro metric. Interestingly, BERT-Base produces better results according to macro-F1 in the 1NN setting, despite lagging behind according to micro-F1. This suggests that data-intensive methods (e.g., fine-tuning) do not generally lead to significantly better results. Indeed, the results in Table 5 also confirm that the gains using a larger BERT model are not massive.

6. Analysis

In this section we perform an analysis on different aspects relevant to WSD on the CoarseWSD-20 data set. In particular, we first present a qualitative analysis on the type of contextualized embeddings learned by BERT (Section 6.1) and then analyze the impact of sense distribution of the training data (Section 6.2.1) as well as its size (Section 6.3) on WSD performance. Finally, we carry out an analysis on the inherent sense biases present in the pretrained BERT models (Section 6.4).

Table 6
Out-of-domain WSD results: Models trained on the CoarseWSD-20 training set and tested on the out-of-domain test set.

	Micro F1				**Macro F1**			
	1NN		**F-Tune**		**1NN**		**F-Tune**	
	BRT-B	BRT-L	BRT-B	BRT-L	BRT-B	BRT-L	BRT-B	BRT-L
bank	97.9	100.0	92.4	93.1	96.4	100.0	89.8	90.5
chair	100.0	100.0	98.3	99.2	100.0	100.0	94.8	97.4
pitcher	82.4	100.0	100.0	100.0	90.0	100.0	100.0	100.0
pound	89.1	87.0	96.4	94.9	94.0	81.5	85.5	77.5
spring	100.0	96.8	94.6	96.8	100.0	91.7	91.2	90.5
square	73.1	73.1	93.6	96.2	89.4	89.4	83.2	92.6
club	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
AVG	91.8	93.8	96.5	**97.2**	**95.7**	94.7	92.1	92.6

6.1 Contextualized Embeddings

The strong performance of the BERT-based 1NN WSD method reported for both fine and coarse-grained WSD proves that the representations produced by BERT are sufficiently precise to allow for effective disambiguation. Figures 2 and 3 illustrate the 2-D semantic space for contextualized representations of two target words (*square* and *spring*) in the test set. For each case, we applied the dimensionality technique that produced the most interpretable visualization, considering UMAP (McInnes et al. 2018) and Principal Component Analysis (PCA), although similar observations could be made using either of these two techniques. BERT is able to correctly distinguish and place most occurrences in distinct clusters. Few challenging exceptions exist, for example, two geometric senses of *square* are misclassified as public-square, highlighted in the figure ("... small *square* park located in ..." and " ... the narrator is a *square* ..."). Another interesting observation is for the season meaning of *spring*. BERT not only places all the contextualized representations for this sense in the same proximity in the space, it also makes a fine-grained distinction for the spring season of a specific year (e.g., "... in *spring* 2005 ...").

Beyond simply checking whether the nearest neighbor corresponds to the correct sense, there is still the question of the extent to which these representations are differentiated. In order to quantitatively analyze this, we plotted the distribution of cosine similarities between the contextual embeddings of the target word (to be disambiguated) from the test set and the closest predicted sense embedding learned from the training set. In Figure 4 we grouped these similarities by correct and incorrect predictions,

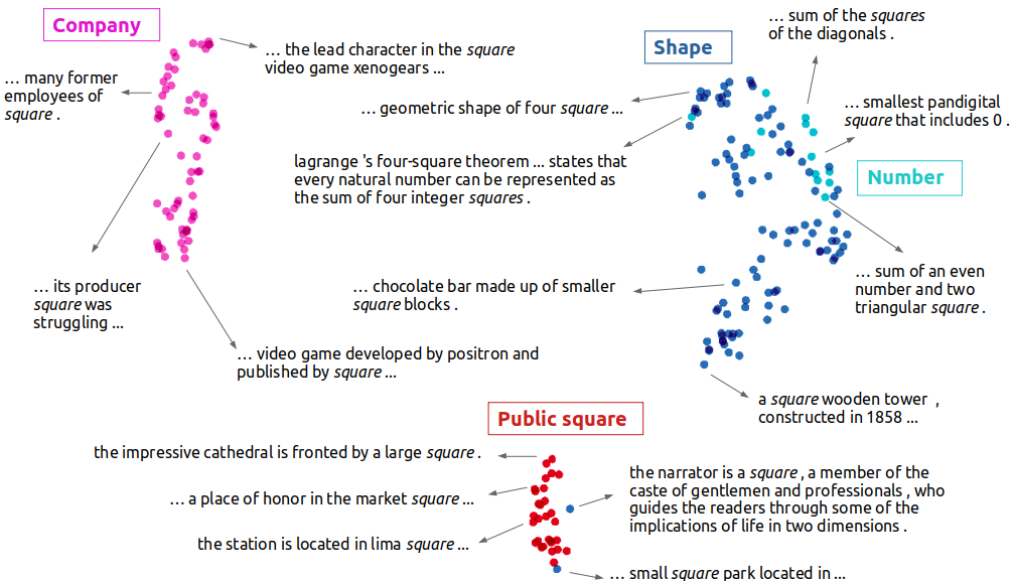

Figure 2
2-D visualizations of contextualized representations for different occurrences of *square* in the test set. While the company and public-square senses are grouped into distinct clusters, the numerical and geometrical meanings mostly overlap. Using UMAP for dimensionality reduction.

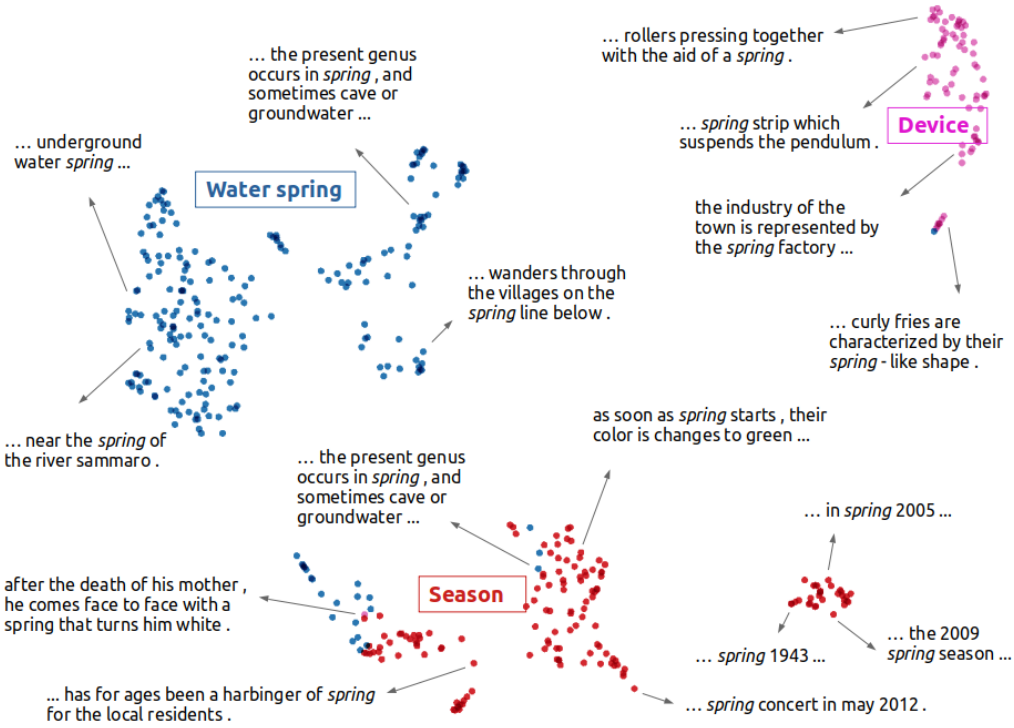

Figure 3
2-D visualizations of contextualized representations for different occurrences of *spring*. A fine-grained distinction can be observed for the season meaning of *spring*, with a distinct cluster (on the right) denoting the spring of a specific year. Using PCA for dimensionality reduction.

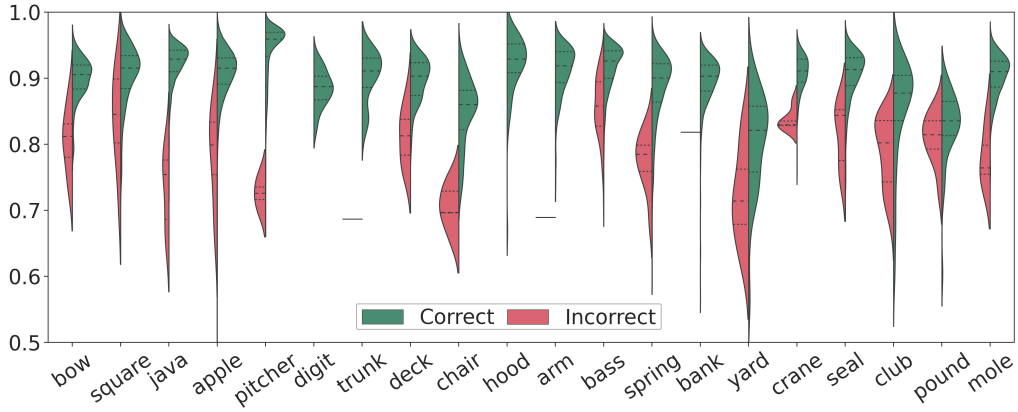

Figure 4
Distribution of cosine similarities between contextual embeddings (BERT-Large) of words to be disambiguated (in test set) and their corresponding closest sense embeddings learned from training data, for each word in the CoarseWSD-20 data set, grouped by correct and incorrect prediction.

revealing substantially different distributions. While incorrect prediction spans across the 0.5–0.9 interval, correct predictions are in the main higher than 0.75 for most words (over 97% of all predictions using BERT-Large with similarity higher than 0.75 are correct, for example). Consequently, this analysis also shows that a simple threshold could be used for effectively discarding false matches, increasing the precision of 1NN methods.

6.2 Role of Training Data

In order to gain insights on the role of training data, we perform two types of analysis: (1) distribution of training data—in particular, a comparison between skewed and balanced training sets (Section 6.2.1), and (2) the size of the training set (Section 6.2.2).

6.2.1 Distribution. To verify the impact of the distribution of the training data, we created a balanced training data set for each word by randomly removing instances for the more frequent senses in order to have a balanced distribution over all senses. Note that the original CoarseWSD-20 data set has a skewed sense distribution, given that it is constructed based on naturally occurring texts.

Table 7 shows the performance drop or increase when using a fully balanced training set instead of the original CoarseWSD-20 skewed training set (tested on the original skewed test set). Performance is generally similar across the two settings for the less entropic words (on top) that tend to have more uniform distributions. For the more entropic words (e.g., *deck*, *bank*, or *pitcher*), even though balancing the data

Table 7
Performance drop or increase when using a fully balanced training set instead of the original CoarseWSD-20 skewed training set.

	Micro F1						Macro F1					
	Static emb.		1NN		F-Tune		Static emb.		1NN		F-Tune	
	FTX-B	FTX-C	BRT-B	BRT-L	BRT-B	BRT-L	FTX-B	FTX-C	BRT-B	BRT-L	BRT-B	BRT-L
crane	−3.8	0.0	0.6	0.0	0.0	0.0	−3.7	0.0	0.6	0.0	0.0	0.0
java	−0.1	0.1	0.0	0.0	0.0	−0.1	−30.3	−15.1	0.1	0.0	0.0	−0.1
apple	−0.2	−0.6	0.0	0.0	0.0	−0.1	0.4	−0.4	0.0	0.0	0.0	−0.1
mole	−11.2	−1.5	0.0	0.0	−0.7	−0.7	−0.9	2.0	0.0	0.0	−0.5	−0.7
spring	−5.0	−2.0	0.0	0.2	−1.1	−0.9	−12.3	1.5	−0.2	0.1	−1.0	−0.7
chair	−6.2	−3.1	0.0	0.0	−1.0	0.3	−4.5	−2.3	0.0	0.0	−1.2	0.3
hood	−7.3	−1.2	0.0	−1.2	−0.4	0.0	12.2	4.4	−0.8	−1.5	−0.9	−0.3
seal	−23.1	−7.2	0.3	0.0	−2.9	−0.7	−9.0	−11.5	0.2	0.0	−7.3	−2.4
bow	−9.3	−3.7	0.0	0.0	−1.4	−0.8	−2.3	−2.0	0.0	0.0	−1.8	−1.5
club	−16.8	−5.9	0.0	−1.5	−0.8	−3.0	−8.6	−0.6	−0.3	−1.5	−0.4	−2.4
trunk	−13.0	−9.1	−3.9	0.0	−0.9	−1.7	−6.4	−4.3	−2.1	0.0	−0.9	−1.7
square	−23.7	−8.2	−6.8	−7.7	−4.7	−1.3	1.4	9.6	−3.4	−3.9	−4.8	1.1
bfarm	−2.4	−1.2	0.0	0.0	0.0	0.0	0.6	−0.8	0.0	0.0	0.0	0.0
digit	−16.7	−7.1	0.0	0.0	0.8	0.0	1.5	−4.5	0.0	0.0	1.2	0.0
bass	−9.1	−8.2	0.4	0.8	−5.1	−4.4	6.8	6.5	0.5	0.9	−5.6	−4.0
yard	−12.5	−5.6	−2.8	−4.2	−6.0	−2.3	18.2	11.6	−1.6	−2.5	−8.9	−3.9
pound	−34.0	−24.7	0.0	−1.0	−8.9	−1.4	18.5	36.7	7.5	−0.6	−8.8	2.0
deck	−26.3	−9.1	−2.0	−1.0	−5.7	−3.7	12.3	28.1	−1.1	−0.5	−5.0	2.1
bank	−17.4	−10.3	0.2	0.0	−2.6	−1.9	10.3	9.7	2.3	0.0	−10.6	−6.5
pitcher	−13.0	−6.4	−0.1	0.0	−1.3	−0.4	16.8	22.4	0.0	0.0	−26.7	−12.7
AVG	−12.6	−5.8	−0.7	−0.8	−2.1	−1.1	1.0	4.6	0.1	−0.5	−4.2	−1.6

inevitably reduces the overall number of training instances to a large extent, it can result in improved macro results for FastText, and even improved macro-recall results for fine-tuning, as we will see in Table 8.

This can be attributed to the better encoding of the least frequent senses, which corroborates the findings of Postma, Izquierdo Bevia, and Vossen (2016) for conventional supervised WSD models, such as IMS or, in this case, FastText. In contrast, the micro-averaged results clearly depend on accurately knowing the original distribution in both the supervised and fine-tuning settings, as was also discussed in previous works (Bennett et al. 2016; Pasini and Navigli 2018). Moreover, the feature extraction procedure (1NN in this case) is much more robust to training distribution changes. Indeed, being solely based on vector similarities, the 1NN strategy is not directly influenced by the number of occurrences of each sense in the CoarseWSD-20 training set.

To complement these results, Table 8 shows the performance difference on the MFS (Most Frequent Class) and LFS (Least Frequent Class) classes when using the balanced training set. The most interesting takeaway from this experiment is the marked difference between precision and recall for the LFS in entropic words (bottom). While the recall of the BERT-Large fine-tuned model increases significantly (up to 52.4 points in the case of *deck*), the precision decreases (e.g., −27.1 points for *deck*). This means that the model is clearly less biased toward the MFS with a balanced training set, as we could expect. However, the precision for LFS is also lower, due to the model's lower sensitivity for higher-frequency senses. In general, these results suggest that the fine-tuned

Table 8
Precision and recall drop or increase on the Most Frequent Sense (MFS) and Least Frequent Sense (LFS) classes when using a fully balanced training set.

	F-Tune (BRT-L)				1NN (BRT-L)			
	Precision		Recall		Precision		Recall	
	MFS	LFS	MFS	LFS	MFS	LFS	MFS	LFS
crane	0.4	−0.4	−0.4	0.4	0.0	0.0	0.0	0.0
java	0.0	−0.3	−0.2	0.0	0.0	0.0	0.0	0.0
apple	−0.1	−0.1	−0.1	−0.2	0.0	0.0	0.0	0.0
mole	−0.9	−0.8	−0.9	−1.5	0.0	0.0	0.0	0.0
spring	−0.6	−1.0	−1.3	−1.4	0.0	0.6	0.0	0.0
chair	0.0	0.9	0.4	0.0	0.0	0.0	0.0	0.0
hood	0.7	0.0	0.0	−2.6	−2.1	0.0	0.0	0.0
seal	−0.3	0.0	−0.5	0.0	0.0	0.0	0.0	0.0
bow	0.8	−1.0	−0.6	−1.4	0.0	0.0	0.0	0.0
club	−3.4	−1.6	−2.2	−6.9	−3.9	0.0	0.9	−5.5
trunk	0.7	−7.4	−3.6	0.0	0.0	0.0	0.0	0.0
square	6.5	−0.5	−9.4	0.0	−0.4	0.0	−15.5	0.0
arm	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
digit	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bass	2.5	−0.4	−8.6	−0.8	−0.7	1.8	0.7	1.1
yard	0.5	−14.0	−3.3	3.0	0.0	−7.9	−4.9	0.0
pound	4.0	−23.4	−5.8	36.7	−2.4	0.0	0.0	−1.1
deck	3.9	−27.1	−8.0	52.4	−2.9	0.0	0.0	−1.1
bank	0.8	−32.5	−2.8	15.2	0.0	0.0	0.0	0.0
pitcher	0.1	−46.0	−0.4	10.3	0.0	0.0	0.0	0.0
AVG	0.8	−7.8	−2.4	5.2	−0.6	−0.3	−0.9	−0.3

Table 9
Macro- and micro-F1 % performance for the two BERT-Large models. The last two rows indicate the F1 performance on the Most Frequent Sense (MFS) and Least Frequent Sense (LFS) classes.

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL%**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL%**
Macro	74.2	81.6	85.8	91.5	94.2	95.1	94.4	95.3	95.6	95.8	96.0	96.4
Micro	89.0	93.5	95.3	96.3	97.0	97.5	95.5	95.8	95.7	95.7	95.6	95.8
MFS	91.9	95.3	96.4	97.2	97.5	98.0	95.8	95.8	95.6	95.6	95.4	95.4
LFS	52.1	64.3	71.9	83.4	88.5	91.0	91.6	93.3	94.1	94.6	95.5	96.6

BERT model is overly sensitive to the distribution of the training data, while its feature extraction counterpart suffers considerably less from this issue. In Section 6.4 we will extend the analysis on the bias present in each of the models.

6.2.2 Size. We performed an additional experiment to investigate the impact of training data size on the performance for the most and least frequent senses. To this end, we shrank the training data set for all words, while preserving their original distribution. Table 9 shows a summary of the aggregated micro-F1 and macro-F1 results, including the performance on the most and least frequent senses.[21] Clearly, the 1NN model performs considerably better than fine-tuning in settings with low training data (e.g., 74.2% to 94.4% macro-F1 with 1% of the training data). Interestingly, the 1NN's performance does not deteriorate with few training data, as the results with 1% and 100% of the training data do not vary much (less than two absolute points decrease in performance for micro-F1 and 0.3 in terms of micro-F1). Even for the LFS, the overall performance with 1% of the training data is above 90 (i.e., 91.6). This is an encouraging behavior, as in real settings sense-annotated data is generally scarce.

To obtain a more detailed picture for each word, Table 10 shows the macro-F1 results for each word and training size.[22] Again, we can observe a large drop for the most entropic words in the fine-tuning setting. Examples of words with a considerable degrading performance are *pitcher* or *bank*, which decrease from macro-F1 scores higher than 95% in both cases (97.3 and 95.6, respectively) to as low as 49.9 and 50.2 (almost random chance) with 1% of the training data, and still lower than 75% with 10% of the training data (63.9 and 74.9, respectively). This trend clearly highlights the need for gathering reasonable amounts of training data for the obscure senses. Moreover, this establishes a trade-off between balancing or preserving the original skewed distribution depending on the end goal, as discussed in Section 6.2.1.

6.3 *n*-Shot Learning

Given the results of the previous section, one may wonder how many instances would be enough for BERT to perform well in coarse-grained WSD. To verify this, we fine-tuned BERT on limited amounts of training data, with uniform distribution over word senses, each having between 1 (i.e., one-shot) and 30 instances. Figure 5 shows the

21 In the Appendix we include detailed results for each word and their MFS and LFS performance.
22 In the Appendix we include the same table for the micro-F1 results.

Table 10
Macro-F1 results on the CoarseWSD-20 test set using training sets of different sizes sampled from the original training set.

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL**
crane	83.3	95.7	95.7	96.8	95.5	98.1	96.4	96.6	96.7	96.7	96.7	96.7
java	99.0	99.1	99.6	99.5	99.6	99.7	99.6	99.6	99.6	99.6	99.6	99.6
apple	99.3	99.4	99.4	99.4	99.5	99.6	99.1	99.1	99.1	99.1	99.1	99.1
mole	79.8	94.8	97.6	99.3	99.3	99.2	98.6	99.1	99.0	99.0	99.0	99.0
spring	94.8	97.6	96.8	96.9	97.8	98.1	97.9	97.9	97.9	97.9	97.9	97.8
chair	76.2	92.2	95.2	96.1	96.4	95.5	94.3	94.6	94.7	94.7	94.7	94.7
hood	57.2	89.3	92.3	96.6	97.7	99.6	94.7	98.6	99.2	99.5	100.0	100.0
seal	80.3	95.8	96.5	98.2	98.0	98.6	98.6	98.6	98.7	98.6	98.6	98.5
bow	49.3	86.8	95.7	96.0	97.5	98.6	93.5	96.0	96.2	95.9	95.7	95.7
club	70.1	77.4	77.0	80.0	83.0	84.1	85.6	86.5	87.4	87.6	88.0	88.7
trunk	77.9	84.6	97.5	98.6	98.6	98.0	97.7	98.3	98.7	99.3	99.3	99.3
square	68.4	69.6	73.5	76.6	79.4	91.4	86.7	88.0	87.8	88.1	91.1	94.7
arm	90.1	98.1	99.2	99.2	99.2	99.2	99.6	99.6	99.6	99.6	99.6	99.6
digit	92.4	79.7	92.1	98.8	100.0	100.0	99.1	100.0	100.0	100.0	100.0	100.0
bass	72.2	79.4	84.3	86.7	87.8	87.6	83.1	83.8	84.4	84.8	84.8	84.0
yard	82.7	85.7	88.3	94.3	99.1	99.1	93.4	93.4	92.8	92.6	92.2	93.4
pound	53.5	50.4	47.3	52.6	83.2	83.9	87.0	92.4	93.3	93.2	94.3	94.3
deck	56.7	48.2	48.2	70.2	77.2	78.0	85.5	85.1	88.9	91.1	92.1	95.7
bank	50.2	55.9	74.9	97.1	95.7	95.6	97.0	98.6	98.9	98.5	97.7	97.7
pitcher	49.9	52.3	63.9	96.5	99.3	97.3	100.0	100.0	100.0	100.0	100.0	100.0
Average	74.2	81.6	85.8	91.5	94.2	95.1	94.4	95.3	95.6	95.8	96.0	96.4

performance of both 1NN and fine-tuning strategies on this set of experiments. Perhaps surprisingly, we can see how having only three instances per sense is enough for achieving a competitive result. Then, only small improvements can be obtained by adding more instances. This is relevant in the context of WSD, as generally current sense-annotated corpora follow Zipf's law (Zipf 1949), and therefore contain many repeated senses that are very frequent. Significant improvements may therefore be obtained by simply getting a few sense annotations for less frequent instances. Figure 6 summarizes Figure 5 by showing the distribution of words according to their performance in the two strategies. In the case of fine-tuning, the performance is generally better in terms of micro compared with macro F-score. This further corroborates the previous observation, that there is a bias toward the most frequent sense (cf. Section 6.2.1). Additionally, in contrast to 1NN, fine-tuning greatly benefits from the increase in the training-data size, which also indicates the more robust behavior of 1NN strategy compared to its counterpart (cf. Section 6.2.1).

6.4 Bias Analysis

Supervised classifiers are known to have label bias toward more frequent classes, that is, those that are seen more frequently in the training data (Hardt et al. 2016), and this is particularly noticeable in WSD (Postma, Izquierdo Bevia, and Vossen 2016; Blevins and Zettlemoyer 2020). Label bias is a reasonable choice for maximizing performance when the distribution of classes is skewed, particularly for classification tasks with a small number of categories (which is often the case in WSD). For the same reason, many

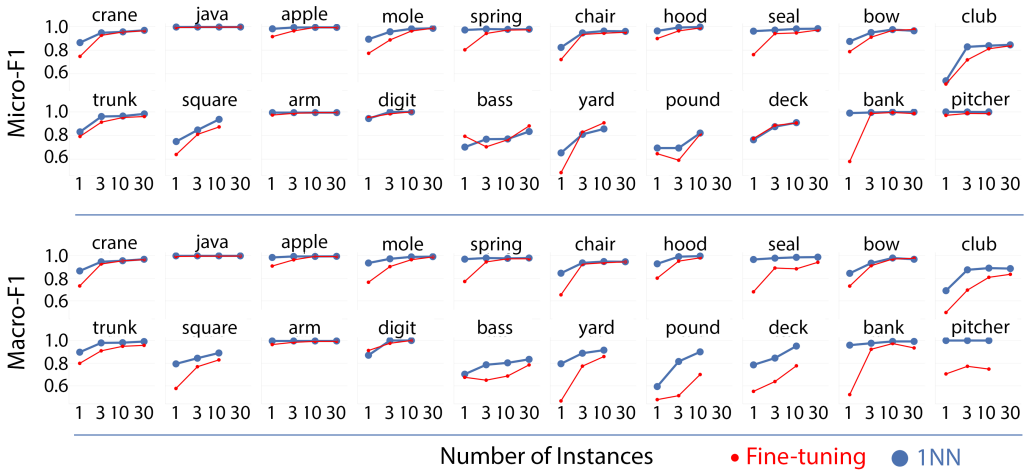

Figure 5
Micro and macro F-scores for different values of n in the n-shot setting, for all the words and for the two WSD strategies. Results are averaged from three runs over three different samples.

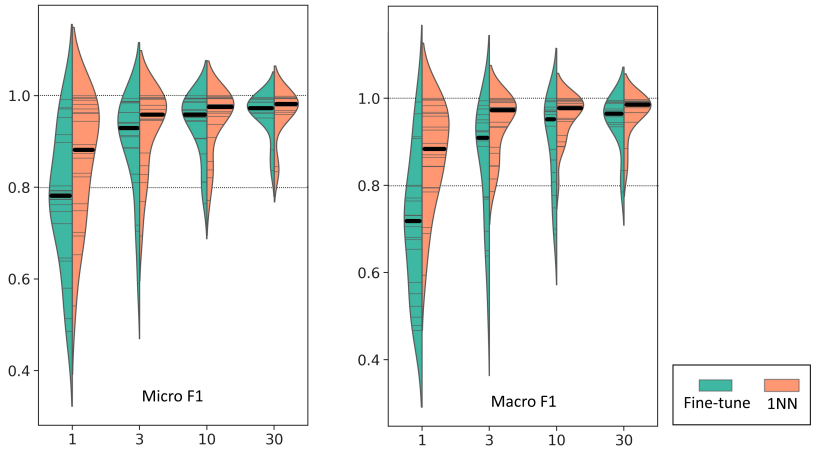

Figure 6
Distribution of performance scores for all 20 words according to micro and macro F1 in the two WSD strategies (left: fine-tuning, right: 1NN) and for different values of n—i.e., 1, 3, 10, 30 (if available).

of the knowledge-based systems are coupled with the MFS back-off strategy: When the system is not confident in its disambiguation, it backs off to the most frequent sense (MFS) of the word (instead of resorting to the low-confidence decision).

We were interested in investigating the inherent sense biases in the two BERT-based WSD strategies. We opted for the n-shot setting given that it provides a suitable setting for evaluating the relationship between sense bias and training data size. Moreover, given that the training data in the n-shot setting is uniformly distributed (balanced), the impact of sense-annotated training data in introducing sense bias is minimized. This analysis is mainly focused on two questions: (1) how do the two strategies (fine-tuning

Table 11
Average sense bias values (B) for the two WSD strategies and for different values of n.

One-shot		3-shot		10-shot		30-shot	
F-Tune	1NN	F-Tune	1NN	F-Tune	1NN	F-Tune	1NN
0.232	0.137	0.111	0.078	0.050	0.052	0.021	0.025

and 1NN) compare in terms of sense bias?, and (2) what are the inherent sense biases (if any) in the pretrained BERT language model?

6.4.1 Sense Bias Definition. We propose the following procedure for computing the disambiguation bias toward a specific sense.[23] For a word with polysemy n, we are interested in computing the disambiguation bias B_j toward its j^{th} sense (s_j). Let n_{ij} be the total number of test instances with the gold label s_i that were mistakenly disambiguated as s_j ($i \neq j$). We first normalize n_{ij} by the total number of (gold-labeled) instances for s_i, that is, $\Sigma_j n_{ij}$, to obtain bias b_{ij}, which is the bias from sense i to sense j. In other words, b_{ij} denotes the ratio of s_i-labeled instances that were misclassified as s_j. The total bias toward a specific sense, B_j, is then computed as:

$$B_j = \sum_{\substack{i=1 \\ i \neq j}}^{n} \left(\frac{n_{ij}}{\Sigma_j n_{ij}}\right) \tag{4}$$

The value of B_j denotes the tendency of the disambiguation system to disambiguate a word with the intended sense of s_k, $k \neq j$, incorrectly as s_j. The higher the value of B_j, the more the disambiguation model is biased towards s_j. We finally compute the **sense bias** B as the *maximum* B_j value toward different senses of a specific word, that is, $max(B_j), j \in [1, n]$. Given fluctuations in the results, particularly for the case of small training data, we take the median of three runs to compute B_j.

In our coarse-grained disambiguation setting, the bias B can be mostly attributed to the case where the system did not have enough evidence to distinguish s_j from other senses and had pretraining bias towards s_j. One intuitive explanation for this would be that the language model is biased toward s_j because it has seen the target word more often with this intended sense than other $s_{k,j \neq k}$ senses.

6.4.2 Results. Table 11 reports the average sense bias values (B) for the two WSD strategies and for different values of n (training data size) in the n-shot setting. We also illustrate using radar charts in Figure 7 the sense bias for a few representative cases. The numbers reported in the figure (in parentheses) represent the bias value B for the corresponding setting (word, WSD strategy, and n's value).

Based on our observations, we draw the following general conclusions.

Bias and Training Size. There is a consistent pattern across all words and for both the strategies: Sense bias rapidly reduces with increase in the training data. Specifically, the

23 The procedure can presumably be used for quantifying bias in other similar classification settings.

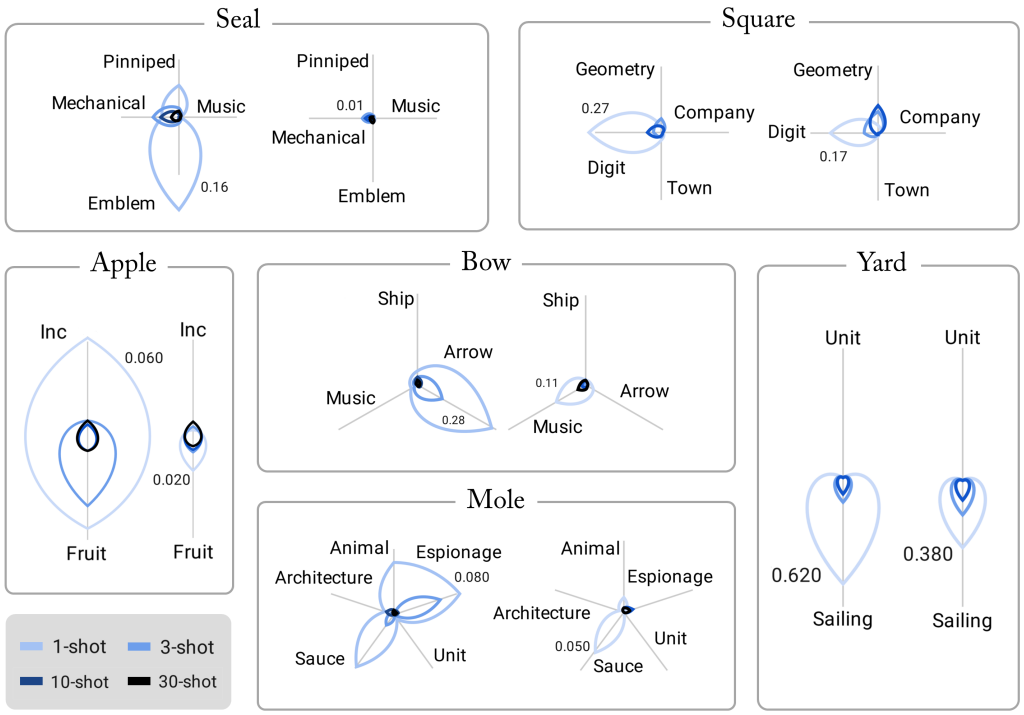

Figure 7
Sense bias for a few representative cases from each polysemy class for the two WSD strategies (left: fine-tuning, right: 1NN) and for different values of n, i.e., 1, 3, 10, 30 (if available).

average bias B approximately reduces by half with each step of increase in the training size. This is supported by the radar charts in Figure 7 (see, for instance, *apple*, *yard*, and *bow*). The WSD system tends to be heavily biased in the one-shot setting (particularly in the fine-tuning setting), but the bias often improves significantly with just 3 instances in the training data (3-shot).

Disambiguation Strategy: 1NN vs. Fine-Tuning. Among the two WSD strategies, the 1NN approach proves to be more robust with respect to sense biases. This is particularly highlighted in the one-shot setting where the average sense bias value is 0.137 for 1NN in comparison to 0.232 for fine-tuning. The trend is also clearly visible for almost all words in the radar charts in Figure 7. This corroborates our findings in Section 6.3 that the 1NN strategy is the preferable choice particularly with limited data. For higher values of n (larger training sizes) the difference between the two strategies diminishes, with both settings proving robust with respect to sense bias.

It is also notable that the two strategies, despite being usually similar in behavior, might not necessarily have matching biases toward the same senses. For instance, the fine-tuning setting shows bias only toward the arrow sense of *bow*, whereas 1NN is instead (slightly) biased toward its music sense. Another example is for the word *digit* for which with the same set of training instances in the one-shot setting (one sentence for each of the two senses), all the mistakes (5 in total) of the fine-tuning model are

numerical digits incorrectly tagged as anatomical, whereas all the mistakes in the 1NN setting (5 in total) are the reverse.

Finally, we also observed that for cases with subtle disambiguation, both the strategies failed consistently in the one-shot setting. For instance, a common mistake shared by the two strategies was for cases where the context contained semantic cues for multiple senses, for example, "the English word *digit* as well as its translation in many languages is also the anatomical term for fingers and toes." in which the intended meaning of *digit* is the numerical one (both strategies failed on disambiguation for this). This observation is in line with the analysis of Reif et al. (2019), which highlighted the failure of BERT in identifying semantic boundaries of words.

Pretraining Label Bias. In most of the conventional supervised WSD classifiers (such as IMS), which rely on sense-annotated training data as their main source of information, the source of sense bias is usually the skewed distribution of instances for different senses of a word (Pilehvar and Navigli 2014). For instance, the word *digit* would appear much more frequently with its numerical meaning than the finger meaning in an open-domain text. Therefore, a sense-annotated corpus that is sampled from open-domain texts shows a similar sense distribution, resulting in a bias toward more frequent senses in the classification.

Given that in the *n*-shot setting we restrict the training data sets to have a uniform distribution of instances, sense bias in this scenario can be indicative of inherent sense biases in BERT's pretraining. We observed that the pretrained BERT indeed exhibits sense biases, often consistently across the two WSD strategies. For instance, we observed the following biases toward (often) more frequent senses of words: *java* toward its programming sense (rather than island), *deck* toward ship deck (rather than building deck), *yard* toward its sailing meaning (rather than measure unit), and *digit* and *square* toward their numerical meanings. We also observed some contextual cues that misled the WSD system, especially in the one-shot setting. For instance, we observed that our BERT-based WSD system had a tendency to classify *square* as its digit meaning whenever there was a *number* in its context, for example, "marafor is a roman square with two temples attached" or "it has 4 trapezoid and 2 square faces." Not surprisingly, the source of most bias toward the digit sense of *square* is from its geometrical sense (which has domain relatedness). Also, classification for *digit* was often biased toward its numerical meaning. Similarly to the case of *square*, the existence of a number in context seems to bias the model toward numerical meanings, for example, "There were five *digit* on each hand and four on each foot."

Sensitivity to Initialization. We observed a high variation in the results, especially for the one-shot setting, suggesting the high sensitivity of the model with little evidence from training to the initialization point. For instance, in the one-shot experiment for the fine-tuning model and the word *bank*, in three runs, 1%, 60%, and 70% of the test instances for the financial bank are incorrectly classified as river bank. Similarly, for *crane*, 12%, 25%, and 72% of the machine instances are misclassified as bird in three runs. The 1NN strategy, in addition to being less prone to sense biases, is generally more robust across multiple runs. For these two examples, the figures are 2%, 0%, and 0% for *bank* and 15%, 0%, and 27% for *crane*. Other than the extent of bias, we observed that the direction can also change dramatically from run to run. For example, in the one-shot 1NN setting and for the word *apple*, almost all the mistakes in the first two runs (37 of 38 and 12 of 14) were incorporation for fruit, whereas in the third run, almost all (6 of 7) were fruit for incorporation.

7. Discussion

In the previous sections we have run an extensive set of experiments to investigate various properties of language models when adapted to the task of WSD. In the following we discuss some of the general conclusions and open questions arising from our analysis.

Fine-Grained vs. Coarse-Grained. A well-known issue of WordNet is the fine granularity of its sense distinctions (Navigli 2009). For example, the noun *star* has 8 senses in WordNet, two of which refer to a "celestial body," only differing in if they are visible from the Earth or not. Both meanings translate to *estrella* in Spanish and therefore this sense distinction serves no advantage in MT, for example. In fact, it has been shown that coarse-grained distinctions are generally more suited to downstream applications (Rüd et al. 2011; Severyn, Nicosia, and Moschitti 2013; Flekova and Gurevych 2016; Pilehvar et al. 2017). However, the coarsening of sense inventories is certainly not a solved task. Whereas in this article we relied either on experts for selecting senses from Wikipedia (given the reduced number of selected words) or domain labels from lexical resources for WordNet (Lacerra et al. 2020), there are other strategies for coarsening sense inventories (McCarthy, Apidianaki, and Erk 2016; Hauer and Kondrak 2020)—for instance, based on translations or parallel corpora (Resnik and Yarowsky 1999; Apidianaki 2008; Bansal, DeNero, and Lin 2012). This is generally an open problem, especially for verbs (Peterson and Palmer 2018), which have not been analyzed in-depth in this article due to lack of effective techniques for an interpretable coarsening. Indeed, while in this work we have shown how contextualized embeddings encode meaning to a similar extent as humans do, for fine-grained distinctions these have been shown to correlate to a much lesser extent, an area that requires further exploration (Haber and Poesio 2020).

Fine-Tuning vs. Feature Extraction (1NN). The distinction between fine-tuning and feature extraction has been already studied in the literature for different tasks (Peters, Ruder, and Smith 2019). The general assumption is that fine-tuned models perform better when reasonable amounts of training data are available. In the case of WSD, however, feature extraction (specifically the 1NN strategy explained in this article) is the more solid choice on general grounds, even when training data is available. The advantages of feature extraction (1NN) with respect to fine-tuning are 3-fold:

1. It is significantly less expensive to train as it simply relies on extracting contextualized embeddings from the training data. This is especially relevant when the WSD model is to be used in an all-words setting.
2. It is more robust to changes in the training distribution (see Section 6.2.1).
3. It works reasonably well for limited amounts of training data (see Section 6.2.2), even in few-shot settings (see Section 6.3).

Few-Shot Learning. An important limitation of supervised WSD models is their dependence on sense-annotated corpora, which is expensive to construct, that is, the so-called knowledge-acquisition bottleneck (Gale, Church, and Yarowsky 1992b; Pasini 2020). Therefore, being able to learn from a limited set of examples is a desirable property of WSD models. Encouragingly, as mentioned above, the simple 1NN method studied in this article shows robust results even with as few as three training examples per word sense. In the future it would be interesting to investigate models relying on knowledge

from lexical resources that can perform WSD with no training instances available (i.e., zero-shot), in the line of Kumar et al. (2019) and Blevins and Zettlemoyer (2020).

8. Conclusions

In this article we have provided an extensive analysis on how pretrained language models (particularly BERT) capture lexical ambiguity. Our aim was to inspect the capability of BERT in predicting different usages of the same word depending on its context, similarly as humans do (Rodd 2020). The general conclusion we draw is that in the ideal setting of having access to enough amounts of training data and computing power, BERT can approach human-level performance for coarse-grained noun WSD, even in cross-domain scenarios. However, this ideal setting rarely occurs in practice, and challenges remain to make these models more efficient and less reliant on sense-annotated data. As an encouraging finding, feature extraction-based models (referred to as 1NN throughout the article) show strong performance even with a handful of examples per word sense. As future work it would be interesting to focus on the internal representation of the Transformer architecture by, for example, carrying out an in-depth study of layer distribution (Tenney, Das, and Pavlick 2019), investigating the importance of each attention head (Clark et al. 2019), or analyzing the differences for modeling concepts, entities, and other categories of words (e.g., verbs). Moreover, our analysis could be extended to additional Transformer-based models, such as RoBERTa (Liu et al. 2019b) and T5 (Raffel et al. 2020).

To enable further analysis of this type, another contribution of the article is the release of the CoarseWSD-20 data set (Section 4), which also includes the out-of-domain test set (Section 4.4). This data set can be reliably used for quantitative and qualitative analyses in coarse-grained WSD, as we performed. We hope that future research in WSD will take inspiration on the types of analyses performed in this work, as they help shed light on the advantages and limitations of each approach. In particular, few-shot and bias analysis along with training distribution variations are key aspects to understanding the versatility and robustness of any given approach.

Finally, WSD is clearly not a solved problem, even in the coarse-grained setting, due to a few challenges: (1) it is an arduous process to manually create high-quality full-coverage training data; therefore, future research should also focus on reliable ways of automating this process (Taghipour and Ng 2015; Delli Bovi et al. 2017; Scarlini, Pasini, and Navigli 2019; Pasini and Navigli 2020; Loureiro and Camacho-Collados 2020; Scarlini, Pasini, and Navigli 2020b) and/or leveraging specific knowledge from lexical resources (Luo et al. 2018; Kumar et al. 2019; Huang et al. 2019); and (2) the existing sense-coarsening approaches are mainly targeted at nouns, and verb sense modeling remains an important open research challenge.

APPENDIX

Word-in-Context Evaluation

Word-in-Context (Pilehvar and Camacho-Collados 2019, WiC) is a binary classification task from the SuperGLUE language understanding benchmark (Wang et al. 2019) aimed at testing the ability of models to distinguish between different senses of the same word without relying on a predefined sense inventory. In particular, given a target word (either a verb or a noun) and two contexts where such target word occurs, the

Table 12
Sample positive (T) and negative (F) pairs from the WiC data set (target word in *italics*).

F	There's a lot of trash on the *bed* of the river | I keep a glass of water next to my *bed* when I sleep
F	*Justify* the margins | The end *justifies* the means
T	*Air* pollution | Open a window and let in some *air*
T	The expanded *window* will give us time to catch the thieves | You have a two-hour *window* of clear weather to finish working on the lawn

Table 13
Accuracy (%) performance of different models on the WiC data set.

Type	Model	Accuracy
Hybrid	KnowBERT (Peters et al. 2019)	70.9
	SenseBERT (Levine et al. 2020)	72.1
	LMMS-LR (Loureiro and Jorge 2019b)	68.1
Fine-tuned/Supervised	BERT-Base	69.6
	BERT-Large	69.6
	FastText-B	52.3
	FastText-C	54.7
Lowerbound	*Most Frequent Class*	50.0
Upperbound	*Human performance*	80.0

task consists of deciding whether the two target words in context refer to the same sense or not. Even though no sense inventory is explicitly given, this data set was also constructed based on WordNet. Table 12 shows a few examples from the data set.

BERT-Based Model. Given that the task in WiC is a binary classification, the 1NN model is not applicable because a training to learn sense margins is necessary. Therefore, we experimented with the BERT model fine-tuned on WiC's training data. We followed Wang et al. (2019) and fused the two sentences and fed them as input to BERT. A classifier was then trained on the concatenation of the resulting BERT contextual embeddings.

Baselines. In addition to our BERT-based model, we include results for two FastText supervised classifiers (Joulin et al. 2017) as baselines: a basic one with random initialization (FastText-B) and another initialized with FastText embeddings trained on the Common Crawl (FastText-C). As other indicative reference points, we added two language models that are enriched with WordNet (Levine et al. 2020; Loureiro and Jorge 2019b) and another with WordNet and Wikipedia (Peters et al. 2019).

Results. Table 13 shows the result of BERT models and the other baselines on the WiC benchmark.[24] We can see that BERT significantly outperforms the FastText static word embedding. The two versions of BERT (Base and Large) perform equally well on this task, achieving results close to the state of the art. As with fine-grained all-words WSD,

24 Data and results from comparison systems taken from `https://pilehvar.github.io/wic/`.

the additional knowledge drawn from WordNet proves to be beneficial, as shown by the results for KnowBERT and SenseBERT.

CoarseWSD-20: Sense Information

Table 17 shows for each sense their ID (as per their Wikipedia page title), definition, and example usage from the data set.

Complementary Results in CoarseWSD-20

1. Table 14 shows micro-F1 results for the experiment with different training data sizes sampled from the original CoarseWSD-20 training set (cf. Section 6.2.2 of the article).
2. Table 15 shows the micro-F1 performance for fine-tuning and 1NN and for varying sizes of the training data (with similar skewed distributions) for both Most Frequent Sense (MFS) and Least Frequent Sense (LFS) classes (cf. Section 6.2.2 of the article).
3. Table 16 includes the complete results for the *n*-shot experiment, including the FastText baselines (cf. Section 6.3 of the article).

Table 14
Micro-F1 results on the CoarseWSD-20 test set using training sets of different sizes sampled from the original training set.

	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	**5%**	**10%**	**25%**	**50%**	**ALL**	**1%**	**5%**	**10%**	**25%**	**50%**	**ALL**
crane	84.1	95.8	95.8	96.8	95.5	98.1	96.5	96.7	96.8	96.8	96.8	96.8
java	99.1	99.1	99.6	99.5	99.6	99.7	99.6	99.6	99.6	99.6	99.6	99.6
apple	99.4	99.4	99.5	99.5	99.5	99.6	99.2	99.2	99.2	99.2	99.2	99.2
mole	80.1	96.0	97.7	99.0	99.0	98.9	97.7	98.6	98.5	98.5	98.5	98.5
spring	95.0	97.5	96.9	96.8	97.8	98.3	98.0	98.0	98.0	98.0	97.9	97.8
chair	82.8	93.6	95.9	96.7	96.9	96.2	95.1	95.8	96.2	96.2	96.2	96.2
hood	77.6	90.7	93.5	97.2	97.6	99.6	97.2	99.0	99.4	99.6	100.0	100.0
seal	92.4	97.6	98.1	98.8	98.5	99.0	98.1	98.2	98.3	98.2	98.2	98.1
bow	74.1	92.4	96.1	96.7	97.5	98.5	94.9	95.9	95.8	95.5	95.3	95.3
club	72.8	78.7	78.7	80.4	83.5	84.7	82.0	82.9	83.8	84.0	84.4	85.1
trunk	86.2	88.7	97.8	98.7	98.7	98.3	97.8	98.2	98.4	98.7	98.7	98.7
square	88.4	87.3	92.6	92.4	92.9	95.7	93.9	94.2	94.1	95.2	95.7	96.1
arm	93.1	98.6	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4
digit	95.2	89.7	95.2	99.2	100.0	100.0	99.6	100.0	100.0	100.0	100.0	100.0
bass	92.0	93.4	94.4	95.1	95.6	95.8	86.6	86.1	85.8	85.5	85.2	84.5
yard	90.7	94.0	95.4	97.2	99.5	99.5	89.8	88.9	87.8	87.5	86.8	88.9
pound	88.3	90.0	89.7	89.0	94.9	94.9	92.6	92.8	92.0	90.4	89.7	89.7
deck	93.6	92.9	92.9	93.9	95.0	95.3	91.4	91.9	91.7	91.6	91.4	91.9
bank	95.2	95.5	97.1	99.5	99.3	99.3	99.7	99.9	99.9	99.9	99.8	99.8
pitcher	99.5	99.6	99.6	99.9	100.0	100.0	100.0	100.0	99.9	100.0	99.9	99.9
Average	89.0	93.5	95.3	96.3	97.0	97.5	95.5	95.8	95.7	95.7	95.6	95.8

Table 15
Micro-F1 performance for the two WSD strategies and for varying sizes of the training data (with similar skewed distributions) for the MFS (top) and LFS (bottom).

	Most Frequent Sense (MFS)											
	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	5%	10%	25%	50%	ALL	1%	5%	10%	25%	50%	ALL
crane	86.9	96.1	96.0	97.0	95.9	98.2	100.0	100.0	100.0	100.0	100.0	100.0
java	99.2	99.3	99.7	99.6	99.7	99.8	99.4	99.4	99.4	99.4	99.4	99.4
apple	99.5	99.5	99.6	99.6	99.6	99.7	99.5	99.5	99.5	99.5	99.5	99.5
mole	75.4	96.2	97.1	98.7	98.7	98.5	95.2	97.7	97.4	97.4	97.4	97.4
spring	96.0	97.7	97.2	97.0	98.0	98.8	97.7	97.8	97.8	97.7	97.7	97.5
chair	88.8	95.5	97.0	97.6	97.8	97.2	96.6	98.2	98.9	98.9	98.9	98.9
hood	91.6	93.4	95.6	98.3	97.9	99.7	100.0	100.0	100.0	100.0	100.0	100.0
seal	90.9	97.0	97.5	98.7	98.4	98.9	98.6	98.5	98.5	98.2	98.5	98.5
bow	85.5	97.7	97.2	98.2	98.2	98.7	97.6	98.1	98.3	98.3	98.3	98.3
club	74.6	80.4	80.6	81.9	84.1	85.2	79.5	78.5	78.5	78.4	78.2	77.8
trunk	90.6	91.4	98.2	98.9	98.9	98.6	97.9	97.9	97.9	97.9	97.9	97.9
square	89.6	88.5	93.0	92.8	93.2	95.7	93.7	93.6	93.4	95.8	95.1	94.2
arm	95.5	99.1	99.6	99.6	99.6	99.6	99.2	99.2	99.2	99.2	99.2	99.2
digit	97.0	93.9	97.1	99.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
bass	95.2	96.0	96.8	96.7	97.1	97.2	86.0	85.2	84.6	84.1	83.7	82.9
yard	94.4	96.6	97.4	98.4	99.7	99.7	88.3	86.9	85.7	85.2	84.4	86.9
pound	93.7	94.7	94.6	94.1	97.2	97.2	94.1	92.9	91.7	89.7	88.5	88.5
deck	96.7	96.3	96.3	96.8	97.3	97.5	92.4	93.0	92.1	91.7	91.3	91.3
bank	97.6	97.7	98.5	99.7	99.6	99.6	100.0	100.0	100.0	100.0	100.0	100.0
pitcher	99.8	99.8	99.8	100.0	100.0	100.0	100.0	100.0	99.9	100.0	99.9	99.9
Average	91.9	95.3	96.4	97.2	97.5	98.0	95.8	95.8	95.6	95.6	95.4	95.4

	Least Frequent Sense (LFS)											
	Fine-Tuning (BRT-L)						1NN (BRT-L)					
	1%	5%	10%	25%	50%	ALL	1%	5%	10%	25%	50%	ALL
crane	79.7	95.4	95.4	96.6	95.2	98.0	92.8	93.2	93.4	93.4	93.4	93.4
java	98.8	98.8	99.5	99.4	99.5	99.6	99.9	99.9	99.9	99.9	99.9	99.9
apple	99.2	99.2	99.3	99.3	99.4	99.5	98.7	98.7	98.7	98.7	98.7	98.7
mole	72.5	86.7	97.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
spring	95.0	98.2	97.0	97.9	97.9	97.5	97.3	97.3	97.3	97.3	97.3	97.3
chair	63.7	88.9	93.4	94.6	95.0	93.8	92.1	91.0	90.5	90.5	90.5	90.5
hood	65.7	82.5	89.2	95.2	95.2	99.2	97.0	97.3	97.7	98.5	100.0	100.0
seal	39.1	89.3	91.0	96.0	96.0	97.3	100.0	100.0	100.0	100.0	100.0	100.0
bow	0.0	73.2	95.3	94.6	98.0	99.4	91.0	98.5	100.0	100.0	100.0	100.0
club	63.5	74.4	73.3	80.0	81.6	82.5	95.2	95.2	95.2	95.2	95.2	95.2
trunk	49.7	66.6	95.2	100.0	100.0	96.5	95.2	97.1	98.2	100.0	100.0	100.0
square	9.5	21.4	4.8	20.5	30.3	76.0	53.8	58.5	57.7	56.4	69.2	84.6
arm	84.7	97.2	98.9	98.9	98.9	98.9	100.0	100.0	100.0	100.0	100.0	100.0
digit	87.7	65.5	87.2	98.0	100.0	100.0	98.1	100.0	100.0	100.0	100.0	100.0
bass	29.5	48.2	61.8	65.6	68.5	67.3	69.7	73.2	75.6	77.7	78.4	77.3
yard	70.9	74.8	79.2	90.3	98.4	98.4	98.5	100.0	100.0	100.0	100.0	100.0
pound	13.3	6.1	0.0	11.1	69.3	70.6	80.0	92.0	95.0	96.7	100.0	100.0
deck	16.7	0.0	0.0	43.6	57.1	58.6	78.6	77.1	85.7	90.5	92.9	100.0
bank	2.9	14.0	51.4	94.4	91.8	91.6	93.9	97.3	97.7	97.0	95.5	95.5
pitcher	0.0	4.8	28.0	93.0	98.7	94.6	100.0	100.0	100.0	100.0	100.0	100.0
Average	52.1	64.3	71.9	83.4	88.5	91.0	91.6	93.3	94.1	94.6	95.5	96.6

Table 16
Micro- and macro-F1 results in the *n*-shot setting for all the two BERT-based WSD strategies (as well as for the static embedding baseline) in our experiments and for all the words in the data set. Results are the average of three runs (standard deviation is shown in parentheses).

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
crane	Static emb.	Fasttext-B	48.6 (5.3)	57.5 (5.3)	57.7 (3.5)	70.9 (5.2)	48.1 (4.5)	57.9 (4.8)	58.5 (3.3)	71.1 (5.3)
		Fasttext-C	52.7 (1.2)	69.4 (6.3)	82.0 (3.0)	83.4 (6.6)	51.4 (1.1)	69.6 (6.3)	81.9 (3.0)	83.5 (6.8)
	1NN	BERT-Base	84.5 (9.8)	93.8 (1.8)	93.6 (1.4)	94.5 (0.3)	84.3 (10.1)	93.7 (1.9)	93.4 (1.4)	94.4 (0.3)
		BERT-Large	86.4 (3.8)	94.7 (3.5)	95.5 (1.4)	96.8 (0.9)	86.4 (3.9)	94.5 (3.7)	95.4 (1.4)	96.7 (0.9)
	Fine-Tuning	BERT-Base	65.6 (1.9)	88.7 (7.8)	94.1 (3.2)	95.8 (0.6)	63.4 (1.3)	88.7 (7.8)	94.0 (3.3)	95.7 (0.6)
		BERT-Large	74.7 (10.5)	92.6 (2.2)	95.3 (1.7)	96.4 (1.3)	73.2 (12.3)	92.5 (2.2)	95.3 (1.7)	96.4 (1.3)
java	Static emb.	Fasttext-B	63.1 (1.3)	66.8 (2.4)	68.5 (2.8)	80.6 (6.8)	55.8 (3.5)	60.5 (3.9)	66.9 (1.6)	80.9 (6.4)
		Fasttext-C	78.4 (5.8)	90.1 (3.3)	90.9 (1.9)	94.9 (2.3)	78.1 (7.8)	90.3 (2.5)	90.5 (2.3)	95.1 (2.1)
	1NN	BERT-Base	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.7 (0.0)	99.7 (0.0)
		BERT-Large	99.6 (0.1)	99.6 (0.1)	99.6 (0.0)	99.6 (0.0)	99.7 (0.0)	99.7 (0.1)	99.7 (0.0)	99.7 (0.0)
	Fine-Tuning	BERT-Base	99.2 (0.4)	98.8 (0.7)	99.4 (0.2)	99.3 (0.1)	99.1 (0.5)	98.8 (0.7)	99.4 (0.2)	99.3 (0.1)
		BERT-Large	99.2 (0.6)	99.4 (0.1)	99.5 (0.1)	99.6 (0.1)	99.1 (0.6)	99.4 (0.1)	99.5 (0.1)	99.5 (0.1)
apple	Static emb.	Fasttext-B	43.1 (2.1)	52.0 (4.9)	55.2 (8.4)	74.7 (1.2)	45.4 (3.8)	55.3 (5.1)	61.3 (5.4)	73.9 (2.4)
		Fasttext-C	71.2 (9.9)	81.2 (2.9)	87.3 (2.1)	93.2 (0.2)	63.7 (12.8)	80.7 (1.8)	86.7 (3.0)	92.4 (0.3)
	1NN	BERT-Base	95.5 (3.9)	99.0 (0.1)	99.0 (0.1)	99.0 (0.0)	96.1 (3.2)	99.0 (0.1)	99.0 (0.1)	99.0 (0.0)
		BERT-Large	98.1 (1.2)	99.2 (0.0)	99.3 (0.1)	99.3 (0.1)	98.3 (0.9)	99.2 (0.1)	99.2 (0.1)	99.3 (0.1)
	Fine-Tuning	BERT-Base	90.7 (9.0)	98.3 (0.6)	99.0 (0.1)	99.0 (0.1)	89.6 (10.3)	98.2 (0.7)	98.9 (0.1)	98.9 (0.1)
		BERT-Large	91.5 (5.5)	96.4 (2.5)	99.3 (0.1)	99.1 (0.5)	90.7 (6.2)	96.2 (2.6)	99.2 (0.1)	99.0 (0.5)
mole	Static emb.	Fasttext-B	21.2 (9.9)	16.3 (7.3)	38.2 (1.1)	65.9 (1.6)	17.9 (2.2)	22.8 (3.9)	41.5 (9.2)	68.8 (2.0)
		Fasttext-C	48.7 (2.6)	63.3 (4.3)	75.9 (6.3)	88.0 (0.9)	57.3 (1.3)	68.6 (2.5)	79.4 (4.3)	89.4 (1.7)
	1NN	BERT-Base	75.9 (5.5)	91.1 (4.0)	95.1 (2.2)	97.4 (0.6)	84.9 (4.6)	93.9 (1.8)	96.6 (1.2)	97.7 (0.3)
		BERT-Large	89.3 (1.1)	95.6 (0.8)	98.1 (0.8)	98.5 (0.0)	93.4 (0.6)	97.1 (0.7)	98.8 (0.4)	99.0 (0.0)
	Fine-Tuning	BERT-Base	71.2 (4.1)	86.2 (5.2)	95.8 (1.3)	97.6 (0.4)	70.7 (4.8)	87.8 (3.3)	95.8 (1.4)	97.5 (0.6)
		BERT-Large	77.3 (2.2)	88.7 (4.3)	96.3 (0.9)	98.5 (0.7)	76.4 (2.0)	90.2 (2.9)	96.3 (1.0)	98.8 (0.7)
spring	Static emb.	Fasttext-B	33.0 (6.8)	43.8 (8.2)	46.4 (7.4)	67.0 (2.2)	35.4 (0.5)	32.8 (0.7)	35.8 (3.6)	66.7 (3.1)
		Fasttext-C	46.0 (14.6)	57.6 (4.0)	73.5 (3.7)	83.7 (2.8)	45.0 (9.0)	64.2 (3.3)	76.5 (3.5)	86.1 (2.8)

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
	1NN	BERT-Base	94.4 (2.0)	97.2 (0.5)	97.1 (0.3)	97.3 (0.1)	94.6 (2.2)	97.3 (0.9)	97.0 (0.4)	97.3 (0.1)
		BERT-Large	97.1 (1.5)	97.9 (0.5)	97.6 (0.4)	97.7 (0.2)	96.8 (1.6)	97.8 (0.2)	97.5 (0.3)	97.8 (0.1)
	Fine-Tuning	BERT-Base	75.2 (4.7)	92.9 (0.3)	96.0 (0.5)	95.3 (0.5)	73.9 (4.3)	92.9 (0.2)	96.1 (0.5)	95.2 (0.6)
		BERT-Large	80.3 (10.1)	94.2 (2.7)	97.0 (0.7)	97.2 (0.2)	77.1 (12.6)	94.4 (2.4)	97.1 (0.6)	97.1 (0.4)
chair	Static emb.	Fasttext-B	62.8 (9.0)	73.8 (6.6)	74.4 (4.5)	74.4 (5.2)	58.4 (6.9)	68.4 (5.1)	68.4 (4.3)	72.1 (4.2)
		Fasttext-C	76.2 (8.0)	75.4 (4.5)	81.3 (2.0)	83.6 (2.4)	64.1 (12.8)	72.9 (0.2)	75.8 (0.9)	81.7 (2.2)
	1NN	BERT-Base	88.7 (7.1)	95.9 (0.7)	95.9 (0.4)	95.6 (0.4)	84.2 (11.6)	94.5 (0.5)	94.5 (0.3)	94.3 (0.3)
		BERT-Large	82.3 (19.0)	94.6 (1.3)	96.2 (0.0)	95.9 (0.4)	84.4 (14.1)	93.5 (0.9)	94.7 (0.0)	94.5 (0.3)
	Fine-Tuning	BERT-Base	84.1 (11.3)	91.0 (5.7)	95.6 (0.7)	96.7 (0.4)	75.6 (21.8)	90.1 (5.8)	94.9 (0.8)	96.1 (0.4)
		BERT-Large	72.1 (11.9)	93.3 (1.5)	94.4 (2.0)	95.1 (1.5)	65.4 (12.9)	92.4 (1.7)	93.6 (2.1)	94.4 (1.6)
hood	Static emb.	Fasttext-B	56.1 (11.3)	33.3 (16.2)	47.6 (18.7)	–	48.8 (4.1)	42.1 (6.3)	51.4 (5.1)	–
		Fasttext-C	66.3 (5.0)	77.6 (2.1)	86.6 (5.0)	–	61.5 (6.2)	73.7 (4.7)	82.1 (6.8)	–
	1NN	BERT-Base	96.8 (3.0)	98.4 (0.6)	98.8 (0.0)	–	94.8 (5.9)	98.0 (0.7)	98.5 (0.0)	–
		BERT-Large	96.3 (4.3)	99.2 (0.6)	99.6 (0.6)	–	92.7 (9.3)	99.0 (0.7)	99.5 (0.7)	–
	Fine-Tuning	BERT-Base	86.6 (3.6)	95.5 (2.5)	97.6 (1.7)	–	79.0 (6.4)	94.4 (3.0)	96.7 (2.4)	–
		BERT-Large	89.8 (5.8)	96.3 (1.0)	98.8 (1.0)	–	80.1 (16.2)	95.1 (1.5)	98.0 (1.7)	–
seal	Static emb.	Fasttext-B	29.9 (1.0)	31.6 (3.2)	39.9 (10.0)	60.2 (4.2)	25.0 (0.0)	25.7 (1.0)	39.8 (5.6)	57.4 (1.1)
		Fasttext-C	46.4 (8.1)	64.6 (4.0)	73.7 (2.3)	82.1 (1.6)	43.6 (11.2)	67.7 (5.1)	79.0 (2.4)	85.4 (2.8)
	1NN	BERT-Base	91.5 (6.8)	96.1 (0.7)	96.4 (0.4)	96.6 (0.3)	89.4 (11.0)	97.0 (0.6)	97.3 (0.3)	97.5 (0.3)
		BERT-Large	96.1 (1.8)	97.0 (0.7)	98.0 (0.6)	98.2 (0.1)	96.5 (1.5)	97.7 (0.6)	98.4 (0.5)	98.6 (0.1)
	Fine-Tuning	BERT-Base	79.6 (7.9)	95.0 (1.0)	94.4 (1.5)	96.6 (0.5)	72.3 (12.5)	90.0 (1.5)	88.7 (3.4)	92.3 (0.8)
		BERT-Large	76.2 (12.4)	94.0 (0.9)	94.6 (2.4)	97.1 (0.6)	68.0 (16.4)	89.0 (3.5)	88.3 (4.4)	94.0 (1.4)
bow	Static emb.	Fasttext-B	29.8 (16.7)	41.2 (20.6)	40.2 (9.1)	63.3 (4.3)	39.1 (7.6)	35.2 (2.3)	39.5 (7.9)	64.8 (1.5)
		Fasttext-C	52.3 (8.1)	62.6 (4.6)	79.4 (0.8)	87.6 (1.2)	49.3 (7.2)	60.5 (6.1)	76.4 (1.8)	87.1 (1.5)
	1NN	BERT-Base	86.5 (1.3)	91.8 (2.7)	95.5 (0.2)	95.3 (0.4)	80.7 (4.2)	88.6 (2.0)	93.7 (1.5)	95.0 (0.5)
		BERT-Large	87.4 (1.7)	94.9 (2.3)	97.4 (0.6)	96.4 (1.0)	84.3 (4.2)	93.3 (4.5)	97.8 (0.5)	96.7 (1.0)

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
	Fine-Tuning	BERT-Base	83.1 (3.1)	89.5 (4.1)	94.0 (0.8)	96.1 (0.2)	73.2 (6.4)	85.6 (4.9)	93.1 (1.5)	95.3 (0.4)
		BERT-Large	78.8 (13.6)	91.0 (3.9)	96.7 (0.7)	97.5 (0.6)	73.1 (12.0)	91.0 (2.2)	96.9 (0.9)	97.5 (1.1)
club	Static emb.	Fasttext-B	35.0 (11.8)	26.2 (19.3)	23.8 (5.5)	56.1 (4.1)	31.5 (1.0)	32.6 (1.0)	37.1 (1.5)	54.4 (1.2)
		Fasttext-C	35.2 (8.1)	47.7 (5.4)	60.2 (3.8)	74.6 (2.8)	37.4 (1.9)	52.3 (3.6)	61.7 (2.5)	79.0 (2.3)
	1NN	BERT-Base	58.3 (5.5)	80.2 (1.9)	81.2 (1.1)	81.2 (0.8)	70.5 (2.8)	84.9 (2.3)	84.9 (1.7)	84.7 (1.0)
		BERT-Large	54.1 (10.5)	82.8 (3.5)	83.8 (1.8)	84.5 (0.2)	69.0 (7.7)	87.4 (3.2)	88.9 (1.7)	88.5 (0.3)
	Fine-Tuning	BERT-Base	52.5 (6.1)	68.5 (8.9)	80.5 (1.5)	84.2 (0.8)	50.0 (6.2)	68.3 (10.0)	79.9 (1.9)	83.4 (0.8)
		BERT-Large	51.3 (7.3)	71.8 (1.9)	81.2 (3.9)	83.7 (1.9)	49.8 (6.5)	69.5 (2.7)	80.8 (4.1)	83.4 (1.5)
	Static emb.	Fasttext-B	32.9 (16.4)	21.2 (0.6)	45.9 (17.7)	66.7 (3.4)	34.0 (2.1)	35.4 (2.0)	43.5 (7.4)	67.2 (2.8)
		Fasttext-C	65.4 (6.8)	63.2 (7.4)	76.6 (2.1)	82.7 (3.7)	65.3 (8.8)	67.0 (7.9)	78.3 (0.7)	87.4 (2.8)
trunk	1NN	BERT-Base	77.9 (18.4)	84.8 (7.1)	94.8 (1.8)	95.2 (2.2)	84.1 (12.1)	91.2 (4.6)	97.2 (1.0)	97.4 (1.2)
		BERT-Large	83.1 (20.2)	96.1 (1.1)	96.5 (1.2)	98.3 (0.6)	89.7 (11.5)	97.9 (0.6)	98.1 (0.7)	99.1 (0.3)
	Fine-Tuning	BERT-Base	72.7 (16.7)	89.2 (5.4)	93.9 (1.6)	97.0 (1.6)	72.5 (11.2)	89.1 (4.6)	93.5 (1.6)	96.7 (1.8)
		BERT-Large	79.2 (14.7)	91.3 (2.4)	95.2 (1.2)	96.1 (1.8)	79.9 (11.7)	90.9 (2.4)	94.9 (1.4)	95.7 (2.0)
	Static emb.	Fasttext-B	20.5 (7.2)	8.9 (3.6)	38.5 (10.3)	–	25.9 (0.8)	25.0 (0.0)	38.8 (1.9)	–
		Fasttext-C	40.1 (19.0)	60.4 (10.8)	71.8 (3.2)	–	39.4 (7.1)	61.5 (5.5)	74.6 (2.9)	–
square	1NN	BERT-Base	70.2 (7.7)	83.3 (7.8)	90.7 (3.0)	–	74.1 (10.3)	83.7 (5.1)	88.6 (2.2)	–
		BERT-Large	74.9 (13.0)	84.7 (8.3)	93.7 (1.2)	–	79.4 (4.6)	84.4 (5.6)	89.0 (4.0)	–
	Fine-Tuning	BERT-Base	64.9 (11.3)	75.4 (11.3)	85.2 (3.6)	–	56.7 (4.3)	71.0 (8.1)	81.3 (4.0)	–
		BERT-Large	63.9 (17.0)	81.0 (10.2)	87.3 (2.0)	–	57.7 (7.3)	76.9 (8.2)	82.9 (1.8)	–
	Static emb.	Fasttext-B	53.7 (11.1)	60.0 (3.9)	53.3 (8.9)	80.3 (2.0)	58.6 (1.9)	61.4 (3.5)	62.3 (3.3)	79.9 (2.5)
		Fasttext-C	79.1 (7.1)	85.4 (7.0)	90.9 (4.7)	95.7 (0.5)	85.1 (5.1)	86.6 (5.5)	91.6 (3.0)	95.1 (0.8)
arm	1NN	BERT-Base	99.4 (0.00)	99.4 (0.0)	99.4 (0.0)	99.4 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)
		BERT-Large	99.4 (0.00)	99.4 (0.0)	99.4 (0.0)	99.4 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)	99.6 (0.0)
	Fine-Tuning	BERT-Base	96.3 (2.8)	99.4 (0.0)	99.4 (0.0)	99.4 (0.0)	95.0 (3.9)	99.2 (0.0)	99.2 (0.0)	99.2 (0.0)
		BERT-Large	97.4 (2.1)	98.8 (0.9)	99.4 (0.0)	99.4 (0.0)	96.4 (2.9)	98.4 (1.1)	99.2 (0.0)	99.2 (0.0)

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
digit	Static emb.	Fasttext-B	45.2 (5.1)	54.0 (16.8)	43.7 (9.0)	–	49.0 (5.1)	69.4 (11.1)	62.8 (5.0)	–
		Fasttext-C	69.8 (9.2)	84.9 (8.1)	87.3 (5.9)	–	63.3 (3.8)	85.0 (5.5)	89.2 (2.7)	–
	1NN	BERT-Base	96.8 (2.2)	99.2 (1.1)	100 (0.0)	–	92.6 (5.2)	98.1 (2.6)	100 (0.0)	–
		BERT-Large	94.4 (6.3)	100 (0.0)	100 (0.0)	–	87.0 (14.6)	100 (0.0)	100 (0.0)	-
	Fine-Tuning	BERT-Base	96.0 (2.2)	100 (0.0)	100 (0.0)	–	93.6 (4.1)	100 (0.0)	100 (0.0)	–
		BERT-Large	95.2 (5.1)	98.4 (1.1)	100 (0.0)	–	91.2 (10.0)	97.5 (1.7)	100 (0.0)	–
bass	Static emb.	Fasttext-B	27.8 (8.9)	22.2 (0.5)	33.5 (15.3)	60.7 (3.6)	39.2 (3.6)	36.8 (2.6)	39.2 (6.7)	71.4 (1.1)
		Fasttext-C	37.1 (8.2)	49.8 (5.3)	65.1 (6.2)	78.9 (3.2)	49.4 (3.9)	57.5 (3.3)	67.1 (1.8)	78.3 (2.2)
	1NN	BERT-Base	65.6 (17.9)	75.2 (7.1)	70.4 (5.7)	77.2 (2.1)	60.4 (3.2)	71.8 (5.8)	71.2 (1.6)	77.2 (0.5)
		BERT-Large	70.2 (17.8)	76.9 (6.5)	77.1 (4.5)	83.4 (4.2)	70.3 (4.6)	78.6 (4.5)	80.3 (1.9)	83.4 (0.6)
	Fine-Tuning	BERT-Base	63.0 (12.3)	67.8 (5.5)	82.5 (1.8)	86.5 (1.2)	54.2 (2.6)	62.2 (3.9)	71.1 (1.4)	76.2 (0.8)
		BERT-Large	79.4 (15.5)	70.4 (10.0)	76.4 (7.5)	88.1 (2.8)	67.6 (7.7)	65.0 (5.3)	68.7 (5.2)	78.5 (2.7)
yard	Static emb.	Fasttext-B	48.6 (10.8)	35.6 (2.6)	40.3 (12.3)	–	56.0 (7.0)	54.6 (7.5)	61.0 (5.6)	–
		Fasttext-C	63.4 (26.2)	74.1 (9.8)	83.8 (9.2)	–	62.3 (13.9)	78.5 (7.2)	86.7 (6.2)	–
	1NN	BERT-Base	52.8 (15.8)	70.8 (11.9)	77.3 (2.9)	–	68.4 (13.0)	82.8 (7.1)	86.6 (1.7)	–
		BERT-Large	65.3 (20.5)	81.0 (14.3)	85.6 (7.6)	–	79.5 (12.1)	88.8 (8.4)	91.5 (4.5)	–
	Fine-Tuning	BERT-Base	56.0 (13.6)	67.1 (15.9)	92.1 (4.6)	–	48.3 (10.5)	62.1 (13.1)	87.8 (6.2)	–
		BERT-Large	48.6 (8.6)	82.9 (9.6)	90.7 (5.7)	–	46.7 (7.2)	77.4 (9.7)	85.9 (7.8)	–
pound	Static emb.	Fasttext-B	57.7 (20.7)	39.9 (18.3)	40.5 (3.4)	–	57.3 (7.0)	51.7 (2.6)	55.1 (2.0)	–
		Fasttext-C	61.2 (20.5)	58.8 (5.8)	68.7 (5.1)	–	57.7 (7.5)	62.3 (2.9)	73.7 (3.8)	–
	1NN	BERT-Base	61.9 (19.0)	66.3 (14.5)	77.7 (9.1)	–	55.1 (6.6)	81.2 (8.1)	86.1 (4.1)	–
		BERT-Large	69.4 (26.0)	69.4 (8.5)	82.1 (5.1)	–	59.4 (3.0)	81.5 (6.1)	90.0 (2.8)	–
	Fine-Tuning	BERT-Base	61.9 (10.2)	63.6 (16.6)	74.2 (9.7)	–	45.1 (4.7)	52.8 (10.7)	64.5 (8.2)	–
		BERT-Large	64.6 (17.2)	59.1 (2.6)	81.1 (9.4)	–	47.9 (5.9)	51.4 (1.5)	70.0 (8.0)	–

Table 16
(*continued*)

			Micro F1				Macro F1			
			1	3	10	30	1	3	10	30
deck	Static emb.	Fasttext-B	54.6 (17.2)	73.1 (10.8)	71.0 (10.5)	–	51.4 (7.9)	54.7 (4.3)	62.4 (2.6)	–
		Fasttext-C	68.4 (32.0)	66.0 (13.0)	77.8 (3.3)	–	61.0 (2.1)	61.9 (4.1)	72.6 (10.0)	–
	1NN	BERT-Base	81.8 (12.0)	85.2 (8.1)	86.9 (1.4)	–	81.4 (9.0)	81.0 (4.6)	90.7 (2.8)	–
		BERT-Large	76.4 (15.9)	87.5 (2.5)	90.9 (0.8)	–	78.5 (2.9)	84.5 (5.3)	95.1 (0.4)	–
	Fine-Tuning	BERT-Base	86.9 (7.3)	87.9 (1.4)	86.9 (2.2)	–	70.8 (11.9)	69.9 (1.3)	70.3 (3.6)	–
		BERT-Large	77.4 (17.8)	88.6 (2.4)	90.6 (2.7)	–	55.2 (10.4)	63.8 (12.2)	77.7 (4.0)	–
bank	Static emb.	Fasttext-B	46.5 (12.1)	46.9 (4.6)	51.0 (8.8)	77.8 (5.5)	56.8 (2.0)	64.9 (1.2)	62.7 (2.7)	72.5 (3.7)
		Fasttext-C	38.4 (6.8)	70.1 (11.9)	80.7 (2.8)	88.2 (4.6)	61.9 (0.3)	72.8 (4.6)	79.1 (3.4)	85.9 (2.4)
	1NN	BERT-Base	98.8 (0.7)	99.4 (0.5)	99.7 (0.1)	99.8 (0.0)	94.3 (3.3)	95.4 (4.9)	97.7 (0.1)	97.7 (0.0)
		BERT-Large	99.0 (0.7)	99.5 (0.1)	99.9 (0.1)	99.9 (0.1)	95.9 (3.5)	97.6 (1.8)	99.2 (1.1)	99.2 (1.1)
	Fine-Tuning	BERT-Base	91.9 (5.7)	97.8 (1.4)	97.9 (0.8)	99.0 (0.1)	76.1 (9.4)	89.4 (5.2)	90.5 (3.3)	95.3 (0.5)
		BERT-Large	58.0 (28.9)	98.6 (0.7)	99.5 (0.1)	98.6 (0.6)	52.3 (28.3)	92.3 (4.0)	97.3 (0.5)	93.5 (2.4)
pitcher	Static emb.	Fasttext-B	92.1 (2.2)	82.7 (9.3)	82.8 (1.3)	–	69.2 (8.0)	73.4 (10.1)	82.4 (4.3)	–
		Fasttext-C	96.5 (4.2)	95.9 (1.8)	91.2 (3.0)	–	84.2 (13.5)	94.1 (0.9)	94.3 (3.0)	–
	1NN	BERT-Base	100 (0.0)	99.9 (0.1)	99.8 (0.0)	–	100 (0.0)	99.9 (0.0)	99.9 (0.0)	–
		BERT-Large	100 (0.0)	100 (0.0)	99.9 (0.0)	–	100 (0.0)	100 (0.0)	100 (0.0)	–
	Fine-Tuning	BERT-Base	98.6 (0.5)	98.9 (0.6)	97.2 (1.1)	–	70.6 (5.3)	76.2 (10.2)	62.8 (4.0)	–
		BERT-Large	97.1 (2.3)	98.7 (1.2)	98.5 (1.1)	–	70.5 (17.3)	77.3 (13.8)	74.8 (13.6)	–

Table 17
Sense definitions in the CoarseWSD-20 data set. Each sense is accompanied with an example usage from the data set. Sense IDs correspond to the current Wikipedia page of each sense by the date of the submission.

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
Crane	crane$_1$	crane (machine)	A crane is a type of machine, generally equipped with a hoist rope, wire ropes or chains, and sheaves, that can be used both to lift and lower materials and to move them horizontally.	launching and recovery is accomplished with the assistance of a shipboard crane .
	crane$_2$	crane (bird)	Cranes are a family, the Gruidae, of large, long-legged, and long-necked birds in the group Gruiformes.	tibet hosts species of wolf , wild donkey , crane, vulture , hawk , geese , snake , and buffalo .
Java	java$_1$	java	Java is an island of Indonesia, bordered by the Indian Ocean on the south and the Java Sea on the north.	in indonesia , only sumatra , borneo, and papua are larger in territory , and only java and sumatra have larger populations .
	java$_2$	java (programming language)	Java is a general-purpose programming language that is class- based, object-oriented, and designed to have as few implementation dependencies as possible.	examples include the programming languages perl , java and lua .
Apple	apple$_1$	apple inc.	Apple Inc. is an American multinational technology company headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software, and online services.	shopify released a free mobile app on the apple app store on may 13, 2010.
	apple$_2$	apple	An apple is an edible fruit produced by an apple tree.	cherry, apple, pear, peach and apricot trees are available.
Mole	mole$_1$	mole (animal)	Moles are small mammals adapted to a subterranean lifestyle (i.e., fossorial).	its primary prey consists of mice, rat, squirrel, chipmunk, shrew, mole and rabbits.
	mole$_2$	mole (espionage)	In espionage jargon, a mole is a long-term spy who is recruited before having access to secret intelligence, subsequently managing to get into the target organization.	philip meets claudia where she tells him that there is a mole working for the fbi.
	mole$_3$	mole (unit)	The mole (symbol: mol) is the unit of measurement for amount of substance in the International System of Units (SI).	so the specific heat of a classical solid is always 3k per atom , or in chemistry units, 3r per mole of atoms .
	mole$_4$	mole sauce	Mole is a traditional marinade and sauce originally used in Mexican cuisine.	food such as cake, chicken with mole, hot chocolate, coffee, and atole are served .

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
	$mole_5$	mole (architecture)	A mole is a massive structure, usually of stone, used as a pier, breakwater, or a causeway between places separated by water.	the islands of pomgues and ratonneau are connected by a mole built in 1822.
Spring	$spring_1$	spring (hydrology)	A spring is a point at which water flows from an aquifer to the Earth's surface. It is a component of the hydrosphere.	the village was famous for its mineral water spring used for healing in sanatorium , including the hawthorne and lithia springs .
	$spring_2$	spring (season)	Spring, also known as springtime, is one of the four temperate seasons, succeeding winter and preceding summer.	the species is most active during the spring and early summer although it may be seen into late june .
	$spring_3$	spring (device)	A spring is an elastic object that stores mechanical energy.	often spring are used to reduce backlash of the mechanism .
Chair	$chair_1$	chairman	The chairperson (also chair, chairman, or chairwoman) is the presiding officer of an organized group such as a board, committee, or deliberative assembly.	gan is current chair of the department of environmental sciences at university of california , riverside .
	$chair_2$	chair	One of the basic pieces of furniture, a chair is a type of seat.	a typical western living room may contain furnishings such as a sofa , chair , occasional table , and bookshelves , electric lamp , rugs , or other furniture .
Hood	$hood_1$	hood (comics)	Hood (real name Parker Robbins) is a fictional character, a supervillain, and a crime boss appearing in American comic books published by Marvel Comics.	the hood has hired him as part of his criminal organization to take advantage of the split in the superhero community caused by the super-human registration act .
	$hood_2$	hood (vehicle)	The hood (North American English) or bonnet (Commonwealth English excluding Canada) is the hinged cover over the engine of motor vehicles that allows access to the engine compartment, or trunk (boot in Commonwealth English) on rear-engine and some mid-engine vehicles) for maintenance and repair.	european versions of the car also had an air intake on the hood .
	$hood_3$	hood (headgear)	A hood is a kind of headgear that covers most of the head and neck, and sometimes the face.	in some sauna suits , the jacket also includes a hood to provide additional retention of body heat.

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
Seal	$seal_1$	pinniped	Pinnipeds, commonly known as seals, are a widely distributed and diverse clade of carnivorous, fin-footed, semiaquatic marine mammals.	animals such as shark , stingray , weever fish , seal and jellyfish can sometimes present a danger .
	$seal_2$	seal (musician)	Henry Olusegun Adeola Samue (born 19 February 1963), known professionally as Seal, is a British singer-songwriter.	she was married to english singer seal from 2005 until 2012 .
	$seal_3$	seal (emblem)	A seal is a device for making an impression in wax, clay, paper, or some other medium, including an embossment on paper, and is also the impression thus made.	each level must review , add information as necessary , and stamp or seal that the submittal was examined and approved by that party .
	$seal_4$	seal (mechanical)	A mechanical seal is a device that helps join systems or mechanisms together by preventing leakage (e.g. in a pumping system), containing pressure, or excluding contamination.	generally speaking , standard ball joints will outlive sealed ones because eventually the seal will break , causing the joint to dry out and rust .
Bow	bow_1	bow (ship)	The bow is the forward part of the hull of a ship or boat.	the stem is the most forward part of a boat or ship's bow and is an extension of the keel itself .
	bow_2	bow and arrow	The bow and arrow is a ranged weapon system consisting of an elastic launching device (bow) and long-shafted projectiles (arrows).	bow and arrow used in warfare .
	bow_3	bow (music)	In music, a bow is a tensioned stick which has hair (usually horse-tail hair) coated in rosin (to facilitate friction) affixed to it.	horsehair is used for brush , the bow of musical instruments and many other things .
Club	$club_1$	club	A club is an association of people united by a common interest or goal.	this is a partial list of women's association football club teams from all over the world sorted by confederation .
	$club_2$	nightclub	A nightclub, music club, or club, is an entertainment venue and bar that usually operates late into the night.	although several of his tracks were club hits, he had limited chart success.
	$club_3$	club (weapon)	A club (also known as a cudgel, baton, bludgeon, truncheon, cosh, nightstick or impact weapon) is among the simplest of all weapons: a short staff or stick, usually made of wood, wielded as a weapon since prehistoric times.	before their adoption of guns, the plains indians hunted with spear, bows and arrows, and various forms of club.

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
Trunk	$trunk_1$	trunk (botany)	In botany, the trunk (or bole) is the stem and main wooden axis of a tree.	its leaves are different from the leaves of true palms, and unlike true palms it does not develop a woody trunk.
	$trunk_2$	trunk (automobile)	The trunk (North American English), boot (British English), dickey (Indian English) (also spelled dicky or diggy) or compartment (South-East Asia) of a car is the vehicle's main storage or cargo compartment.	unlike the bmw x5, the x-coupe had an aluminium body, a trunk opening downwards and two doors that swing outward.
	$trunk_3$	trunk (anatomy)	The torso or trunk is an anatomical term for the central part orx core of many animal bodies (including humans) from which extend the neck and limbs.	surface projections of the major organs of the trunk, using the vertebral column and rib cage as main reference points of superficial anatomy.
Square	$square_1$	square	In geometry, a square is a regular quadrilateral, which means that it has four equal sides and four equal angles (90-degree angles, or 100-gradian angles or right angles).	similarly , a square with all sides of length has the perimeter and the same area as the rectangle.
	$square_2$	square (company)	Square Co., Ltd. was a Japanese video game company founded in September 1986 by Masafumi Miyamoto. It merged with Enix in 2003 to form Square Enix.	video game by square , features the orbital elevator " a.t.l.a.s. ".
	$square_3$	town square	A town square is an open public space commonly found in the heart of a traditional town used for community gatherings.	here is a partial list of notable expressways, tunnel, bridge, road, avenues, street, crescent, square and bazaar in hong kong.
	$square_4$	square number	In mathematics, a square number or perfect square is an integer that is the square of an integer.	in mathematics eighty-one is the square of 9 and the fourth power of 3.
Arm	arm_1	arm architecture	Arm (previously officially written all caps as ARM and usually written as such today), previously Advanced RISC Machine, originally Acorn RISC Machine, is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments.	windows embedded compact is available for arm, mips, superh and x86 processor architectures.

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
	arm_2	arm	In human anatomy, the arm is the part of the upper limb between the glenohumeral joint (shoulder joint) and the elbow joint.	on the human body, the limb can be divided into segments, such as the arm and the forearm of the upper limb, and the thigh and the leg of the lower slimb.
Digit	$digit_1$	numerical digit	A numerical digit is a single symbol (such as "2" or "5") used alone, or in combinations (such as "25"), to represent numbers (such as the number 25) according to some positional numeral systems.	it uses the digit 0 , 1 , 2 and 3 to represent any real number.
	$digit_2$	digit (anatomy)	A digit is one of several most distal parts of a limb, such as fingers or toes, present in many vertebrates.	a finger is a limb of the human body and a type of digit, an organ of and found in the hand of human and other primate.
Bass	$bass_1$	bass (guitar)	The bass guitar, electric bass, or simply bass, is the lowest-pitched member of the guitar family.	the band decided to continue making music after thirsk's death, and brought in bass guitarist randy bradbury from one hit wonder.
	$bass_2$	bass (voice type)	A bass is a type of classical male singing voice and has the lowest vocal range of all voice types.	he is known for his distinctive and untrained bass voice.
	$bass_3$	double bass	The double bass, also known simply as the bass (or by other names), is the largest and lowest-pitched bowed (or plucked) string instrument in the modern symphony orchestra.	his instruments were the bass and the tuba.
Yard	$yard_1$	yard	The yard (abbreviation: yd) is an English unit of length, in both the British imperial and US customary systems of measurement, that comprises 3 feet or 36 inches.	accuracy is sufficient for hunting small game at ranges to 50 yard.
	$yard_2$	yard (sailing)	A yard is a spar on a mast from which sails are set.	aubrey improves sophie sailing qualities by adding a longer yard which allows him to spread a larger mainsail.

Table 17
(continued)

Word	Sense #	Sense ID	**Definition** (1st sentence from Wikipedia)	**Example usage** (tokenized)
Pound	$pound_1$	pound (mass)	The pound or pound-mass is a unit of mass used in the imperial, United States customary and other systems of measurement.	it is approximately 16.38 kilogram (36.11 pound).
	$pound_2$	pound (currency)	A pound is any of various units of currency in some nations.	in english, the maltese currency was referred to as the pound originally and for many locals this usage continued.
Deck	$deck_1$	deck (ship)	A deck is a permanent covering over a compartment or a hull of a ship.	the protective deck was thick and ran the full length of the ship.
	$deck_2$	deck (building)	In architecture, a deck is a flat surface capable of supporting weight, similar to a floor, but typically constructed outdoors, often elevated from the ground, and usually connected to a building.	typically , it is a wooden deck near a hiking trail that provides the hikers a clean and even place to sleep.
Bank	$bank_1$	bank	A bank is a financial institution that accepts deposits from the public and creates a demand deposit, while simultaneously making loans.	the bank , which loans money to the player after they have a house for collateral .
	$bank_2$	bank (geography)	In geography, a bank is the land alongside a body of water.	singapore's first market was located at the south bank of the singapore river.
Pitcher	$pitcher_1$	pitcher	In baseball, the pitcher is the player who throws the baseball from the pitcher's mound toward the catcher to begin each play, with the goal of retiring a batter, who attempts to either make contact with the pitched ball or draw a walk.	kasey garret olemberger (born march 18 , 1978) is an italian american professional baseball pitcher.
	$pitcher_2$	pitcher (container)	In American English, a pitcher is a container with a spout used for storing and pouring liquids.	pottery was found as grave goods, including combinations of pitcher and cup.

Acknowledgments

We would like to thank Claudio Delli Bovi and Miguel Ballesteros for early pre-BERT discussions on the topic of ambiguity and language models. We would also like to thank the anonymous reviewers for their comments and suggestions that helped improve the article. Daniel Loureiro is supported by the European Union and Fundação para a Ciência e Tecnologia through contract DFA/BD/9028/2020 (Programa Operacional Regional Norte). Jose Camacho-Collados is supported by a UKRI Future Leaders Fellowship.

References

Agirre, Eneko, 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 *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 29–33, Melbourne. https://doi.org/10.18653/v1/W18-2505

Agirre, Eneko, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84. https://doi.org/10.1162/COLI_a_00164

Agirre, Eneko, Lluís Màrquez, and Richard Wicentowski, editors. 2007. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). Prague. https://doi.org/10.3115/1621474.1621476

Aina, Laura, 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. https://doi.org/10.18653/v1/P19-1324

Akbik, Alan, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, NM.

Amrami, Asaf 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. https://doi.org/10.18653/v1/D18-1523

Apidianaki, Marianna. 2008. Translation-oriented word sense induction based on parallel corpora. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), pages 3269–3275, Marrakech.

Banerjee, Satanjeev and Ted Pedersen. 2003. Extended gloss overlap as a measure of semantic relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 805–810, Acapulco.

Bansal, Mohit, John DeNero, and Dekang Lin. 2012. Unsupervised translation sense clustering. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782, Montréal.

Basile, Pierpaolo, Annalina Caputo, and Giovanni Semeraro. 2014. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin.

Belinkov, Yonatan and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72. https://doi.org/10.1162/tacl_a_00254

Bennett, Andrew, Timothy Baldwin, Jey Han Lau, Diana McCarthy, and Francis Bond. 2016. LexSemTm: A semantic data set based on all-words unsupervised sense distribution learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1524, Berlin. https://doi.org/10.18653/v1/P16-1143

Bevilacqua, Michele 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. https://doi.org/10.18653/v1/2020.acl-main.255

Blevins, Terra 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. https://doi.org/10.18653/v1/2020.acl-main.95

Bond, Francis and Ryan Foster. 2013. Linking and extending an open multilingual WordNet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Camacho-Collados, Jose and Roberto Navigli. 2017. BabelDomains: Large-scale domain labeling of lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 223–228, Valencia. https://doi.org/10.18653/v1/E17-2036

Camacho-Collados, Jose and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, 63:743–788. https://doi.org/10.1613/jair.1.11259

Chaplot, Devendra Singh and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 5062–5069, New Orleans, LA.

Chronis, Gabriella and Katrin Erk. 2020. When is a bishop not like a rook? When it's like a rabbi! Multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. https://doi.org/10.18653/v1/2020.conll-1.17

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence. https://doi.org/10.18653/v1/W19-4828

Conneau, Alexis, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne. https://doi.org/10.18653/v1/P18-1198

de Vries, Wietse, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about BERT's layers? A closer look at the NLP pipeline in monolingual and multilingual models. *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4339–4350, Online.

Delli Bovi, Claudio, Jose Camacho-Collados, Alessandro Raganato, and Roberto Navigli. 2017. EuroSense: Automatic harvesting of multilingual sense annotations from parallel text. In *Proceedings of ACL*, volume 2, pages 594–600, Vancouver. https://doi.org/10.18653/v1/P17-2094

Devlin, Jacob, 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, MN.

Edmonds, Philip and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse.

Ethayarajh, Kawin. 2019. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong. https://doi.org/10.18653/v1/D19-1006

Ettinger, Allyson. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48. https://doi.org/10.1162/tacl_a_00298

Federmeier, Kara D., Jessica B. Segal, Tania Lombrozo, and Marta Kutas. 2000. Brain

responses to nouns, verbs and class-ambiguous words in context. *Brain*, 123(12):2552–2566. https://doi.org/10.1093/brain/123.12.2552, PubMed: 11099456

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*, MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/7287.001.0001

Flekova, Lucie and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2029–2041, Berlin. https://doi.org/10.18653/v1/P16-1191

Gale, William A., Kenneth Church, and David Yarowsky. 1992a. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 249–256, Newark, DE. https://doi.org/10.3115/981967.981999

Gale, William A., Kenneth W. Church, and David Yarowsky. 1992b. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5):415–439. https://doi.org/10.1007/BF00136984

Goldberg, Yoav. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Haber, Janosch and Massimo Poesio. 2020. Word sense distance in human similarity judgements and contextualised word embeddings. In *Proceedings of the Probability and Meaning Conference (PaM 2020)*, pages 128–145, Barcelona.

Hardt, Moritz, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3315–3323, Curran Associates, Inc.

Hauer, Bradley and Grzegorz Kondrak. 2020. One homonym per translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7895–7902, New York, NY. https://doi.org/10.1609/aaai.v34i05.6296

Hewitt, John 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, MN.

Hovy, Eduard H., Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27. https://doi.org/10.1016/j.artint.2012.10.002

Huang, Luyao, 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 3507–3512, Hong Kong. https://doi.org/10.18653/v1/D19-1355

Iacobacci, Ignacio, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin. https://doi.org/10.18653/v1/P16-1085

Ide, Nancy, Collin Baker, Christiane Fellbaum, Charles Fillmore, and Rebecca Passonneau. 2008. MASC: The manually annotated sub-corpus of American English. *6th International Conference on Language Resources and Evaluation, LREC 2008*, pages 2455–2460, Miyazaki.

Ilievski, Filip, Piek Vossen, and Stefan Schlobach. 2018. Systematic study of long tail phenomena in entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 664–674, Santa Fe, NM.

Jawahar, Ganesh, 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. https://doi.org/10.18653/v1/P19-1356

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia. https://doi.org/10.18653/v1/E17-2068

Jurgens, David and Mohammad Taher Pilehvar. 2015. Reserating the awesometastic: An automatic extension of the WordNet taxonomy for novel terms. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1459–1465, Denver, CO. https://doi.org/10.3115/v1/N15-1169

Kumar, Sawan, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence. https://doi.org/10.18653/v1/P19-1568

Kuncoro, Adhiguna, 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. https://doi.org/10.18653/v1/P18-1132

Lacerra, Caterina, Michele Bevilacqua, Tommaso Pasini, and Roberto Navigli. 2020. CSI: A coarse sense inventory for 85% word sense disambiguation. In *Proceedings of the 34th Conference on Artificial Intelligence*, pages 8123–8130, New York, NY. https://doi.org/10.1609/aaai.v34i05.6324

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, OpenReview.net.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual Conference on Systems Documentation*, pages 24–26, Toronto. https://doi.org/10.1145/318723.318728

Levine, Yoav, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. https://doi.org/10.18653/v1/2020.acl-main.423

Ling, Xiao, Sameer Singh, and Daniel S. Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328. https://doi.org/10.1162/tacl_a_00141

Linzen, Tal, 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. https://doi.org/10.1162/tacl_a_00115

Liu, Nelson F., Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual 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 1073–1094, Minneapolis, MN.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loureiro, Daniel and Jose Camacho-Collados. 2020. Don't neglect the obvious: On the role of unambiguous words in word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3514–3520, Online. https://doi.org/10.18653/v1/2020.emnlp-main.283

Loureiro, Daniel and Alípio Jorge. 2019a. Language modeling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence. https://doi.org/10.18653/v1/P19-1569

Loureiro, Daniel and Alípio Jorge. 2019b. LIAAD at SemDeep-5 challenge: Word-in-context (WiC). In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 1–5, Macau.

Luo, Fuli, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pages 2473–2482, Melbourne. https://doi.org/10.18653/v1/P18-1230,

Mallery, J. C. 1988. *Thinking about foreign policy: Finding an appropriate role for artificial intelligence computers*. Ph.D. Thesis, M.I.T. Political Science Department, Cambridge, MA.

Maru, Marco, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: Challenging supervised word sense disambiguation with lexical-semantic combinations. 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 3534–3540, Hong Kong. https://doi.org/10.18653/v1/D19-1359

McCarthy, Diana, Marianna Apidianaki, and Katrin Erk. 2016. Word sense clustering and clusterability. *Computational Linguistics*, 42(2):245–275. https://doi.org/10.1162/COLI_a_00247

McCrae, John Philip, Alexandre Rademaker, Ewa Rudnicka, and Francis Bond. 2020. English WordNet 2020: Improving and extending a WordNet for English using an open-source methodology. In *Proceedings of the LREC 2020 Workshop on Multimodal Wordnets (MMW2020)*, pages 14–19, Marseille.

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Groberger. 2018. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861. https://doi.org/10.21105/joss.00861

Melamud, Oren, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin. https://doi.org/10.18653/v1/K16-1006

Mickus, Timothee, Denis Paperno, Mathieu Constant, and Kees van Deemter. 2020. What do you mean, BERT? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 279–290, New York, NY.

Mihalcea, Rada, Timothy Chklovski, and Adam Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28, Lisbon.

Mihalcea, Rada and Andras Csomai. 2007. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference On Information And Knowledge Management*, pages 233–242, Lisbon. https://doi.org/10.1145/1321440.1321475

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

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, Lake Tahoe, NV.

Miller, George A., Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*, pages 303–308, Plainsboro, NJ. https://doi.org/10.3115/1075671.1075742

Moro, Andrea and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *Proceedings of SemEval-2015*. Denver, CO. https://doi.org/10.18653/v1/S15-2049

Moro, Andrea, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244. https://doi.org/10.1162/tacl_a_00179

Nair, Sathvik, Mahesh Srinivasan, and Stephan Meylan. 2020. Contextualized word embeddings encode aspects of human-like word sense knowledge. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 129–141, Online.

Navigli, Roberto. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69. https://doi.org/10.1145/1459352.1459355

Navigli, Roberto, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual word sense disambiguation. In *Proceedings of SemEval 2013*, pages 222–231, Atlanta, GA.

Navigli, Roberto and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250. https://doi.org/10.1016/j.artint.2012.07.001

Palmer, Martha, Hoa Dang, and Christiane Fellbaum. 2007. Making fine-grained and

coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 13(2):137–163. https://doi.org/10.1017/S135132490500402X

Pasini, Tommaso. 2020. The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4936–4942. https://doi.org/10.24963/ijcai.2020/687

Pasini, Tommaso and Jose Camacho-Collados. 2020. A short survey on sense-annotated corpora. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5759–5765, Marseille.

Pasini, Tommaso and Roberto Navigli. 2018. Two knowledge-based methods for high-performance sense distribution learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5374–5381, New Orleans, LA.

Pasini, Tommaso and Roberto Navigli. 2020. Train-o-matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279:103215. https://doi.org/10.1016/j.artint.2019.103215

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, LA. https://doi.org/10.18653/v1/N18-1202

Peters, Matthew, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels. https://doi.org/10.18653/v1/D18-1179

Peters, Matthew E., Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong. https://doi.org/10.18653/v1/D19-1005, PubMed: 31383442

Peters, Matthew E., Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14. https://doi.org/10.18653/v1/W19-4302

Peterson, Daniel and Martha Palmer. 2018. Bayesian verb sense clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 5398–5405.

Pilehvar, Mohammad Taher and Jose Camacho-Collados. 2019. WiC: the word-in-context data set for evaluating context-sensitive meaning 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 1267–1273, Minneapolis, MN.

Pilehvar, Mohammad Taher, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a seamless integration of word senses into downstream NLP applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1857–1869, Vancouver. https://doi.org/10.18653/v1/P17-1170

Pilehvar, Mohammad Taher and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881. https://doi.org/10.1162/COLI_a_00202

Postma, Marten, Ruben Izquierdo, Eneko Agirre, German Rigau, and Piek Vossen. 2016. Addressing the MFS bias in WSD systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1695–1700, Portorož.

Postma, Marten, Ruben Izquierdo Bevia, and Piek Vossen. 2016. More is not always better: balancing sense distributions for all-words word sense disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3496–3506, Osaka.

Pradhan, Sameer, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*, pages 87–92. https://doi.org/10.3115/1621474.1621490

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Raganato, Alessandro, Jose Camacho-Collados, and Roberto Navigli. 2017. 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. https://doi.org/10.18653/v1/E17-1010

Raganato, Alessandro, Claudio Delli Bovi, and Roberto Navigli. 2017. 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. https://doi.org/10.18653/v1/D17-1120

Raganato, Alessandro, Tommaso Pasini, Jose Camacho-Collados, and Taher Mohammad Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. *EMNLP 2020*, pages 7193–7206. https://doi.org/10.18653/v1/2020.emnlp-main.584

Reif, Emily, 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*, pages 8592–8600.

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

Resnik, Philip and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):113–133. https://doi.org/10.1017/S1351324999002211

Rodd, Jennifer M. 2020. Settling into semantic space: An ambiguity-focused account of word-meaning access. *Perspectives on Psychological Science*, 15(2):411–427. https://doi.org/10.1177/1745691619885860, PubMed: 31961780

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866. https://doi.org/10.1162/tacl_a_00349

Rüd, Stefan, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 965–975, Portland, OR.

Saphra, Naomi 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, MN.

Scarlini, Bianca, Tommaso Pasini, and Roberto Navigli. 2019. Just "OneSeC" for producing multilingual sense-annotated data. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 699–709, Florence.

Scarlini, Bianca, Tommaso Pasini, and Roberto Navigli. 2020a. SensEmBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *Proceedings of the Thirty-Fourth Conference on Artificial Intelligence*, pages 8758–8765. https://doi.org/10.1609/aaai.v34i05.6402

Scarlini, Bianca, Tommaso Pasini, and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. https://doi.org/10.18653/v1/2020.emnlp-main.285

Schütze, Hinrich. 1993. Word space. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 895–902, Morgan-Kaufmann.

Scozzafava, Federico, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–46, Online. https://doi.org/10.18653/v1/2020.acl-demos.6

Severyn, Aliaksei, Massimo Nicosia, and Alessandro Moschitti. 2013. Learning semantic textual similarity with structural representations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 714–718, Sofia.

Shwartz, Vered and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419. https://doi.org/10.1162/tacl_a_00277

Soler, Aina Garí, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2019. A comparison of context-sensitive models for lexical substitution. In *Proceedings of the 13th International Conference on Computational Semantics-Long Papers*, pages 271–282.

Taghipour, Kaveh and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing. https://doi.org/10.18653/v1/K15-1037

Taylor, Wilson L. 1953. "Cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433. https://doi.org/10.1177/107769905303000401

Tenney, Ian, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence. https://doi.org/10.18653/v1/P19-1452

Tenney, Ian, 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. In *Proceeding of the 7th International Conference on Learning Representations (ICLR)*.

Usbeck, Ricardo, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, and others. 2015. GERBIL: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1133–1143.

van Schijndel, Marten, 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 5830–5836, Hong Kong. https://doi.org/10.18653/v1/D19-1592

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Vial, Loïc, Benjamin Lecouteux, and Didier Schwab. 2018. UFSAC: Unification of sense annotated corpora and tools. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki.

Vial, Loïc, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic state-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Voita, Elena, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the Transformer: A study with machine translation and language modeling objectives. 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 4396–4406, Hong Kong.

Wang, Alex, 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors. *Advances in Neural Information Processing Systems*, volume 32, Curran Associates, Inc.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels. https://www.aclweb.org/anthology/W18-5446.

Wiedemann, Gregor, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does

BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, and others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online.

Yaghoobzadeh, Yadollah, Katharina Kann, T. J. Hazen, Eneko Agirre, and Hinrich Schütze. 2019. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5740–5753, Florence. https://doi.org/10.18653/v1/P19-1574

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763, Curran Associates, Inc.

Yenicelik, David, Florian Schmidt, and Yannic Kilcher. 2020. How does BERT capture semantics? A closer look at polysemous words. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 156–162, Online. https://doi.org/10.18653/v1/2020.blackboxnlp-1.15

Yuan, Dayu, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. 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.

Zhong, Zhi and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL System Demonstrations*, pages 78–83, Uppsala.

Zhou, Wangchunshu, 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. https://doi.org/10.18653/v1/P19-1328

Zipf, George K. 1949. *Human behavior and the Principle of Least-Effort*, Addison-Wesley, Cambridge, MA.