SAPIENZA
UNIVERSITÀ DI ROMA

DEPARTMENT OF COMPUTER, CONTROL AND MANAGEMENT
ENGINEERING ANTONIO RUBERTI

Word Sense Disambiguation: a Semantic Journey from Sense Granularity to Large Language Models

THESIS

MASTER PROGRAM IN ARTIFICIAL INTELLIGENCE AND ROBOTICS

Supervisor:
Prof. Roberto Navigli

Student:
Leonardo Lavalle

Academic Year 2023/2024

Contents

1 Introduction

My semantic journey through Artificial Intelligence (AI) and in particular through Natural Language Processing (NLP) began last September (2023) when I started to collaborate with SapienzaNLP group, headed by professor Roberto Navigli.

Semantics is the study of meaning in language, focusing on how words, phrases, and sentences represent and convey meaning within context. Human language is ambiguous, allowing many words to be interpreted in various ways depending on their context. For example in Figure 1 we can observe how the word *mouse* may have three different meanings which depends on the context. In this particular case the most appropriate meaning is "technological" one.

Humans possess an innate ability to disambiguate word meanings effortlessly and intuitively. We interpret words based on context, making it hard to grasp why this task is so complex and challenging for machines. If we think about it for a moment, we constantly use our capability to understand semantics in our daily lives without even realizing it. This skill largely influences our reading comprehension, our understanding of others, and the decisions we make based on our interpretations. Understanding semantics helps us distinguish between literal and figurative language, detect sarcasm or irony, and identify author's tone and intent. In communication, it is crucial for accurately interpreting meanings, affecting our responses and interactions. Misunderstanding semantics can lead to conflict, while a clear grasp fosters mutual understanding and effective problem-solving. Semantics also influence decisions based on our interpretations, such as legal choices from contract wording or political opinions from rhetoric. Essentially, semantics is foundational to our cognitive processes, shaping comprehension, communication, and decision-making.

The primary goal of developing AI solutions is often to replicate human intelligence and behavior. When we read or listen, our brains actively decode semantics to understand the message, infer implicit meanings, and integrate new information with our existing knowledge. This capability is precisely what we aim to achieve with Natural Language Understanding (NLU) systems that can effectively understand lexical semantics. This remains a formidable challenge for machines due to language's inherent ambiguity and the nuanced human perceptions of word and sentence meanings. While advancements have facilitated syntactic text processing across numerous languages, semantic analysis, particularly for arbitrary languages, continues to pose significant difficulties. Neural networks have enhanced various aspects of NLP, but it is the advent of *word embeddings* that has profoundly transformed lexical-semantic tasks, greatly advancing the analysis of language meaning.

One of these semantic-related tasks is Word Sense Disambiguation (WSD), i.e., the computational process of determining the meaning of words in context. This fundamental task in NLP and AI traces its origins to (Weaver, 1949), who identified the

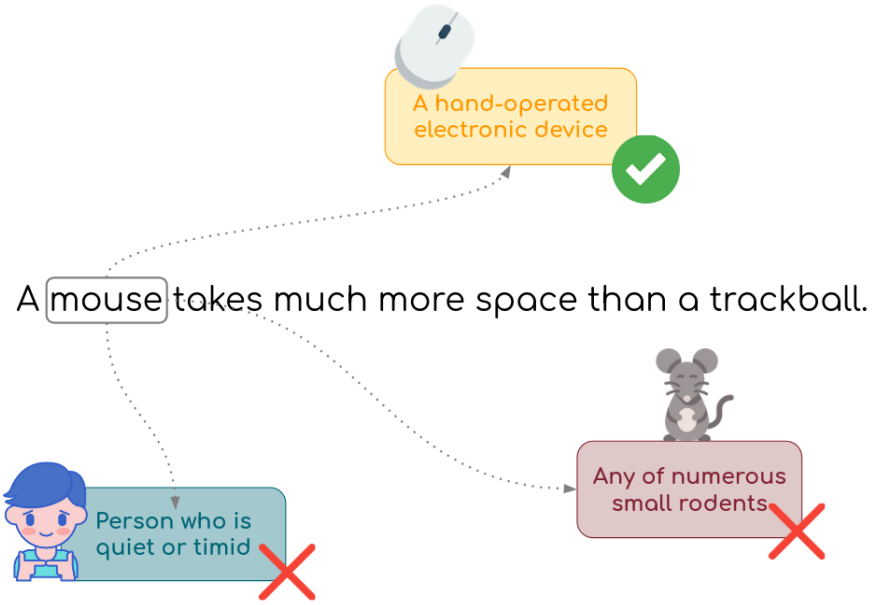


Figure 1: Disambiguation example of word *mouse*.

issue of polysemous words within the realm of machine translation. Despite decades of research, word polysemy remains a significant and widespread challenge in NLP today. Granularity of sense inventories and incopleteness of knowledge resources are just a few of the many problems that WSD has and may encounter. The manual creation of these resources is an expensive and time-consuming effort (Ng and Lee, 1996), but it is necessary to perform satisfactorily the task. In the past, the limited progress in WSD was primarily due to the scarcity of extensive machine-readable knowledge. This is a fundamental problem which pervades the field of WSD, and is called the *knowledge acquisition bottleneck* (Gale et al., 1992b). However, a significant turning point occurred in the 1980s with the introduction of large-scale lexical resources. These resources enabled the development of automatic methods for knowledge extraction, significantly advancing the field (Wilks et al., 1990). Fortunately, we can now exploit numerous resources. Among these are WordNet (Miller et al., 2008), a large lexical database of English that groups word meanings into synsets (concepts) and interlinks them through various semantic relations; BabelNet (Navigli and Ponzetto, 2010), a multilingual semantic network that integrates concepts and lexical items from multiple sources, including WordNet and Wikipedia; and SemCor (Miller et al., 1993), an English corpus of manually annotated texts with part-of-speech tags and WordNet senses, commonly used as a training set for WSD systems. Even with more data available than in previous years, the WSD task remains unsolved. Researchers are still struggling with various inherent difficulties and persistent questions (see Section 2.5). For instance, should meaning representations be implicit, explicit, or a combination of both? Implicit representations involve learning and utilizing word embeddings derived naturally from the training of neural networks. Explicit representations, on the other

hand, utilize lexical-semantic knowledge resources to link vectors with clearly defined concept entries, such as WordNet synsets. From this distinction we can already identify the two main families of WSD approaches: **data-driven** and **knowledge-based** (Section 2.4). Recent developments indicate that implicit (data-driven) and explicit (knowledge-based) methods can coexist, with vector representations for senses and synsets embedded in the same vector space as word representations (Camacho-Collados et al., 2016).

For the past year, I have been proud to be part of a group of researchers dedicated to exploring and evaluating the strengths and weaknesses of current NLU methods, with a particular focus on Word Sense Disambiguation (WSD). These my recent contributions:

- My first one is on the *Analyzing Homonymy Disambiguation Capabilities of Pretrained Language Models* paper (Proietti et al., 2024), presented at the **LREC-COLING** conference held in Turin this May (2024). The main contribution is the development of a new resource designed to overcome the limitations of traditional sense inventories based on WordNet, which often distinguish meticulously between subtle nuances of word meanings, resulting in unnecessary complexity (Section 3).

- I co-author a scientific paper currently under revision for the **EMNLP** 2024 conference, titled *LLMantics: A Novel Benchmark for Measuring Lexical Semantic Capabilities of Instruction-Tuned Large Language Models* (Martelli et al., 2024) Our aim is to investigate the semantic abilities of instruction-tuned Large Language Models (LLMs) and to determine whether they are as powerful as claimed. In Section 4, we address this question.

In the following sections we first analyze more in details Word Sense Disambiguation task (Section 2), by exploring the current datasets and State-of-the-Art (SOTA) models, reporting how the transformers revolution had a big impact on WSD and more importantly which are the problems yet to be defeated. We then explode the sense granularity problem in WSD and precisely explain how we tried to deal with it (Section 3). To get an immediate understanding of what is the difference between homonymous and polysemous senses of a word see the example of the word *plane* in Figure 2. Polysemous senses are clustered to form homonyms cluster (three in this case).

Subsequently, the LLMantics evaluation benchmark is showcased in all its semantic evaluation settings (Section 4). To establish a connection between the two studies, I decided to test some instruction-tuned LLMs on another semantic task: the Homonymy Disambiguation (HD) task. In this task, the system must predict to which homonym cluster the word in context belongs. Essentially, we compare the performance of these Large Language Models in both the Word Sense Disambiguation

Figure 2: Senses of the noun *plane* grouped by its three homonyms. Source: (Proietti et al., 2024)

(WSD) and Homonymy Disambiguation (HD) tasks, and compare the results with those achieved in *Analyzing Homonymy Disambiguation Capabilities of Pretrained Language Models* paper (Proietti et al., 2024) (Section 5).

2 Word Sense Disambiguation

WSD is the task of automatically associating senses with words in context and it is a difficult and yet not solved NLP and AI problem. In fact, WSD has been described as an AI-complete problem (Mallery, 1988), that is, by analogy to NP-completeness in complexity theory, a problem whose difficulty is equivalent to solve the most fundamental problems in AI, such as the Turing Test (Turing, 1950).

2.1 Applications

Why is important to have intelligent systems which are able to disambiguate word meanings with a certain precision? Does all the research orbiting WSD have any utility in real-world applications? Yes, the task can benefit many downstream applications (Proietti et al., 2024), including but not limited to Information Retrieval (IR) (Zhong and Ng, 2012; Stokoe et al., 2003) and Machine Translation (MT) (Neale et al., 2016; Carpuat and Wu, 2007; Vickrey et al., 2005; Pu et al., 2018). For an IR system to function effectively, it must utilize explicit semantic techniques to filter out irrelevant documents. Accurate disambiguation of both the documents and the query terms significantly enhances the search engine's performance by ensuring that the retrieved documents are contextually relevant to the user's query. In the context of MT, the contribution WSD systems can make is particularly evident. Translation models inherently require the capability to comprehend semantics and disambiguate word meanings. However, challenges arise when dealing with infrequent or rare word senses. Although this poses difficulties for WSD systems as well, specialized disambiguation models are better equipped to manage these extreme cases, thereby enhancing the overall accuracy and quality of translations. MT and WSD tasks are so semantically intertwined that exist cases where neural MT systems are used to enhance WSD performance (Luan et al., 2020). In this paper the authors leverage translations in BabelNet to refine the output of any arbitrary WSD system by comparing the translation of the output senses with the target's translations provided by an neural MT model. Moreover, there exist cases where disambiguation analysis is employed to verify consistency and quality in translation. In (Campolungo et al., 2022) they studied semantic biases, specifically addressing how MT models handle lexical ambiguity. The primary goal of the paper was to detect biases where words are disproportionately translated into their more frequent meanings instead of contextually appropriate ones. This underscores the concept that Machine Translation and Word Sense Disambiguation systems are intrinsically interconnected. It is anticipated that their interdependence will grow, with each increasingly relying on the other to achieve optimal performance in the future.

2.2 Task description

We can view WSD as a classification task (Navigli, 2009): word senses are the classes, and an automatic classification method is used to assign each occurrence of a word to one (or more classes). This assignment is based on three main ingredients:

- *sense inventory*: the most debated and fundamental component in a canonical WSD system. It is what partitions the range of meaning of a word into its senses and decrees the granularity degree of a given word. It is quite clear how this delicate choice has an impact on the success of all the disambiguation pipeline. Word senses cannot be easily discretized into a finite set of distinct entries due to the inherently fluid and interpretive nature of language. Additionally, it is often challenging to delineate where one sense of a word ends and another begins. For example, consider the example of noun *knife* in Figure 3. Should we add a further sense to the inventory for "a cutting blade forming part of a machine" or does the first sense comprise this sense? This kind of uncertainties lead to subjective and different choices generating different dictionaries and resources. This is still an open problem in WSD and many works are attempting to move away from the traditional paradigm of discrete sense inventories (we discuss the problem in Section 2.5), exploring alternative methods that do not rely on predefined sets of word senses Bevilacqua et al. (2020); Pilehvar and Camacho-Collados (2019); McCarthy and Navigli (2007a).

knife *n.* **1.** a cutting tool composed of a blade with a sharp point and a handle. **2.** an instrument with a handle and blade with a sharp point used as a weapon.

Figure 3: Example of two senses for the noun *knife*.

- *knowledge resources*: provide data which are essential to associate senses with words. Among them, the ones that are the most widely used in the field are WordNet (Miller et al., 2008) and SemCor (Miller et al., 1993). All the advancements in WSD achieved until now would not have been possible without the huge and manual effort provided by the WordNet inventory. Created and maintained at Princeton University, it encodes concepts (*synsets*) in terms of sets of synonyms. To give an idea, below is represented the synset of *automobile* concept (superscript is the word's sense identifier and subscript is the part-of-speech tag).

$$\{car_n^1, auto_n^1, automobile_n^1, machine_n^4, motorcar_n^1\}$$

  All semantic-related resorces derive from WordNet. Indeed, SemCor sense-annotated corpora have been manually annotated with part-of-speech tags, lemmas, and word senses from the WordNet inventory. More details about resources and WSD evaluation benchmarks in Section 2.3.

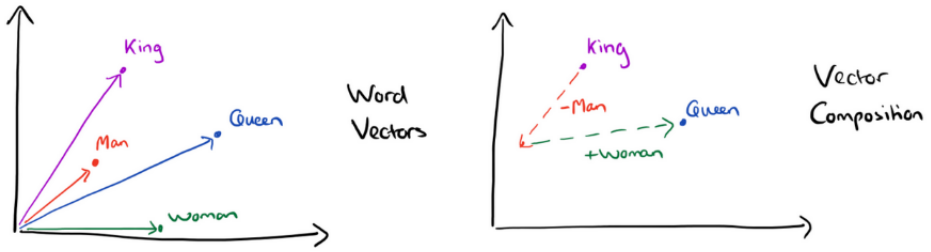


Figure 4: Word2vec example explaining why $King - Man + Woman = Queen$.

- *context representation*: another crucial aspect is how to make text, and in particular the word in context to disambiguate, a suitable input to be ingested by an automatic WSD method. In other words, how can we embed text to effectively capture and encode its semantics? One of the pioneering word embedding models was Word2vec (Mikolov et al., 2013) which revolutionized the field of lexical semantics and the impact of which is still being felt today. We all remember the example depicted in Figure 4 where for the first time mathematical operations between words became possible.
  Static word embeddings, such as those generated by Word2vec, are limited by their context insensitivity. These embeddings assign the same representation to a word regardless of its context, failing to account for the fact that different contexts can invoke distinct meanings of the same word, which may be semantically unrelated. We do not need word embeddings but *sense embeddings*. We want $plane_n^2$ and $plane_n^3$ (see Figure 2) to have different embeddings. More recent works, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), succeded in addressing this limitation by employing language modeling training objectives (e.g. Masked Language Modeling (MLM) and Next Sentence Prediction (NSP)) that generate context-sensitive representations. These models compute word meanings embeddings that closely align with human word sense knowledge (Nair et al., 2020). Contextualized embeddings, such as those produced by Transformer-based models like BERT, have seamlessly integrated into various NLP models with minimal modifications, resulting in unprecedented performance improvements. This significant boost, which many have called *transformers revolution*, has been observed across a wide range of NLP applications, including WSD (see Section 2.4.2 for more details).

The final step in performing Word Sense Disambiguation involves selecting the appropriate algorithm to resolve word ambiguity. In the following sections, we first provide a detailed examination of the WSD resources and evaluation metrics utilized by the research community (Section 2.3). Subsequently, we present a comprehensive review of the latest WSD approaches, including historically significant methods and state-of-the-art (SotA) models (Section 2.4).

2.3 Resources

WSD is a knowledge-intensive task that requires two distinct types of data: (i) *sense inventories*, which are reference computational lexicons that enumerate possible meanings of words; and (ii) *annotated corpora*, where a subset of words is tagged with one or more possible meanings derived from the given inventory. Before delving into the details of my two contributions to WSD (Section 3 and Section 4), it is essential to provide an overview of the most widely used sense inventories and annotated corpora employed for training and testing WSD systems by the research community.

2.3.1 Sense Inventories

Sense inventories enumerate the set of possible senses for a given lexeme. The most popular ones are:

- **WordNet**: a comprehensive, manually-curated lexicographic database of English that serves as the standard inventory for WSD (Miller et al., 2008). It is organized into a graph with nodes as synsets (groups of contextual synonyms) linked by lexical-semantic relations such as *hypernymy* (is-a) and *meronymy* (part-of). WordNet provides glosses and usage examples for each synset. The most recent version used in WSD is 3.0, containing 117.659 synsets. The 2020 extension introduced approximately 3.000 new synsets, including slang and neologisms.

- **BabelNet**: a multilingual dictionary and semantic network created by semi-automatically mapping various resources, including WordNet and Wikipedia. BabelNet features multilingual synsets (groups of synonyms in multiple languages) linked by semantic relations. The latest version, 5.3 (2023), covers 600 languages and includes more than 23 million synsets.

In Section 2.5, both inventories suffer from the well-known *fine-granularity problem*, where it is often difficult even for humans to discriminate between the senses of a given word. Additionally, these sense inventories operate under the assumption that word meanings can be fully enumerated in a finite list, which significantly limits the expressive power of WSD systems.

2.3.2 Sense-Annotated Data

We now describe the standard benchmarks used in WSD, both multilingual and English. However, the focus will be primarily on the English benchmarks, as my work has exclusively utilized English-only datasets.

Training data SemCor (Miller et al., 1993) is the largest manually annotated dataset for WSD, containing 200.000 sense annotations based on the WordNet sense inventory. Despite its extensive effort, it only covers 22% of the nearly 118.000 WordNet synsets. Additionally, as a subset of the English Brown Corpus from the 1960s, SemCor's sense distribution differs from contemporary texts, lacking many modern meanings such as the "computer" meaning of *mouse*. An attempt to improve the annotation coverage has been made by Bevilacqua and Navigli (2020), which incorporated the English Princeton WordNet Gloss Corpus (WNG[1]) as supplementary data, being able to achieve higher performances. While ample English training data exists, obtaining large-scale, hand-labeled data for other languages remains challenging.

Testing data Evaluation in WSD is typically conducted using manually annotated datasets from the Senseval and SemEval evaluation campaigns. These competitions standardize the evaluation of semantic processing systems by providing shared tasks, datasets, and metrics, allowing for consistent comparisons of different approaches. They drive advancements in NLP by encouraging the development and refinement of systems based on competitive benchmarks, highlighting strengths and weaknesses, and guiding future research. Additionally, Senseval and SemEval foster a collaborative research environment, bringing together researchers worldwide to address common challenges and share findings, thus enhancing progress in NLP and semantic processing. They serve as the fuel that keeps the fire of research burning.
Raganato et al. (2017) proposed the Unified Evaluation Framework for WSD (English-only) which combines together five gold-standard datasets, all from Senseval and SemEval competitions: Senseval-2 (Edmonds and Cotton, 2001), Senseval-3 (Snyder and Palmer, 2004), SemEval-2007 Task 17 (Pradhan et al., 2007), SemEval-2013 Task 12 (Navigli et al., 2013a) and SemEval-2015 Task 13 (Moro and Navigli, 2015). According to Figure 5, SemCor is used as the training set, SemEval-2007 as the validation set, and the concatenation of all remaining datasets constitutes the evaluation set, namely the ALL test set.
This framework represents a milestone for WSD evaluation and research, as it standardizes the evaluation process in English WSD using the WordNet sense inventory. By facilitating the comparison of systems in a general domain, it aids the field in developing increasingly better-performing models. In Section 2.5.2, a more recent test set is discussed (ALL_{NEW}), that has been proposed by Maru et al. (2022), amending the original ALL dataset to correct its imperfections. In the LLMantics benchmark (Martelli et al., 2024), we use ALL as the testing set, while in Proietti et al. (2024), ALL_{NEW} is one of the four datasets included in the train/test/val split (more details in Section 3).

[1] `https://wordnetcode.princeton.edu/glosstag.shtml`

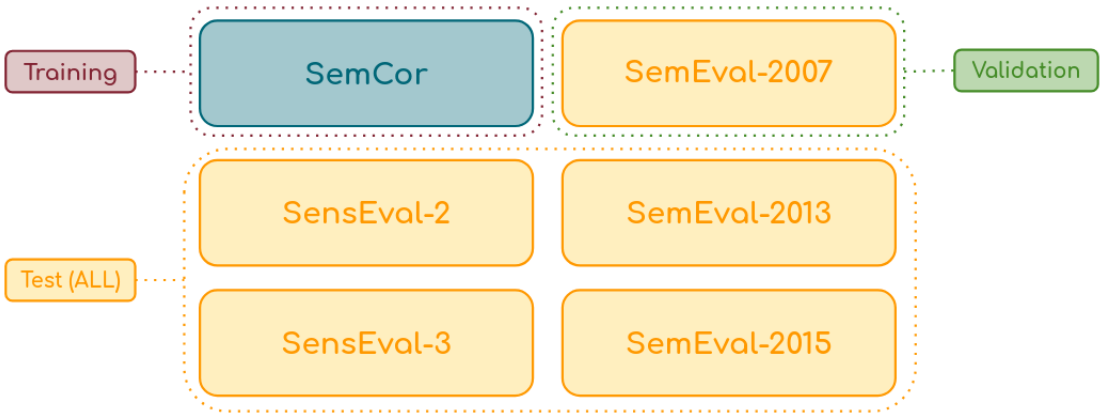


Figure 5: Split of Unified Evaluation framework (Raganato et al., 2017).

For non-English languages, WSD evaluation datasets have historically received less attention, often being annotated with varied and outdated inventories. Only recently has a comprehensive benchmark been introduced to standardize evaluation in this domain as well Pasini et al. (2021). The XL-WSD benchmark extends the English evaluation framework of Raganato et al. (2017) and introduces test data for 18 languages, facilitating, for the first time, a large-scale monolingual and multilingual evaluation of WSD models. Research in multilingual WSD is now thriving, with significant efforts being made to bridge the gap between English and non-English language evaluations.
It is crucial to emphasize the importance of creating robust benchmarks and encouraging benchmark creators to design more solid and transparent evaluation frameworks. This practice helps to avoid the risk of claiming superhuman performance when the reality is different (Tedeschi et al., 2023). Establishing reliable benchmarks is essential for the accurate evaluation and comparison of models, and is vital for the future progress of WSD. The same importance applies to the evaluation of Large Language Models. Martelli et al. (2024), aims to address a gap in assessing LLMs' capabilities, particularly concerning the semantics of words. It seeks to determine whether LLMs truly understand text (Section 4).

Evaluation measures How do we measure performances of WSD systems? The evaluation metrics employed in classification tasks are used here as well, namely *recall* (R), *precision* (P), and *f1-score* ($F1$). According to Navigli (2009), coverage C is defined as the percentage of items in the test set for which the system provided a sense assignment:

$$C = \frac{\#\text{provided answers}}{\#\text{answers to provide}}$$

In our WSD systems, we always provide an answer, resulting in 100% coverage. Precision (P) is defined as the percentage of correct answers provided by the system,

while recall (R) represents the number of correct answers divided by the total number of answers to be provided. Given that our coverage C is 100%, these metrics are equal, i.e., P=R=$F1$, and can be used interchangeably. In WSD literature, these metrics are often referred to as *accuracy*, effectively representing the proportion of correct predictions made by a model out of the total number of predictions.

2.4 Main approaches

In this section we review different kinds of system, ranging from those that do not require training data (knowledge-based) and those who are hungry for data (data-driven models). Both families of WSD approaches present distinct advantages and disadvantages. Data-driven methods, while demonstrating superior performance, require extensive amounts of annotated data for training. Conversely, knowledge-based methods do not necessitate training data but face the challenge of integrating and aligning diverse lexical resources, which can be complex and resource-intensive. A significant advantage of knowledge-based methods is their high sense coverage. They handle rare and unseen senses more effectively than data-driven models, which depend on finite and discrete sense inventories. If a particular meaning is absent from these inventories, data-driven models are unable to predict it. Therefore, the choice between these approaches depends on the specific needs and constraints of the application, balancing the need for data availability, performance, and sense coverage.

2.4.1 Knowldege-Based WSD

Knowledge-based approaches leverage computational lexicons, such as WordNet or BabelNet, especially their graph structure, in which synsets act as nodes and the relations between them as edges. The algorithms employ graph algorithms such as clique approximation or random walk (Agirre et al., 2014). Figure 6 shows intuitively how the method disambiguate meanings.
The algorithm utilizes the graph structure of WordNet to identify connections between the synsets within the context sentence. In this instance, the word *suite* is correctly disambiguated to refer to the "cards game" meaning rather than the "clothing" meaning. One of the most successful purely knowledge-based approaches is SyntagRank (Scozzafava et al., 2020). It is entirely graph-based and, following Agirre et al. (2014), uses a variant of the PageRank algorithm, the Personalized PageRank (PPR) (Page et al., 1999), in which the initial probability mass is distributed over a restricted set of specific nodes (i.e. the nodes representing the content words to be disambiguated in a given context). Thanks to the use of BabelNet as lexical knowledge base, SyntagRank showed itself to also be able to scale across many different language.

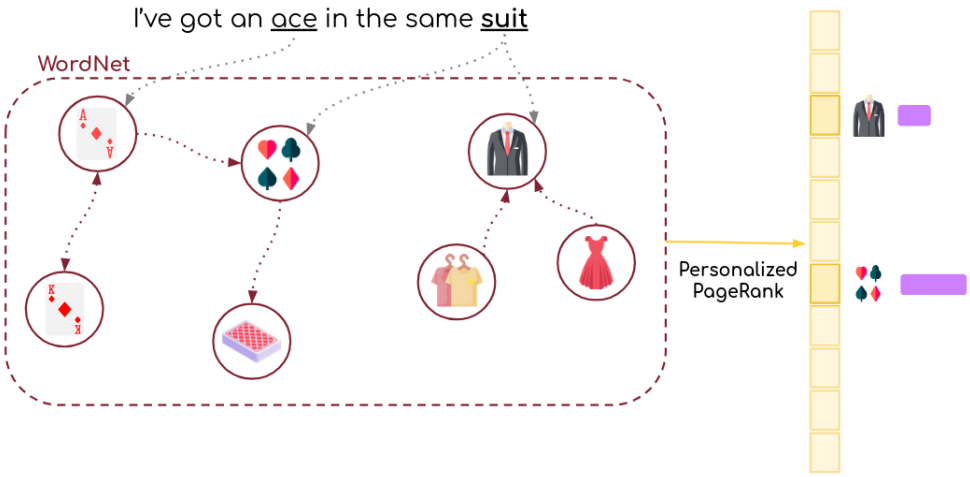


Figure 6: Random walk approach used by Agirre et al. (2014). Intuitively, the system predicts the "cards game" meaning because there are words (*ace* in this case) connected to it in WordNet.

2.4.2 Data-Driven WSD

The most successful approaches to WSD are those who are in dire need of data, in particular neural and supervised ones. Their objective is to learn a parameterized function f_θ which, given a word w in a sentence context c, has to output a sense s that belongs to a finite and discrete sense inventory. Supervised WSD has been demonstrated to significantly outperform knowledge-based approaches, even prior to the advent of pre-trained language models (Raganato et al., 2017). But performances started to have further improvements with the introduction of transformers architecture (Vaswani et al., 2023), especially with BERT-based models (Devlin et al., 2019) which are currently approaching *human-level* (is it true?) performance (Huang et al., 2019; Bevilacqua and Navigli, 2020; Barba et al., 2021a,c; Conia and Navigli, 2021; Blevins and Zettlemoyer, 2020).

Transformer-based language models have revolutionized many fields within NLP (*transformers revolution*). Models such as BERT and its derivatives have come to dominate most evaluation benchmarks, including those for WSD, due to their exceptional capability in capturing context-sensitive semantic nuances. There are two main types of Transformer-based Language Models, differentiated by their language modeling objectives. Causal models, such as GPT-3 (Brown et al., 2020b), predict the next word given the past sequence. Masked models, like BERT, predict a masked word (MLM) given its surrounding context. Benchmark results reported in Devlin et al. (2019) and Brown et al. (2020b) show that masked LMs are preferred for semantic tasks, whereas causal LMs are more suitable for language generation tasks. Masked LM objective leads to more generalized representations in intermediate layers, explaining the success of BERT-based models in semantic tasks (Voita et al., 2019) . According

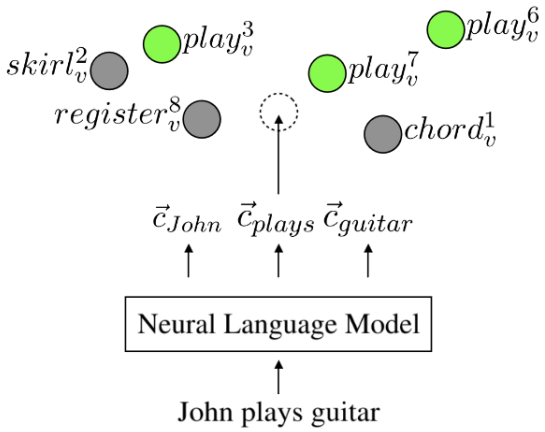


Figure 7: Illustration of a nearest neighbors approach for WSD. For simplification, we label senses as synsets. Grey nodes belong to different lemmas. Source: (Loureiro and Jorge, 2019).

to Loureiro et al. (2021) we can identify two dominant WSD approaches based on language models: (i) nearest neighbors classifiers (*feature extraction*) based on features extracted from the model, and (ii) *fine-tuning* of the model for WSD classification. In the following we describe the two strategies.

Feature extraction LMs are often used to encode the context of a target word and to generate a contextual embedding for that word. Considering their ability to capture word semantics, what if we simply use these word-in-context embeddings to choose the "*nearest*" meaning? The approach is straightforward: match the contextual embedding of the word to be disambiguated with its corresponding pre-computed sense embeddings (see Figure 7). This matching is typically performed using a simple k-Nearest Neighbors classifier (often with k = 1). This approach, as demonstrated in Loureiro and Jorge (2019), consistently outperforms previous systems that use advanced neural sequencing models.
But why is that? How can such a simple strategy have such high performances? Lots of papers have tried to analyze in more detail this BERT ability to generate distinct vector representations for the same token depending on its context. In Wiedemann et al. (2019), they compared the performance of different Contextualized Word Eembeddings (CWE) models for WSD task (including BERT, ELMo (Peters et al., 2018) and Flair NLP (Akbik et al., 2018)) and reported improvements above the current state of the art for two standard WSD benchmark datasets (SenseEval-2 and SensEval-3). They further show that the pre-trained BERT model is able to place polysemic words into distinct "sense" regions of the embedding space, while ELMo and Flair NLP do not seem to possess this ability. As we can see in Figure 8, BERT is able to correctly distinguish and place most occurrences of "spring" in distinct clusters. A fine-grained distinction can be observed for the season meaning

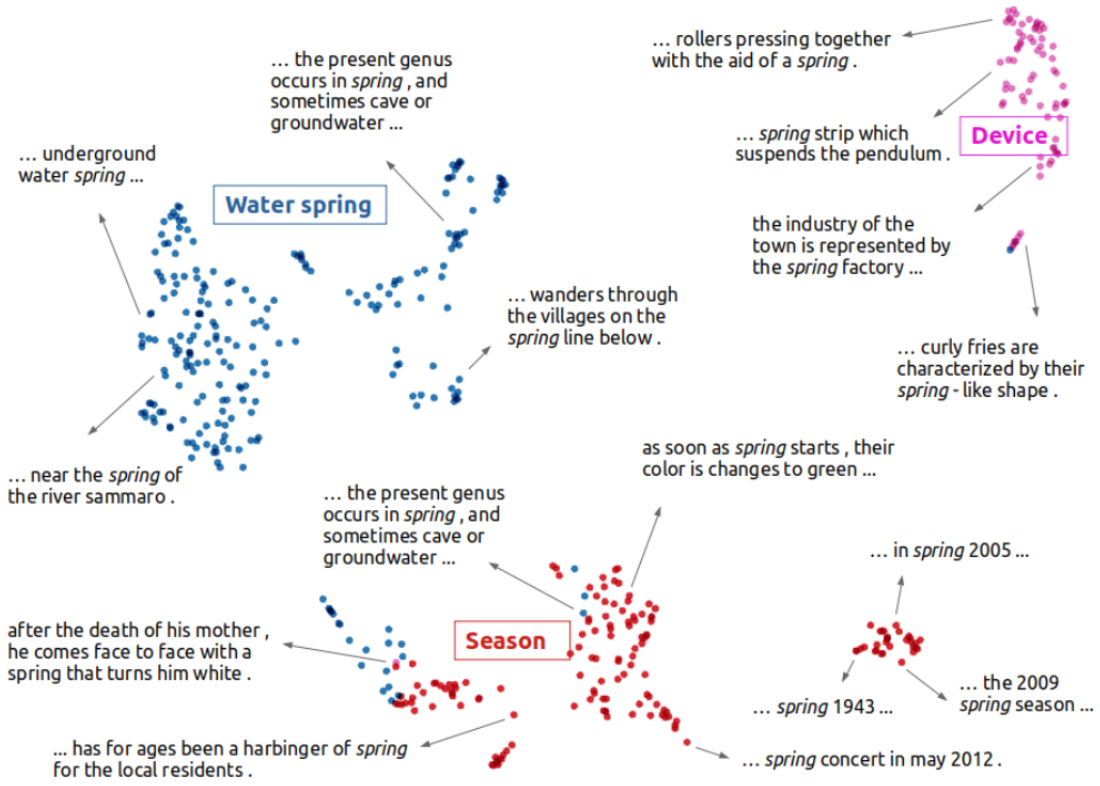


Figure 8: 2-D visualizations of contextualized representations for different occurrences of word *spring*. Using PCA for dimensionality reduction. Source: (Loureiro et al., 2021)

of spring, with a distinct cluster (on the right) denoting the spring of a specific year. We can state that BERT CWEs actually seem to encode some form of sense knowledge. A detailed study by Jawahar et al. (2019) claims that each BERT layer learns different structural aspects of natural language. Their work contributes to the growing field of neural network interpretability, demonstrating that BERT effectively captures the structural properties of the English language. An interesting paper, SenseBERT (Levine et al., 2020), tried to convey semantics to BERT at pre-training stage. Basically the model, is pre-trained to predict not only the masked words but also their WordNet supersenses. SenseBERT is capable to beat BERT in some semantics task, in particular WiC (Pilehvar and Camacho-Collados, 2019), showing that semantic signals can be similarly introduced at the pre-training stage, obtaining a lexical-semantic level Language Model.

This astonishing pre-trained LMs ability to place polysemous words into distinct "sense" regions within the embedding space, also applies to homonyms? In our paper (Proietti et al., 2024), we try to answer this question by investigating how close the representations of senses in the same homonymy cluster are, and, contextually, how far from each other those of different homonymous senses are. More details in Section 3.5.

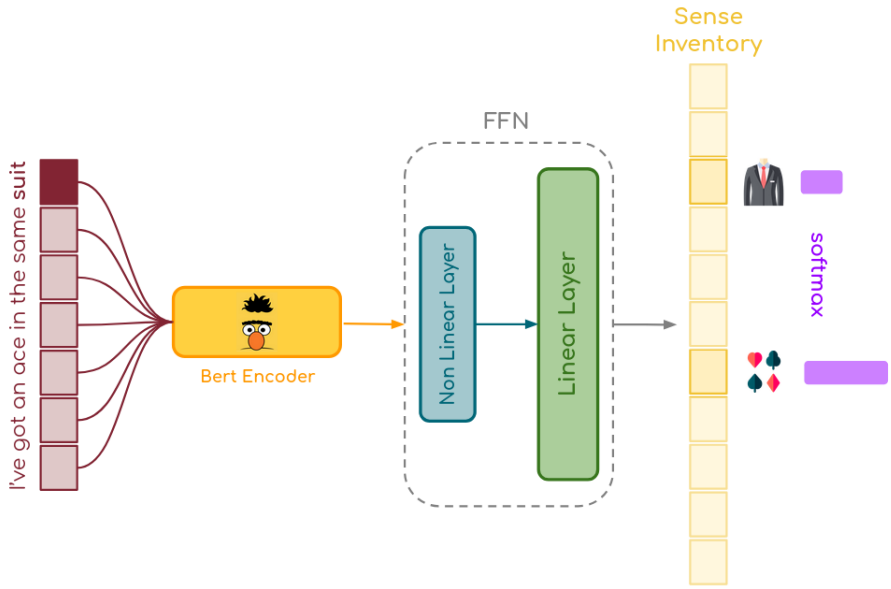


Figure 9: Model architecture used by our paper (Proietti et al., 2024). In particular the Feed Forward Network (FFN) is composed by a non-linear and then a linear layer which projects

Fine-tuning The other common data-driven approach is through fine-tuning of LMs. The most effective strategies involve concatenating a classification head on top of the pre-trained language model, and fine-tune all or part of the parameters end-to-end. This procedure adjusts model's parameters according to the objectives of the target task, e.g., the classification task in WSD. A simple yet effective architecture is the one exhibited in Hadiwinoto et al. (2019) where a Feed Forward Network (FFN) linearly projects the hidden vectors produced by BERT and applies a softmax function to output a probability distribution over all senses. It is worth mentioning that at training time, the objective function considers all the sense inventory, while at prediction time it focuses on target sense only. In this case only on the two possible meanings of "*suit*", outputting the most likely.

The same architecture idea has been employed in our paper Proietti et al. (2024) (see Figure 9), we employed a similar architectural approach to probe whether WSD systems also function as homonymy disambiguators. Specifically, inspired by Conia and Navigli (2021), the model is simply composed of BERT (large-cased and frozen), a non-linear layer and a linear classifier. Given a word w in context, we build its contextualized representation $e_w \in \mathbb{R}^{d_{BERT}}$ as the concatenation of the corresponding last four hidden layers of BERT, apply a non-linear transformation to obtain $h_w \in \mathbb{R}^{d_h}$ with $d_h = 512$, and finally a linear projection to $o_w \in \mathbb{R}^{|S|}$ to compute the sense scores (S is the sense inventory). Formally:

$$
\begin{aligned}
e_w &= \text{BatchNorm } (l_w^{-1} \oplus l_w^{-2} \oplus l_w^{-3} \oplus l_w^{-4}) \\
h_w &= \text{Dropout } (\text{ Swish } (W_h e_w + b_h)) \\
e_w &= W_o h_w + b_o
\end{aligned}
$$

where l_w^{-i} is the hidden state of the ith layer of the Transformer starting from its topmost layer, BatchNorm($\cdot$) is the batch normalization operation Ioffe and Szegedy (2015), Dropout($\cdot$) is the dropout regularization Srivastava et al. (2014), and Swish(x) = x $\cdot$ sigmoid(x) is the Swish activation function Ramachandran et al. (2017). During fine-tuning, the pre-trained weights of BERT are kept frozen, in line with Conia and Navigli (2021). We are actually modifying only classification head parameters, still enabling us to achieve great results.

Which data-driven method is the best? Depends on the specific use-case and application. In Loureiro et al. (2021) a quantitatively comparison between feature extraction and fine-tuning approaches has been made. 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 k-NN strategy) is the more solid choice on general grounds, even when training data is available. With respect to fine-tuning it is significantly less expensive to train and it works reasonably well for limited amounts of training data. Even if supervised classifiers are in general known to have label bias towards more frequent classes, i.e., those that are seen more frequently in the training data (Hardt et al., 2016), k-NN methods deviate from the trend and demonstrate greater robustness concerning sense bias.

2.4.3 Hybrid Approaches

As always, it is a winning strategy to take the best aspects of each approach and combine them. This is evident in the integration of purely data-driven and purely knowledge-based strategies, which consistently yields SOTA performance. By augmenting the high performance of supervised methods with the scalability and generalization capabilities of knowledge-based strategies, we enhance the system's ability to handle unseen words and senses. This hybrid approach improves data efficiency, reducing the need for extensive data, and proves more suitable for multilingual WSD. We can roughly group this family of hybrid methods based on what kind of additional information are able to exploit: (i) supervised models exploiting **glosses**, and (ii) supervised models exploiting **relations** in a knowledge graph.
The first conspicuous source of information in sense inventories consists of textual definitions, also known as *glosses*. They have proven themselves to be quite useful for increasing WSD performances, with multiple ways to exploit them being explored in the literature. GlossBERT (Huang et al., 2019), which achieved SOTA results on WSD task, is a prime example of good use of glosses. Each possible sense (gloss) of the ambiguous word is paired with the context sentence to form multiple input pairs. For example for the word *suit* in sentence *I've got an ace in the same suit* we produce these two input pairs:

1. [CLS] *I've got an ace in the same suit* [SEP] **a set of garments**

(usually including a jacket and trousers or skirt) for outerwear all of the same fabric and color [SEP]

2. [CLS] *I've got an ace in the same suit* [SEP] **playing card in any of four sets of 13 cards in a pack; each set has its own symbol and color** [SEP]

They are then input into BERT, which processes them to generate contextualized embeddings representing the compatibility between the context and the gloss. Finally, a simple linear classifier scores each gloss match to the context, allowing us to select the most appropriate sense for the ambiguous word in the given context. In our above example, is the second input pair that should produce the highest score. Another interesting approach which leverages glosses is Blevins and Zettlemoyer (2020), whose main focus is improving the performance for rare and less frequent word senses. The proposed model has a bi-encoder architecture, where one encoder processes the context containing the ambiguous word, and the other encoder processes the glosses of the possible senses. The contextual embedding of word w_i is then compared to each sense embedding s using a dot product. The sense with the highest similarity to w_i is assigned as the predicted label.

WordNet offers another rich source of knowledge in the edges that interweave its senses and synsets, i.e. *relations*. Many recent supervised systems integrate information from knowledge graphs to enhance the understanding of word senses, providing additional context and relationships between words and their senses, which are not captured by text alone. Such strategies have been very effective indeed, like EWISER (Bevilacqua and Navigli, 2020). The paper sets a new state of the art on almost all the evaluation settings considered, also breaking through, for the first time, the 80% ceiling fixed by interannotator agreement (IAA) (Gale et al., 1992a) for fine-grained WSD. The latter is an upper bound which specifies the highest performance reasonably attainable, that is, the percentage of words tagged with the same sense by two or more human annotators. The method, thanks to relations integration also obtain state-of-the-art results on multilingual WSD by training on nothing but English. One year later in 2021, Conia and Navigli (2021) proposed a novel model-agnostic approach designed to be independent of any specific machine learning model, allowing for the integration of different types of knowledge into the WSD process. But the true innovation lies in approaching WSD as a soft multi-label classification problem, allowing multiple senses to be assigned to each target word. The authors argue that forcing a system to treat WSD as a single-label classification problem (where only one sense is deemed correct for a word in a given context) fails to capture the way humans disambiguate text. While this single-label approach has proven extremely effective, as evidenced by EWISER (Bevilacqua and Navigli, 2020) achieving the estimated upper bound of inter-annotator agreement (IAA) for WSD, it overlooks a critical aspect of human text disambiguation. Previous studies (TUGGY, 1993; Kilgarriff, 1997) have shown that it

is not uncommon for a word to have multiple appropriate meanings in a given context. These meanings can often be used interchangeably because their boundaries are not always distinct.
The SapienzaNLP group has consistently been at the forefront of Word Sense Disambiguation, producing in recent years two outstanding works: ESCHER (Barba et al., 2021a) and ConSeC (Barba et al., 2021c), the current state-of-the-art in the Unified Evaluation Framework (Raganato et al., 2017).

ESCHER It redefines Word Sense Disambiguation (WSD) as an Extractive Sense Comprehension (ESC) task. This innovative method diverges from traditional WSD approaches by incorporating elements of machine comprehension, treating WSD similarly to extractive question answering. In this framework, instead of selecting predefined sense labels, the model receives a sentence with a target word and its possible sense definitions as input. The model then extracts the text span corresponding to the gloss that best expresses the target word's most suitable meaning. ESCHER demonstrates efficient utilization of training data, achieving comparable performance to its closest competitor while requiring nearly three times fewer annotations. This efficiency is crucial in a task where high performance is typically dependent on extensive data coverage and can provide an effective solution to the WSD problem of the *data acquisition bottleneck*.

ConSec It approaches WSD not as a discrete classification problem but as a continuous process of sense comprehension. This method diverges from traditional approaches, where each word is disambiguated individually based on its context, without considering the senses assigned to other words in the same context. To overcome this limitation, ConSeC leverages the recent re-framing of WSD as a text extraction problem (Barba et al., 2021a), allowing the disambiguation of a target word to be conditioned not only on its context but also on the explicit senses assigned to nearby words. ConSeC incorporates the process of understanding word meanings within the broader context of the sentence, mirroring a more natural, human-like approach to language comprehension. We can see in Figure 10 that ConSeC model extracts the predicted gloss by analyzing not only the word and the context, but also the definitions of other disambiguated words within the same context. In this case, *trackball* definitions helps the model to orient its decision to the electronic meaning of *mouse*.

After a reviewing of the main WSD approaches, it is evident that pretrained language models play a crucial role in achieving high performance for both supervised and knowledge-based methods. The information contained within knowledge bases remains valuable, and as discussed in previous sections, many successful supervised methods effectively hybridize with knowledge-based approaches. In contrast, purely

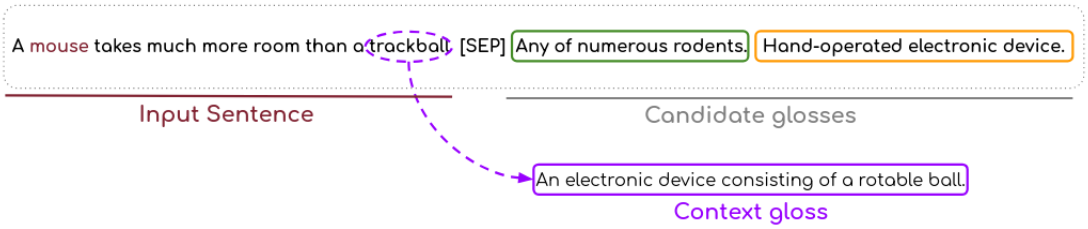


Figure 10: ConSeC input example where the word to disambiguate is *mouse*.

knowledge-based methods are entirely outperformed by these hybrid and supervised techniques.

How do we address the data acquisition bottleneck in WSD? Should we focus on creating and augmenting existing datasets with additional data, or should we explore alternative solutions, such as leveraging knowledge from semantic resources? In the following section, we delve into the issue and examine other pertinent challenges in the field, providing a comprehensive overview of future directions of WSD research.

2.5 WSD Challenges

2.5.1 Knowledge acquisition bottleneck

Disambiguating words in context presents numerous challenges and implications. The scarcity of semantically annotated data, often referred as the *knowledge acquisition bottleneck*, remains a significant challenge in the field of Word Sense Disambiguation (WSD). This issue has persisted for decades, with the largest manually curated dataset, WordNet, dating back 25 years (Miller et al., 2008). Despite substantial efforts to generate more data, such as the OntoNotes and CoNLL datasets, there is still a pressing need for larger datasets, especially for non-English languages, where coverage is insufficient. Expecting an adequate amount of manually tagged data is unrealistic. Consequently, many studies (Pasini, 2020; Pasini and Camacho-Collados, 2020) have thoroughly examined this issue and proposed advancements to mitigate it by producing annotated data semi-automatically and automatically. A very interesting approach is Exemplification Modeling (ExMod) (Barba et al., 2021b), where BART (Lewis et al., 2019), an encoder-decoder architecture, given the definition of a word, generates an a usage example of that word. For example, given the concept "*Bank*: a building where financial services are offered" ExMod generates "He went to the *bank* to deposit a check". The model is trained to generate examples that are coherent, contextually appropriate, and illustrative of the given concept. This approach is capable of automatically producing high-quality sentences that accurately convey the requested semantics. Consequently, it facilitates the creation of sense-tagged data covering the full range of meanings in any lexical inventory, significantly mitigating the problem of sense-annotated data scarcity. Human annotators concur that the generated sentences

are as fluent and semantically coherent as those in manually annotated corpora. Furthermore, when these synthetic data are incorporated into gold-standard datasets, they enable the current state-of-the-art performance to be surpassed. The practice of augmenting standard training datasets such as SemCor with additional data has proven to be highly effective. Incorporating supplementary training data, such as the English Princeton WordNet Gloss Corpus (WNG), significantly enhances performance, despite the presence of noisy silver annotations in this corpus. Several studies (Bevilacqua and Navigli, 2020; Conia and Navigli, 2021) have demonstrated that appending WNG to SemCor improves system performance, establishing this approach as a rule of thumb in the field.

Is this the way we want to go? Continuing to add more and more data cannot be the solution. Promising works such as Conia and Navigli (2021) and Barba et al. (2021a) demonstrate that an alternative approach is possible: achieving comparable performance to leading methods with a reduced amount of annotated text. This is feasible through knowledge integration, which I believe will be a critical aspect for the future of WSD.

2.5.2 Super-human performance or too fine-grained inventories?

Another important aspect is the discussion about WSD systems performances. After Bevilacqua and Navigli (2020) has broken for the first time the 80% performance ceiling (established by the interannotator agreement (Gale et al., 1992a)), subsequent models have continued to get better and better each time, reaching 83.2 F1-score on ALL dataset (Raganato et al., 2017). It is unclear how to interpret a performance which beats the interannotator agreement: if humans cannot agree more than a certain percentage of times, what does it mean if a system overcomes that benchmark and is more accurate? Is it considered super-human intelligence? A well-known issue with WordNet is the fine granularity of its sense distinctions (Navigli, 2009). It is possible that the fine-grained WSD task itself is ill-posed and needs reconsideration. The excessively fine granularity results in senses that are difficult to distinguish, even for experienced human annotators. For instance, the noun *star* has eight fine-grained senses in WordNet, two of which refer to a "celestial body" differing only in their visibility from Earth. Both meanings translate to *estrella* in Spanish, rendering this sense distinction irrelevant for Machine Translation. In fact, it has been demonstrated that coarse-grained distinctions are generally more suitable for downstream applications such as MT and Information Retrieval. The SemEval-2007 competition was the first to shift the focus toward coarse-grained WSD, allowing for the assessment of state-of-the-art systems on sense inventories with lower granularity than WordNet.

We need coarse-grained inventories! Fine-grained inventories often render the WSD task unnecessarily difficult both in the performance of disambiguation systems and in

the agreement between human annotator. Creating such inventories is neither a solved nor an easy task. Efforts have been made to develop coarse-grained datasets aimed at reducing the excessively detailed sense distinctions (Meaningful Clustering of Senses (Navigli, 2006), Coarse Sense Inventory (CSI) (Lacerra et al., 2020), CoarseWSD-20 (Loureiro et al., 2021)), but none of them has resulted in a large-scale inventory that can be reliably utilized in lexical-semantic tasks like our coarse-grained resource in Proietti et al. (2024) (more details in Section 3). Like our resource, (Navigli, 2006) presents a method for reducing the granularity based on the mapping to the Oxford Dictionary of English (ODE) (Soanes and Stevenson, 2003), a manually crafted dictionary. The hardness of WSD strictly depends on the granularity of the sense distinctions taken into account (Navigli, 2009) and it seems therefore that the major obstacle to effective WSD is the fine granularity of the WordNet sense inventory, rather than the performance of the best disambiguation systems. Interestingly, (Ng et al., 1999) shows that, when a coarse-grained sense inventory is adopted, the increase in interannotator agreement is much higher than the reduction of the polysemy degree.

It is now evident that state-of-the-art SOTA models surpassing the 80% inter-annotator agreement upper bound by a significant margin can be attributed to the excessively fine-grained representation of sense-annotated corpora. It certainly does not mean that WSD task has been solved, we are still far. In "What's the Meaning of Superhuman Performance in Today's NLU?" paper (Tedeschi et al., 2023), it is discussed the tendency of researchers in Natural Language Understanding (NLU) to assert with conviction that their models have achieved superhuman performance, along with the provocative claim that certain tasks (like WSD) have been definitively solved. Given the significant impact of claiming superhuman performance, these assertions warrant a critical examination and it is crucial for researchers to thoroughly understand the underlying factors contributing to these results. Most of the time, despite high scores on benchmarks, the practical effectiveness of these models in real-world applications remains questionable. The core issue lies in whether high-performing WSD algorithms genuinely possess human-like understanding and inferential capabilities, or if the metrics used to evaluate them are inherently flawed. The work, aspires to encourage skepticism and rigor when evaluating claims of "superhuman" performance and to motivate benchmark creators to develop more robust and transparent benchmarks to advance our understanding of NLU systems.

Necessity for more robust Benchmarks WSD is now considered among the array of NLP tasks that appear to be solved. However, a closer examination beyond the surface of raw performance figures reveals that current approaches still make trivial mistakes that a human would typically avoid (Maru et al., 2022). This paper, "Nibbling at the Hard Core of Word Sense Disambiguation," aims to address persistent challenges in WSD, such as rare senses and highly ambiguous contexts, by focusing specifically

on these difficult cases. In doing so, the paper follows the recommendations outlined by (Tedeschi et al., 2023), which advocate for the development of new and more robust benchmarks that highlight and investigate the weaknesses of WSD systems. The paper sets out to achieve this by thoroughly examining why these cases pose such significant challenges and evaluating the performance of current WSD models in these scenarios. For instance, they produce and release *hardEN*, a new challenging evaluation benchmark made up solely of instances which none of the investigated state-of-the-art systems can solve. By focusing on the hard core cases, this research aims to guide future efforts in developing more robust and effective WSD systems. To consider WSD as solved, it would be reasonable to expect disambiguation errors to be little more than mismatches between the reference ground truth and another different, but still reasonable interpretation. Instead, to give a little bit of context, consider the following example, where the predicted sense by ESCHER model (Barba et al., 2021a) for the word *wind* is compared with the gold answer from the test set of SemEval-2013 Task 12 (Navigli et al., 2013b):

Input sentence: The banks battling against a strong wind in the USA several years later. Investors and regulators (. . .)
Gold sense: A tendency or force that influences events.
ESCHER answer: Air moving (. . .) from an area of high pressure to an area of low pressure.

In this example, the contextual meaning of the word *wind* is clear to any English speaker, with no cues in the sentence that would lead a human reader to choose the "air" meaning. This is an illustrative case of why, despite having achieved (on paper) superhuman performance (Tedeschi et al., 2023), WSD systems continue to make errors that are not justifiable by inter-annotator agreement. Another contribution of this paper is the development of an amended version (ALL_{NEW}) of the standard evaluation benchmark (ALL) (Raganato et al., 2017). The goal is to provide a cleaner test set by removing non-system-dependent issues and correcting lexical and semantic inaccuracies. To achieve this, an expert linguist with extensive experience in tagging with the WordNet inventory was first asked to review the test instances in ALL, tagging each instance with one of the following labels:

- **unchanged**, to indicate that the annotator agreed with the existing ground truth;
- **fine-grained**, to indicate that one or more senses need to be added to the ground truth, without removing the existing ones;
- **error:token-lemma**, to indicate that the test instance was originally assigned a wrong lemma, or was improperly tokenized;

- **error:pos**, to indicate that the test instance was originally assigned a wrong part of speech (PoS);
- **error:sense**, to indicate that one or more senses in the ground truth are wrong;
- **error:inventory**, to indicate that the ground truth is wrong, but there is no appropriate sense for the target word in the inventory of WordNet.

The same linguist was then tasked with updating the instances from ALL based on the labels assigned during the first phase. This involved assigning additional word senses to instances labeled as **fine-grained**, amending existing annotations for **error:sense** cases, correcting PoS tagging errors (**error:pos**), and fixing lemmatization and tokenization errors (**error:token-lemma**). The instances were subsequently updated with the appropriate word senses. As a result, we developed ALL_{NEW}, featuring 4.917 polysemous instances that amend the original ALL dataset.

2.5.3 Rare and unseen senses

Zipf's Law, a well-known statistical principle in natural language, states that the frequency of any word is inversely proportional to its rank in the frequency table. This results in a distribution where a few words are extremely common, while the vast majority are rare. Consequently, in WSD, annotated datasets are dominated by a small number of frequent words and senses, with many rare senses receiving little to no representation. This disproportionate representation creates a significant challenge for Word Sense Disambiguation systems. The over-representation of common senses in training data biases models towards these senses, leading to poor performance on rare senses. This imbalance complicates the accurate disambiguation of words with multiple, less frequent meanings. Addressing this issue has been a focal point for WSD researchers, with considerable efforts dedicated to finding solutions (Blevins and Zettlemoyer, 2020; Su et al., 2022; Pilehvar and Collier, 2017).

2.5.4 Fixed Sense Inventories: blessing or curse?

Another inherent issue of the task is that systems need to choose senses from *one-size-fits-all* sense inventories, that is, discrete and finite sense inventories. While it is computationally convenient, it prevents scaling to newer and more creative uses of words, and constrains systems to a given sense granularity (as previously discussed) and to a limited number of senses, which may be sub-optimal for the real-world applications. A first emancipatory gesture from discrete meanings was taken by the introduction of Word-in-Context (Pilehvar and Camacho-Collados, 2019). This task evaluates the semantic competence of models without relying on predefined inventories. WiC requires a model to take as input two contexts containing the same target word and predict whether the word is used with the same meaning in both contexts.

We need to go beyond sense inventories! To this end, two tasks have emerged that reframe the WSD task by eliminating the need for predefined inventories: (i) **lexical substitution** (McCarthy and Navigli, 2007b), where models disambiguate a word in context by identifying meaning-preserving substitutes, and (ii) **definition modeling** (Noraset et al., 2016), where instead of selecting the most relevant sense class, a system generates a description of the word's meaning. While the lexical substitution for WSD is still a research direction to be pursued, definition modeling has already been employed in WSD, reframing it from Natural Language Understanding to Natural Language Generation with Generationary (Bevilacqua et al., 2020). The authors propose a method where models generate glosses, i.e. short definitions or explanations of word meanings directly from context. By generating glosses rather than selecting from a fixed inventory, the model can handle a wider range of word senses, including those not covered by traditional sense inventories. This flexibility is particularly beneficial for handling rare and domain-specific senses. They also showed that Generationary matches or surpasses the state-of-the-art in discriminative tasks such as WSD and WiC.

2.5.5 Where to go next?

Research community thinks that definition modeling is a promising way forward for the task, expanding the scope of WSD without big sacrifices as a trade-off (Bevilacqua et al., 2021). A primary challenge in generating sense definitions lies in the evaluation method: how should these systems be evaluated in the context of WSD? The most straightforward and intuitive approach involves computing similarity scores between the system-generated gloss and the glosses associated with the candidate senses. The candidate with the highest similarity score is then predicted as the correct sense. In Bevilacqua et al. (2020), cosine similarity is employed between the gloss vectors produced via the Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019) (more accurate sentence embedders are now available). The predicted sense $\hat{s}$ is selected as follows:

$$\hat{s} = \arg\max_{s \in S_t} sim(\hat{g}, G(s))$$

where S_t is the set of possible senses s for target t, $\hat{g}$ is the generated definition, $G(s)$ is the gloss of sense s and sim is the SBERT similarity. In our LLMantics benchmark (Martelli et al., 2024), one evaluation setting is based on the same *definition modeling* approach as in Bevilacqua et al. (2020). In this setup, the Large Language Model is asked to generate a definition for a word in context. As with Generationary, the choice of evaluation method is critical and requires further refinement. The similarity matching method described earlier has several issues. For instance, how do we ensure that each generated definition leads the system to select the most semantically related sense? This largely depends on the sentence embedder used, and it is well-known

within the research community that these embedders do not always perform as expected. This issue must be resolved and thoroughly analyzed in the context of WSD to ensure reliable and more accurate results.
The potential of the lexical substitution task in WSD has not been thoroughly explored, but, however, if applied judiciously, it could yield promising results. For instance, in the sentence "Would you give me a lift?", if the word *lift* is disambiguated by proposing *ride* as a candidate for substitution, this information can assist a WSD system in discerning the correct meaning and guiding its decision towards the sense *"a ride in a car"*. One promising approach for future research, which I would call MaskWSD, leverages the Masked Language Modeling (MLM) pre-training objective of BERT-like models. This method involves asking the model to generate a list of the most likely words that can substitute the target word to be disambiguated. After obtaining these words, their embeddings can be processed (either concatenated or averaged) and compared to the embeddings of the target word's possible meanings. The sense corresponding to the highest similarity score is then selected.
This innovative idea can be further extended to Large Language Models (LLMs). By creating an ad-hoc prompt, one can instruct the LLM to produce a list of n-words that could substitute the word to be disambiguated. This evaluation setting, referred to as Lexical Substitution (LS_σ), can be seamlessly integrated into our LLMantics benchmark, providing a novel and effective way to enhance WSD performance.

We have thoroughly analyzed the limitations and challenges of WSD that the research community is actively addressing. Our discussion includes various solutions employed by researchers to tackle these issues, as well as our own contributions to the field. Firstly, to address the problem of fine-grained inventories (Section 2.5.2), we create a new coarse-grained resource (Proietti et al., 2024). Our second contribution, as detailed in Martelli et al. (2024), primarily focuses on investigating the semantic capabilities of LLMs. This work aligns with efforts such as Maru et al. (2022), which aims to create novel and robust benchmarks for WSD and NLU in general (Section 2.5.2). In addition, one of the evaluation settings of Martelli et al. (2024) exploits *definition modeling* to perform WSD, that is a way to overcome the limitations imposed by fixed sense inventories (Section 2.5.4). Finally, we presented promising future research directions aimed at further improving and exploring the WSD task. These directions will continue to enhance our understanding and development of more effective WSD solutions.
A significant step towards future advancements has been made with the evaluation of Large Language Models in the Homonymy Disambiguation task (Section 5). Although this work is not yet exhaustive, it represents progress in expanding the scope of LLMantics. The goal is to develop it into a comprehensive and robust benchmark by incorporating additional tasks and enhancing its evaluative capabilities.

3 Analyzing Homonymy Disambiguation

We now move on to present our work on the analysis of homonym disambiguation, which results in a paper presented in May 2024 at the LREC-COLING conference. Its primary contribution is the development of a new coarse-grained resource designed to provide the research community with a valuable dataset for testing both existing and novel WSD systems. This work aligns with the branch of WSD research that seeks to simplify sense granularity (Section 2.5.2), addressing the issue that excessively fine-grained senses can be challenging even for humans. This granularity can hinder the effective application of WSD systems in downstream tasks such as MT and IR. Current Pretrained Language Models (PLMs) appear to have the capability to perform disambiguation, but the extent and level of granularity at which they operate remain unclear. In this paper, we address these issues by first introducing a new large-scale resource that leverages homonymy relations to systematically cluster WordNet senses, effectively reducing the granularity of word senses to a very coarse-grained level. We separate homonymous senses in WordNet and cluster polysemous ones by manually aligning word senses with their corresponding definitions in the Oxford Dictionary of English (ODE) (Soanes and Stevenson, 2003). This approach allows our resource to strike a balance between the semantic richness of word meanings and their suitability for practical applications. Furthermore, we use this resource to train Homonymy Disambiguation systems and investigate whether PLMs can inherently differentiate coarse-grained word senses. This study raises the following research questions:

- Is the capability to properly capture homonymy already acquired by PLMs during pretraining? (Section 3.3)
- Do fine-grained WSD systems inherently distinguish homonymous senses? (Section 3.4)

Finally, since the new resource clusters the fine-grained senses of ambiguous words based on homonymy relations, in Section 3.5 we conduct a study analyzing these clusters using representations generated by PLMs.

3.1 Resource creation

Over the past years, a variety of manual and automated techniques have been proposed for clustering senses within well-known sense inventories. Vial et al. (2019) proposes a methodology leveraging semantic relationships within WordNet, specifically hypernymy and hyponymy, to compress the sense vocabulary by clustering related senses into broader categories, thus reducing granularity. Lacerra et al. (2020) introduces CSI (Coarse Sense Inventory), which aligned 83K WordNet synsets with 45 high-level semantic labels using a combination of manual and semi-automatic steps.

While both approaches aim to address fine-grained sense distinctions in WordNet, Vial et al. (2019) maintains the original sense inventory framework, whereas CSI organizes senses into a new domain-based inventory. Similar to our work, Maudslay and Teufel (2022) employes an automatic approach to link WordNet sense definitions to the Oxford English Dictionary, enriching WordNet with homonymy annotations.
Despite the valuable contributions of the aforementioned works in simplifying the organization of word senses, none have produced a large-scale coarse-grained inventory that can be reliably utilized in lexical-semantic tasks. Furthermore, except for Maudslay and Teufel (2022) work, previous studies have not exploited homonymy, despite its potential as a powerful linguistic tool for systematically distinguishing between word senses and reducing their granularity. Moreover, even when WSD systems are evaluated using the latest coarse-grained inventories, they achieve only up to 85.9% accuracy (Lacerra et al., 2020), which is still insufficient for significant improvements in downstream tasks. We now proceed to the creation of the resource.

In order to transfer homonyms from the Oxford Dictionary of English (ODE) to WordNet, we ask three linguists to manually associate WordNet senses with Oxford homonyms based on their definitions:

- as a first step, we automatically extract all (lemma, PoS) pairs of ODE that have at least two homonymous senses. More formally: let ODE be the set of (lemma, PoS) pairs in the Oxford Dictionary of English, then $\text{ODE}_h \subset \text{ODE}$ is the set of (lemma, PoS) pairs with at least two homonymous senses. Then, considering a (lemma, PoS) pair l, $H_l = \{h_1, h_2, ..., h_n\}$ is the set of its homonymous senses, and therefore $\text{ODE}_h = \{l \in \text{ODE} : |H_l| > 1\}$;

- then, each linguist associates the fine-grained senses of WordNet with the coarse-grained ones present in ODE, when possible. Being WN the set of all (lemma, PoS) pairs in WordNet, we define $\text{WN}_h = \text{WN} \cap \text{ODE}_h$. For every $l \in \text{WN}_h$, let $S_l = \{s_1, s2, ..., s_k\}$ be the set of its candidate WordNet senses.
  Then, for each $l \in \text{WN}_h$, our goal is to find a mapping f_l from set S_l to set H_l. It may happen that some sense in WordNet cannot be associated with any homonymy cluster in Oxford, or vice versa. Indeed, each mapping f_l is not necessarily injective or surjective, i.e. multiple senses in S_l can be mapped to the same homonymous sense in H_l, and not all elements of H_l necessarily have to be mapped from S_l;

For the manual annotation process just mentioned, we measure the Fleiss' kappa score (Chklovski and Mihalcea, 2003) to be $\kappa = 0.79$, highlighting a substantial agreement among the annotators.
The obtained mapping f_l between WordNet senses and ODE homonymy clusters, however, does not cover the entire WordNet repository because: (i) the manual mapping involves only those (lemma, PoS) of ODE that have at least two homonymous

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

Figure 11: Example of two (lemma, PoS) pairs and their corrispective homonyms and synsets. Source: (Proietti et al., 2024)

senses; (ii) even for those (lemma, PoS) that have been mapped, some of their WordNet senses do not have a correspondence in ODE. We cannot leave the resource as it is, and because we are interested in enriching the entire WordNet repository with homonymy information, we devise an automatic strategy for extending the resource.
Each non-mapped WordNet sense belongs to one of the two disjoint sets:

- $U_1 = \bigcup_{S_l \in \text{WN}_r} S_l$, where $\text{WN}_r = \text{WN} \setminus \text{WN}_h$ is the set of WordNet (lemma, PoS) pairs not involved in the previously described manual annotation procedure. It is the set containing all candidate senses of the WordNet (lemma, PoS) pairs not in ODE_h;

- $U_2 = \{s \in S_l \mid l \in \text{WN}_h, s \notin \text{dom}(f_l)\}$, where its senses, instead, are those for which human annotators could not identify a matching homonymy cluster in ODE, with $|U_2| = 506$.

Since we adopt the Oxford Dictionary of English as the authoritative inventory for homonyms, if a WordNet (lemma, PoS) pair is not in ODE_h (hence it belongs to U_1), we make the assumption that it does not have any homonymous senses and, as a result, all candidate senses associated with such (lemma, PoS) are automatically mapped to a single newly-created homonymy cluster.
For 250 of the 506 senses $\in U_2$ the solution is straightforward: in fact, each of these is the only non-mapped sense of a (lemma, PoS) pair; therefore, since it does not belong to other homonymy clusters in ODE, we can create a new cluster that contains only this sense. For the remaining 256 senses, instead, we are not able to automatically determine whether they should be new singleton clusters, or whether some of them should be grouped in the same cluster. For this reason, we ask the annotator to inspect these senses and decide the composition of the new clusters. As a result, our resource maps every (lemma, PoS) pair in WordNet to its set of homonymous senses. See Figure 11 for viewing two (lemma, PoS) examples mapped to their homonyms.

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

Figure 12: Number of instances in standard WSD datasets. The FGA (Fine-Grained Ambiguous) items are the instances with more than one candidate sense, while the HA (Homonymy Ambiguous) items are those instances that have more than one candidate homonymy cluster. Source: (Proietti et al., 2024)

3.2 Experimental setup

In our work, we will first describe which data are adopted, the models involved, as well as the hyperparameters used for their architecture and training process.

Data Having the resource, we just have to tag instances of standard WSD datasets with their coarse-grained senses. The considered datasets are:

- **SemCor** (Miller et al., 1993), already mentioned,a large sense-annotated corpus for WSD, generally used as training set (see Figure 5).
- **WordNet Examples**, contextual examples associated with specific synsets in WordNet.
- **SemEval-2007** (Pradhan et al., 2007), typically used as development set for WSD systems (see Figure 5).
- **ALL**$_{NEW}$ (Maru et al., 2022), already discussed in Section 2.5.2, is an amended version of the ALL test set Raganato et al. (2017).

An established practice in the WSD literature is to use SemCor as training set, Semeval-2007 as validation set and ALL as test set. Following recent works (Barba et al., 2021a; Bevilacqua and Navigli, 2020; Conia and Navigli, 2021), we also include the WordNet Examples dataset in our training data. However, after mapping the aforementioned datasets with coarse-grained sense annotations thanks to our resource, we observe too few polysemous instances in the development and test sets, as shown in Figure 12. Such polysemous degree distribution would impede the effectiveness of our analysis of coarse-grained WSD systems. Therefore, we concatenated all these datasets and re-split them into new training, validation, and test sets, ensuring a more suitable number of polysemous instances for our purposes (see Figure 13).
In Section 5, the total testing split is referred as *test*, its FGA subset as $test_{FGA}$ and its Homonymy Ambiguous portion as $test_{HA}$.

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

Figure 13: Number of instances in the new train/dev/test split. Source: (Proietti et al., 2024)

Models Among all existing PLMs we choose four of the most popular ones: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020) and DeBERTa-v3 (He et al., 2021). BERT and RoBERTa are Transformer-based encoder models pretrained using the MLM objective. In contrast, ELECTRA, as introduced by Clark et al. (2020), employs a different pretraining objective known as Replaced Token Detection, which was also utilized in training DeBERTa-v3. Given these differing pretraining objectives, we anticipate that their output representations will exhibit distinct properties, thereby enhancing the robustness and generality of our analysis.

Hyperparameters The architecture used to investigate whether WSD systems can also function as homonymy disambiguators was inspired by Conia and Navigli (2021) and follows the equations described in 2.4.2. In line with the family of supervised WSD approaches that employ fine-tuning, the model consists of a frozen pre-trained language model, a non-linear layer (with a dimension of 512), and a linear classifier. For a given word in context, its contextualized representation is created by concatenating the last four hidden layers of BERT, followed by a non-linear transformation, and then a linear projection to compute sense scores. The word representation is batch normalized, transformed using a Swish activation function, and regularized with dropout. Only the classification head parameters are fine-tuned, while the pre-trained weights of the transformer model remain frozen.
During the training phase, we used the following hyperparameters: RAdam as the optimizer, a learning rate of $1e^{-4}$, a batch size of 8, and a dropout rate of 0.1.

3.3 Do LMs learn how to disambiguate homonyms already during pretraining?

Our goal is to establish the extent to which current PLMs are capable of disambiguating homonyms. To investigate whether Pre-trained Language Models inherently acquire the ability to disambiguate homonymous senses during pretraining, we use a *distance-based* disambiguation method. This method falls within the category of *feature extraction* techniques, as discussed in Section 2.4.2, and does not require any

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

Figure 14: Distance-based Homonymy Disambiguation accuracy when using *Cosine* and *Euclidean*. The highest accuracy is in bold, and the top two accuracy values with the two distance measures are underlined. Source: (Proietti et al., 2024)

additional training.

For a given test instance to be disambiguated, we aim to assign it to the homonymy cluster that contains the closest sense, determined by either *Cosine* or *Euclidean* distance. To compute the distance between test instances and candidate senses, we use PLMs to extract their vector representations. An instance is represented by the contextualized embedding of the word in context, extracted from the last hidden layer of the PLM. If the word has been split by the tokenizer, we use the first sub-word embedding.

Each sense may have multiple representations, corresponding to different training instances tagged with that sense. We calculate the distance between the test instance and each of these sense representations and use the smallest distance for the prediction. However, it is not possible to extract vector representations for all senses, as some do not have any associated training instances. Consequently, we restrict the test data to instances whose candidate homonymy clusters contain at least one sense with a vector representation. This refinement yields a specific subset of $test_{HA}$, $test_{HAp}$, which consists of 609 items.

The results of the experiments are presented in Figure 14. Across all experiments, we achieve an accuracy exceeding 89%. Notably, the BERT model demonstrates the best performance, with the Cosine distance measure performing slightly better than Euclidean. Although the number of test instances is limited, this simple method, which requires no additional training, yields impressive results. Interestingly, the Most Frequent Sense baseline (which always selects the most frequent sense among its candidates) achieves a disambiguation accuracy of 84.40, more than 10 points lower than our best distance-based Homonymy Disambiguation system. This suggests that, although PLMs are not explicitly trained to disambiguate homonyms during the pretraining phase, their extensive exposure to textual data frequently involves homonymous words. In Section 5, we compare this method with Large Language Model-based systems for Homonymy Disambiguation (Table 9) and observe comparable results.

Test data	System	WSD	HD
Total	WSD	81.77	**99.23**
	HD	-	99.16
HA	WSD	73.91	**94.13**
	HD	-	93.64
HA_p	WSD	74.06	**96.39**
	HD	-	96.06

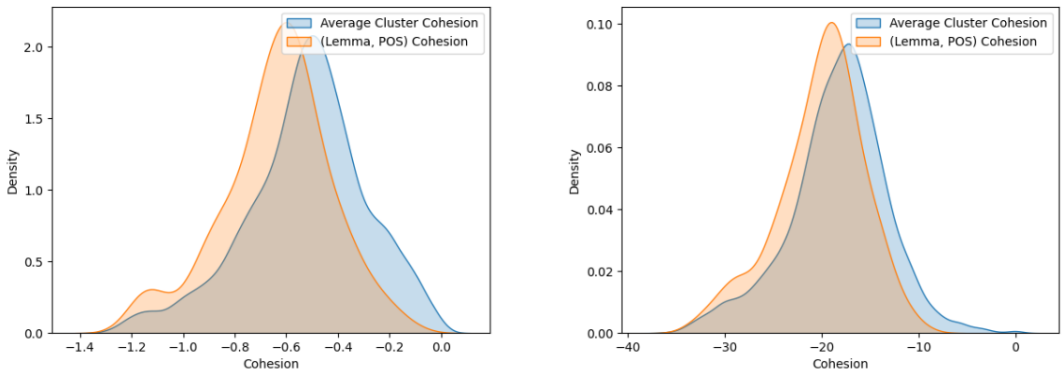
Figure 15: Performance of WSD and HD systems when evaluated on both tasks. The accuracy values are measured on different samples of the test set described in Figure 13: (i) **Total** represents the entire test set (*test*), (ii) **HA** (Homonymy Ambiguous) comprises only those instances that have more than one candidate homonymy cluster($test_{HA}$), and (iii) $\mathbf{HA}_p$ comprises only test instances used in Section 3.3 ($test_{HA_p}$). The best results in HD for each test set sample are in bold. Source: (Proietti et al., 2024)

3.4 Are WSD systems also homonyms disambiguators?

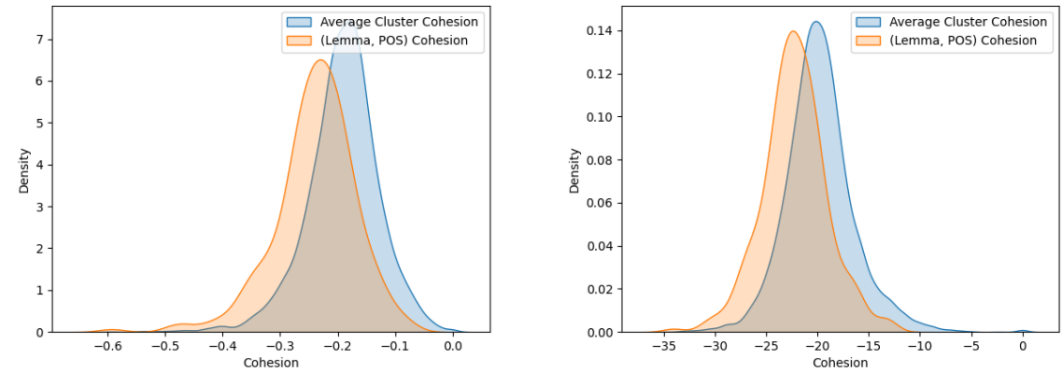
We aim to investigate the relationship between Homonymy Disambiguation and fine-grained Word Sense Disambiguation. Since Homonymy Disambiguation is a simpler task compared to WSD, we are interested in determining whether a system trained for the more complex task of WSD can effectively distinguish homonyms as well as a system trained specifically for Homonymy Disambiguation. To do this, we map the fine-grained senses predicted by a WSD system to their corresponding homonymy clusters, allowing us to evaluate its performance on Homonymy Disambiguation. Additionally, we compare this performance with that of a system trained specifically for Homonymy Disambiguation. Both systems are based on a BERT model, chosen because it demonstrated the best performance in distance-based disambiguation (Section 3.3), and feature a classification head (as described in Section 3.2). We train both models for up to 10 epochs, selecting the checkpoint with the highest accuracy on the validation set described in Figure 13.

As expected, the WSD system demonstrates the ability to disambiguate homonyms effectively. Both systems exhibited comparable accuracy across all three test sets, with the fine-grained system even outperforming the coarse-grained system on HD task. This outcome aligns with the expectation that a model trained to distinguish fine-grained senses, which is a more challenging task, can implicitly differentiate between coarser homonymous senses, which is an easier task.

Additionally, training a dedicated Homonymy Disambiguation system yielded better results compared to a distance-based disambiguation system. On the $test_{HA_p}$ set, the former achieved an accuracy score of 96.06, while the best distance-based disambiguation system achieved 95.24. Remarkably, the WSD system reached an



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



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

Figure 16: Density function estimations for cluster (CC) and (lemma, PoS) cohesion (LC). Source: (Proietti et al., 2024)

accuracy of 96.39. These findings suggest that a fine-grained WSD system can effectively handle homonymy disambiguation, often surpassing systems specifically trained for the task.

3.5 Clustering Analysis

At the end, we perform a clustering analysis on the contextualized representations from the last hidden layer of BERT and DeBERTa. These PLMs are not only among the top performers in distance-based Homonymy Disambiguation (see Figure 14), but they also exemplify the two distinct pre-training strategies discussed in Section 3.2.

We are interested in getting a general sense of how close the representations of senses in the same homonymy cluster are, and, contextually, how far from each other those of different homonymous senses of the same (lemma, PoS) are.

To conduct the analysis, we require two metrics: (i) **Cluster Cohesion (CC)**, which measures the proximity of senses within the same homonym cluster, and (ii) **(lemma,**

PoS) Cohesion (LC), which quantifies the distance between homonym clusters of the same (lemma, PoS). Higher CC values indicate better cohesion, while lower LC values are desirable. For the mathematical details of these metrics, please refer to the main paper (Proietti et al., 2024).
Without delving into the specifics of the analysis, let's examine the cluster and (lemma, PoS) cohesion density functions in Figure 16. High cohesion is indicated by narrow distributions, while low cohesion suggests more spread-out values. Notably, the average cluster cohesion distribution is shifted to the right compared to the (lemma, PoS) cohesion distribution. This shift indicates that our homonymy-based clustering groups senses that are, on average, closer to each other.
The paper also includes a quantitative analysis comparing how often the cohesion of a given (lemma, PoS) is greater than its average cluster cohesion, and vice versa. In most cases, the average CC is greater than LC, confirming previous results. This demonstrates that our resource effectively clusters senses that are closer to each other (in terms of cosine and Euclidean distances) than to other senses of the same (lemma, PoS).

In summary, we have presented an approach for clustering WordNet senses based on homonymy relations. This results in a comprehensive, high-quality resource that significantly reduces the granularity of WordNet. We conducted extensive experiments utilizing this new resource, which we hope will be valuable for future researchers and for downstream tasks requiring coarser senses. Now, we turn our attention to the other contribution (Martelli et al., 2024).

4 A Semantic Benchmark for Large Language Models

Following the creation of a resource for Homonymy Disambiguation, we aimed to develop a new type of resource: a semantic benchmark for Large Language Models, named LLMantics. The paper (Martelli et al., 2024), is currently under review and was sent to the EMNLP 2024 Conference which will be held in Miami, Florida. LLMantics is benchmark whose first objective is to investigate lexical semantic capabilities of *instruction-tuned* Large Language Models.

Foundational LLMs are initially pre-trained on vast amounts of text data to develop a general understanding of linguistic patterns and to generate human-like text. These foundational models provide the essential linguistic and contextual framework necessary for a wide range of natural language processing tasks. Instruction-tuned LLMs take this a step further by undergoing additional fine-tuning with datasets that contain natural language instructions for specific tasks. This fine-tuning process significantly enhances the model's ability to interpret and follow explicit instructions, thereby improving its performance on diverse tasks. In essence, instruction-tuned LLMs leverage the comprehensive linguistic base provided by foundational LLMs and specialize it to better handle task-specific instructions. This refinement process not only enhances the model's usability but also its effectiveness in executing specific tasks, making instruction-tuned LLMs more adaptable and efficient in practical applications.

We previously discussed the importance of developing robust benchmarks for Natural Language Understanding (NLU) tasks (Section 2.5.2). However, it is perhaps even more critical to establish comprehensive evaluation benchmarks for Large Language Models to rigorously assess their capabilities. This necessity is especially pertinent during the early stages of this rapidly advancing technology, which has already demonstrated astonishing potential and garnered worldwide attention. Creating these benchmarks is essential for several reasons. First, they provide standardized criteria for evaluating the performance of LLMs across a variety of tasks, ensuring consistency and comparability in assessments. Second, they help identify strengths and weaknesses in current models, guiding future research and development efforts. Despite the efforts made so far in terms of performance evaluation, very few works have been proposed to investigate a crucial aspect of natural language, namely lexical ambiguity.

Do LLMs really understand text semantics?

We evaluate a total of 11 open-source and commercial models, ranging in size from one billion to several hundred billion parameters, across 33 different evaluation scenarios. These scenarios include three distinct evaluation settings, namely: (i)

Word Sense Disambiguation setting ($\mathbf{WSD}_{\sigma}$[2]): in this setting, the LLM is provided with a list of candidate senses for the target word in context and must select the most appropriate sense; **Definition Generation** setting ($\mathbf{DG}_{\sigma}$): inspired by Generationary Bevilacqua et al. (2020), this setting involves asking the model to generate a sense definition for a given word within a context sentence. This method aligns with approaches that seek to eliminate the constraints imposed by predefined sense inventories, thereby enhancing the expressive power of WSD systems. Given their exceptional text generation capabilities, LLMs are ideally suited for this task; (iii) **Word-in-Context** setting ($\mathbf{WiC}_{\sigma}$): in the third setting, we evaluate the LLMs' ability to perform the WiC task (Pilehvar and Camacho-Collados, 2019), which involves determining whether the same word in two different contexts has the same meaning. More details about evaluation settings in Section 4.3.

In this work, our primary objective is to determine whether instruction-tuned LLMs are capable of disambiguating words within a given context. Additionally, we seek to understand the impact of prompt engineering and fine-tuning on disambiguation performance. To achieve this, we conduct a comprehensive prompt engineering analysis, investigating how variations in prompt formulation affect overall performance through extensive experimentation (see Section 4.4). Furthermore, we fine-tune four LLMs and examine how fine-tuning influences their performance across all our evaluation settings (Section 4.5.4).

4.1 Evaluation Benchmarks

In recent years, several evaluation benchmarks have been introduced to assess the performance of language models. Among the earliest comprehensive benchmarks for evaluating natural language understanding abilities are the General Language Understanding Evaluation (GLUE) (Wang et al., 2018) and its refined version, SuperGLUE (Wang et al., 2020). SuperGLUE includes tasks such as WiC, which requires implicit disambiguation. More recently, new benchmarks have been proposed to evaluate *instruction-tuned* LLMs across multiple tasks and domains. Notable examples include the Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020), the Language Model Evaluation Harness (Gao et al., 2023), BIG-bench (Srivastava et al., 2023), and InstructEval (Chia et al., 2023). Despite these extensive evaluation efforts, there is a surprising lack of studies explicitly investigating the disambiguation capabilities of LLMs. There are very few. Riccardi and Desai (2023) proposes "Two Word Test", a new benchmark for assessing LLM comprehension using simple two-word phrases to investigate their semantic abilities. Similarly to our

[2]In order to distinguish the setting from the task, the former is referenced by adding σ as a superscript to the standard task abbreviation.

work but limited to ChatGPT, Kocoń et al. (2023) analyzes ChatGPT's performance on a wide range of tasks, including WSD and WiC, without exploring the impact of fine-tuning.
These observations highlight the need for more specific benchmarks to evaluate the semantic capabilities of LLMs comprehensively.

4.2 Experimental setup

Data To generate the prompts for our benchmark, we rely on two well-established datasets for evaluating models' disambiguation capabilities. Specifically, for both WSDσ and DGσ, we utilize ALL, the dataset proposed by Raganato et al. (2017), which includes 7.253 instances. For WiCσ, we use the English section of the manually-curated dataset with 1.000 instances created by Martelli et al. (2021). For the fine-tuning phase, instead, we employ SemCor Miller et al. (1993) (226,036 instances) for WSDσ and DGσ, while for WiCσ, we use the training set of the same resource provided by Martelli et al. (2021). We allocate 90% to training and 10% to development.

Models We first illustrate the 11 evaluated models, including both commercial and open-source LLMs. The commercial LLMs tested are:

1. **GPT-3.5-turbo**[3] is a large and cost-effective language model optimized for chat applications by OpenAI.

2. **GPT-4-turbo**[4] is one of the latest and most efficient LLMs released by OpenAI.

The open-source models are instead:

3. **TinyLlama-1.1B-Chat-v1.0** (Zhang et al., 2024) is a language model featuring 1.1 billion parameters pre-trained on approximately 1 trillion tokens. TinyLlama utilizes the architecture introduced by Llama 2, including some advances proposed by the open-source community such as the FlashAttention (Dao et al., 2022).

4. **Phi-3-Mini-128K-Instruct** (Abdin, 2024) is a cutting-edge, lightweight open model with 3.8 billion parameters. The model is part of the Phi-3 family, available in the Mini version with two variants: 4K and 128K, indicating the context length (in tokens) it can support.

5. **stablelm-2-1.6b-chat** (Bellagente et al., 2024) is a 1.6 billion parameter instruction tuned language model trained utilizing Direct Preference Optimization (DPO) (Rafailov et al., 2023).

[3] `https://platform.openai.com/docs/models/gpt-3-5-turbo`
[4] `https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4`

6. **h2o-danube2-1.8b-chat** (Singer et al., 2024) is a chat fine-tuned model by H2O.ai with 1.8 billion parameters. They fine-tuned it through the combination of Supervised Fine Tuning (SFT) and DPO (Rafailov et al., 2023).

7. **Llama-2-7b-chat-hf** is a fine-tuned version of Llama 2 for conversational use cases (Touvron et al., 2023). For the purposes of our work, we employ the 7-billion parameter model via the Hugging Face API[5].

8. **Meta-Llama-3-8B-Instruct**[6] model is part of the Meta Llama 3 family of LLMs, a very recent collection of instruction-tuned generative text models available in 8B and 70B sizes. The 8 billion parameter model is optimized for chat applications and outperforms many other open-source chat models in common industry benchmarks, e.g. MMLU (Hendrycks et al., 2020).

9. **falcon-7b-instruct**[7] is an open-source 7B parameters decoder-only model built by Technology Innovation Institute (TII) and fine-tuned from the Falcon-7B base model using a combination of chat and instruction datasets.

10. **Mistral-7B-Instruct-v0.2** (Jiang et al., 2023) is an instruction-tuned version of the Mistral-7B-v0.2. This model utilizes grouped-query attention (GQA) (Ainslie et al., 2023) for faster inference, combined with sliding window attention (SWA) (Beltagy et al., 2020) to efficiently manage sequences of arbitrary length while reducing inference costs.

11. **vicuna-7b-v1.5**[8], developed by LMSYS organization, is an auto-regressive transformer-based language model. Specifically, it is a chat assistant developed by fine-tuning Llama 2 (Touvron et al., 2023) using user-shared conversations gathered from ShareGPT.

Hyperparameters Then we present the *hyperparameters* utilized both for inference and fine-tuning of Large Language Models (LLMs). Thanks to *transformers* library and *pipeline* method developed by HuggingFace[9], we can simply perform text generation by specifying the name of the LLM for inference, its corresponding tokenizer, and the maximum number of tokens to generate. The maximum token limit is set to 25 across all evaluation scenarios, including DG_σ, where we also experimented with maximum lengths of 50 and 100 tokens, resulting in inferior outcomes. All other parameters were maintained at their default settings, such as $do\_sample = True$ and $temperature = 1.0$. For inference with GPT-3.5-turbo and GPT-4-turbo, the OpenAI

[5] `https://huggingface.co/meta-llama/Llama-2-7b-chat-hf`
[6] `https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct`
[7] `https://huggingface.co/tiiuae/falcon-7b-instruct`
[8] `https://huggingface.co/lmsys/vicuna-7b-v1.5`
[9] `https://huggingface.co/docs/transformers/index`

Setting	Example Prompt	Example Output
$\mathbf{WSD}_\sigma$	**Question:** select the most suitable meaning for "*computer*" in the following sentence: *I am using a computer for the creation of 3-D models.* Choose the corresponding definition among: 1) a machine for performing calculations automatically; 2) an expert at calculation (or at operating calculating machines). Answer by reporting the corresponding definition and do not motivate your answer.	**Answer:** 1) a machine for performing calculations automatically.
$\mathbf{DG}_\sigma$	**Question:** define the meaning of "*floor*" in the following sentence: *they needed rugs to cover the bare floors.* Provide a definition which identifies the meaning of "*floor*" in the context provided and do not motivate your answer.	**Answer:** the inside lower horizontal surface (as of a room, hallway, tent, or other structure).
$\mathbf{WiC}_\sigma$	**Question:** determine whether the target words, *play* and *play*, refer to the same meaning in the following sentences: First sentence: *In that context of coordination and integration, Bolivia holds a key play in any process of infrastructure development.* Second sentence: *A musical play on the same subject was also staged in Kathmandu for three days.* If the meaning expressed by the two target words is the same, answer with True. Instead, if the meaning is not the same, answer with False. Do not motivate your answer.	**Answer:** False.

Table 1: Examples of *zero-shot* prompts (version v1) and their corresponding outputs for all evaluation settings. Source: (Martelli et al., 2024)

API service was employed[10].
For DG_σ, we selected two sentence embedders from the top performers on the Sentence Transformers leaderboard, namely *all-MiniLM-L6-v2* and *all-mpnet-base-v2*. Although there were variations in the results between the two, these differences were not substantial enough to demonstrate a significant advantage for either. Notably, all experiments were conducted using *all-MiniLM-L6-v2*.
To speed-up the fine-tuning of LLMs, we utilized the widely-recognized QLoRA (Quantized Low-Rank Adaptation) technique (Dettmers et al., 2023), which integrates quantization with the Low-Rank Adaptation (LoRA) training strategy (Hu et al., 2021). This approach significantly reduces the computational resources required for fine-tuning while maintaining or enhancing the model's performance on specific tasks. The LoRA configuration was set with $lora\_alpha = 16$, $lora\_dropout = 0.1$, and $r = 64$.
During the training phase, we fixed the batch size at 16, used *paged_adamw_32bit* as the optimizer, set the learning rate to 0.0002, employed the *cosine* learning rate scheduler, and ran the training for 10 epochs.

We now present a comprehensive overview of all the evaluation settings in LLMantics, followed by an in-depth exploration of the prompt strategies employed.

4.3 Evaluation settings

LLMantics, the first evaluation benchmark which enables a comprehensive assessment of explicit and implicit disambiguation abilities of instruction-tuned LLMs, covers three

[10] `https://openai.com/index/openai-api/`

settings. The evaluation settings are named after the corresponding computational task covered therein and are as follows: WSD_σ, DG_σ and WiC_σ.
Each of the aforementioned settings includes three sub-settings, namely zero-, one- and few-shot, in which prompts contain zero, one or three human-annotated examples, respectively. While the one-shot sub-setting includes one noun as example, the few-shot sub-setting covers one noun, one verb and one adverb. Each settings adopts a specific prompt formulation (see Table 1) as well as a specific evaluation strategy, which we detail below.

4.3.1 Word Sense Disambiguation

Task Framed as a classification problem, WSD is the computational task of determining the meaning of a word or expression in a given context (Navigli, 2009; Bevilacqua et al., 2021).

Prompt In this setting, models are provided with a prompt which contains: (i) an instruction requiring models to choose, among a set of candidate definitions, the one which best identifies the meaning of a given target word occurring in a sentence provided; (ii) a sentence including an ambiguous target word; (iii) a set of candidate definitions.

Evaluation strategy In order to identify a definition among the candidate ones to be considered as the model's answer[11], we compute the lexical overlap between the definition selected by the model and all candidate definitions as follows. First, let Δ_w be the set of candidate definitions associated with a given focus word w to be disambiguated, δ_σ the definition selected by the model and $\hat{\Delta}_w$ the set of gold definitions.[12] Subsequently, we tokenize[13] both δ_σ, resulting in δ_σ^τ and all definitions contained in Δ_w, thus obtaining Δ_w^τ. Then, we compute a lexical overlap score between δ_σ and all candidate definitions. Hence, we consider the answer to be the definition δ_w^{max} reporting the highest overlap score according to the following criterion:

$$\delta_w^{max} = \arg\max_{\delta_w \in \Delta_w} \frac{|\delta_\sigma^\tau \cap \delta_w^\tau|}{|\delta_\sigma^\tau \cup \delta_w^\tau|}$$

Finally, if $\delta_w^{max} \in \hat{\Delta}_w$, we classify the answer as correct.

4.3.2 Definition Generation

Task Definition generation is the computational task aimed at generating dictionary definitions for a given target word (Li et al., 2020; Kong et al., 2022; Bevilacqua et al.,

[11] Not infrequently, language models alter the selected definition or refrain from providing a response altogether, which hampers direct string comparison.

[12] $|\hat{\Delta}_w| \geq 1$

[13] Tokenization is performed using the NLTK library available at: `https://www.nltk.org/`

2020).

Prompt In this setting, models receive a prompt consisting of the following elements: (i) an instruction requiring models to generate a definition from scratch which identifies the meaning of a given ambiguous target word occurring in a given sentence; (ii) a sentence containing an ambiguous focus word.

Evaluation strategy In order to assess the correctness of the generated definition, we proceed as follows. For each instance, we first extract a dense representation for the generated definition δ_σ resulting in δ_σ^v as well as for all candidate definitions in Δ_w, thus obtaining Δ_w^v.[14] Subsequently, we compute the cosine similarity between δ_σ^v and each candidate definition δ_w included in Δ_w.The candidate definition reporting the highest cosine similarity is considered to be the answer of the model:

$$\delta_w^{max} = \arg\max_{\delta_w \in \Delta_w} \cos\_\text{sim}(\delta_\sigma^v, \delta_w^v)$$

As in the previous case, if $\delta_w^{max} \in \hat{\Delta}_w$, the answer is categorized as correct.

4.3.3 Word in Context

Task Word in Context is the computational task of determining whether two occurrences of a given target word in context refer to the same semantics (Pilehvar and Camacho-Collados, 2019; Martelli et al., 2021).

Prompt In this setting, prompts include: (i) an instruction asking models to determine whether two instances of a given target word occurring in two different sentences express the same meaning or not; (ii) two sentences, each containing an occurrence of the focus word. If the two occurrences of a focus word convey the same meaning, models are required to answer with True, otherwise with False.

Evaluation strategy In order to carry out the evaluation, we verify whether the answer provided by the model corresponds to the correct answer. If the answer cannot be classified, e.g. in some cases models answer with both True and False or they generate some text unrelated to the prompt, the answer is considered as incorrect. If neither the string True nor the string False can be detected in a given output, such instance is classified as incorrect.

[14] No pre-processing is performed before extracting the vector representation which is obtained with two models, namely all-mpnet-base-v2 and all-MiniLM-L6-v2 made available by library *sentence-transformers* (Reimers and Gurevych, 2019).

4.4 Prompt Robustness

In the realm of LLMs, designing effective prompts is crucial for unlocking the full potential of these sophisticated systems. Prompts serve as the primary interface between the model and the user, guiding responses and influencing the model's ability to perform specific tasks. The quality and structure of these prompts directly impact LLM performance and accuracy, making prompt design (commonly referred to as *prompt engineering*) a critical aspect of developing reliable and robust models. To effectively test the robustness of LLMs, an evaluation benchmark must incorporate multiple prompt settings. Different prompts can challenge various aspects of the model's understanding and generation abilities, revealing strengths and weaknesses that may not be apparent with a uniform prompt strategy.
Our experiments examine the robustness of LLMs by introducing linguistic variations to a default prompt formulation (version **v1**, see Table 1) while maintaining the underlying instruction unchanged. LLMs typically perform optimally when the prompt structure provided by users during inference closely mirrors the prompts encountered during training. To evaluate the ability of LLMs to understand user instructions despite minor linguistic variations, we employ the following additional prompt versions for each evaluation setting:

- **v1.1**: the words *"Question:"* and *"Answer:"* are removed.
- **v1.2**: the words *"Question:"* and *"Answer:"* are replaced with *"Instruct:"* and *"Output:"*.
- **v1.3**: prompts are formulated as questions.

Results Our observations indicate that formulating the prompt as a question (prompt version **v1.3**) leads to a significant performance increase. Conversely, prompt version **v1.1**, which removes the words *Question* and *Answer*, tends to confuse several models and proves detrimental to performance. For example, vicuna-7b-v1.5 experiences a performance drop of 54.69 in the DG_σ setting. This suggests that further efforts are needed to enhance models' ability to understand user input regardless of prompt formulation variations. Llama-2-7b-chat-hf shows a performance drop in WiC_σ zero-shot, while TinyLlama-1.1B-Chat-v1.0 is similarly affected in the DG_σ zero-shot sub-setting. Additionally, prompt version **v1.2**, consistently leads to lower performance in models such as TinyLlama-1.1B-Chat-v1.0, Meta-Llama-3-8B-Instruct, and Phi-3-mini-128k-instruct in WiC_σ. The efficacy of prompt version **v1.3** is particularly evident in WSD_σ few-shot settings, where Phi-3-mini-128k-instruct achieves an F1 score of 73.21, Meta-Llama-3-8B-Instruct reaches 71.66, Mistral-7B-Instruct-v0.2 attains 73.98, and vicuna-7b-v1.5 scores 71.95. These results are comparable to those of the commercial model GPT-3.5-turbo using

prompt version **v1** in the few-shot setting (see Table 4). All these results are detailed in Tables 10, 11 and 12 (Appendix A), as they are too extensive to be shown here.

Other prompts In addition to the prompts described, we explored several other prompt versions and variations during our research. The results, measured in terms of F1-score, were neither significant nor did they differ meaningfully from the reported prompt versions. Therefore, we have opted not to include these additional results in detail. We now briefly describe and show some of these alternative prompt approaches.

Version v2 and v3 As a result of the current literature, we created two prompts versions: namely **v2** and **v3**, both for WSD_σ and DG_σ evaluation settings. For the sake of brevity we only show in detail the *zero-shot* prompts.
We start with WSD_σ setting. Version **v2** prompt is a more structured and outlined version where we also employed delimiters to better help, the LLM to understand the request. Here the prompt:

> **[v2]**
> # CONTEXT #
> *<text>*
>
> # CANDIDATES LIST #
> *<definitions>*
>
> QUESTION: Given the CONTEXT, your task is to disambiguate the word "*<word>*" by selecting the most appropriate definition from the following CANDIDATES LIST.
> Your output must be the definition that best fits the CONTEXT provided. You will be penalized if you choose a definition that is not present in CANDIDATES LIST.
> ANSWER:

Version **v3** in WSD_σ is inspired by a different modality of performing WSD through LLMs (Kocoń et al., 2023). Instead of outputting the most suitable definition, we ask the model to output the sense key:

> **[v3]**
> Question: Which meaning of the word "*<word>*" is expressed in the following context:
> *<text>*
> The meanings are as follows:
> *<definitions>*
> Return only the key of the most relevant meaning.
> Answer:

Regarding DG_σ setting, we followed some guidelines and suggestions from prompt engineering literature (Bsharat et al., 2024; White et al., 2023). In order to leverage all generative LLM capabilities to produce qualitative text, we produced a custom prompt (version **v2**) following the best recommendations: from *ExpertPrompting* technique (Xu et al., 2023) to the *Persona* pattern technique (White et al., 2023). This is the result:

> **[v2]**
> # SENTENCE #
> <*text*>
>
> QUESTION: which is the sense of "<*word*>" in the previous SENTENCE?
> Your task is to generate a definition which identifies the meaning of "<*word*>" in the SENTENCE provided.
> You are an expert in linguistics and word semantics.
> Answer the question given in a natural and human-like manner.
> Do not produce any preamble text, only the definition of the word. The answer must not exceed two sentences.
> If you are not sure about the sense of the word, do not make up an answer.
> ANSWER:

We finally tried another prompt version for DG_σ (version **v3**), based on the CO-STAR Framework[15], one of the most popular and trusted templates for better results with prompts. CO-STAR stands for Context (the background information), Objective (target for LLM to focus on), Style (writing style), Tone (response attitude), Audience (the intended receiver) and Response (format of response).

> **[v3]**
> # CONTEXT #
> I want to create definitions for word meanings by examining the contextual sentences in which the word is used.
>
> # OBJECTIVE #
> The objective is to produce a definition which identifies the meaning of "<*word*>" in the sentence <*text*>. The answer must be both descriptive and concise.
>
> # STYLE #
> Follow the writing style of WordNet glosses.
>
> # TONE #
> Formal.

[15] a brainchild of *GovTech Singapore's Data Science & AI* team, is a handy template for structuring prompts. See Medium article for more details.

	WSD_σ					DG_σ				
Model	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1	36.00	43.35	19.85	53.75	34.13	62.27	61.15	39.04	69.07	57.16
Phi-3-mini-128k-instruct	73.53	69.31	52.84	69.36	68.06	50.88	47.32	37.53	50.57	47.35
stablelm-2-1_6b-chat	55.18	53.19	32.68	63.00	50.17	65.88	63.76	41.46	72.25	60.34
h2o-danube2-1.8b-chat	56.53	63.56	39.04	64.16	53.83	66.11	62.72	41.34	71.76	60.29
Llama-2-7b-chat-hf	59.93	60.73	42.73	69.07	56.55	63.67	61.67	37.16	67.91	57.57
Meta-Llama-3-8B-Instruct	64.88	63.66	40.43	68.49	59.32	66.79	65.34	42.31	66.79	61.17
falcon-7b-instruct	30.53	34.45	16.82	47.10	28.71	66.25	64.39	40.01	69.36	60.18
Mistral-7B-Instruct-v0.2	71.74	70.47	51.99	68.20	66.91	69.46	69.28	43.46	67.91	63.04
vicuna-7b-v1.5	68.72	69.31	43.88	67.63	63.09	67.07	66.70	42.07	70.23	61.47
GPT-3.5-turbo	75.65	75.18	55.69	74.85	71.00	71.16	72.56	46.18	72.83	65.73
GPT-4-turbo	83.67	83.87	65.00	81.79	79.36	70.58	68.79	45.33	72.25	64.67
Mean	61.48	62.46	41.90	66.12	57.37	65.46	63.97	41.44	68.26	59.90

Table 2: F1 scores obtained by models when evaluated against our disambiguation benchmark for LLMs in the *zero-shot* scenario (*All-MiniLM-L6-v2* is the sentence embedder used for the generation setting). Source: (Martelli et al., 2024)

> # AUDIENCE #
> Artificial Intelligence and Data Scientists.
>
> # RESPONSE #
> The generated definition of the word, without any other preamble text. It must not exceed two sentence.

Prompt additions In addition to the previously described prompts (v2 and v3), we experimented with strategies that involve augmenting the original prompts with additional sentences. According to the literature, such augmentations can potentially enhance performance. Specifically, we employed four different paradigms: (i) *Zero-shot Chain-of-Thought* prompting (Kojima et al., 2023), which involves instructing the model to "Think step by step" to encourage a reasoning process; (ii) *Reflection pattern* (White et al., 2023), where the model is prompted to explain the rationale and assumptions behind its answers; (iii) *Cognitive Verifier pattern* (White et al., 2023), inspired by research suggesting that breaking down a question into smaller parts and then combining the answers can improve reasoning (Zhou et al., 2023); (iv) *EmotionPrompt* technique (Li et al., 2023), which explores the LLMs' ability to understand and harness emotional stimuli by prompting them to provide a confidence score for their answers.

	WSD_σ					DG_σ				
Model	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1	65.72	73.92	50.36	80.63	64.01	60.30	63.45	37.46	66.47	59.36
Phi-3-mini-128k-instruct	74.67	72.25	52.17	70.52	69.03	66.55	65.96	43.03	73.12	61.43
stablelm-2-1_6b-chat	65.37	66.07	44.24	69.65	60.85	66.51	63.35	42.19	69.95	60.70
h2o-danube2-1.8b-chat	67.16	72.35	50.60	75.43	64.46	67.18	63.03	39.40	69.36	60.41
Llama-2-7b-chat-hf	67.41	67.95	45.46	73.12	62.76	64.00	65.23	40.31	71.38	59.12
Meta-Llama-3-8B-Instruct	70.86	74.65	50.78	76.87	67.07	67.69	67.53	42.25	69.36	61.96
falcon-7b-instruct	59.34	66.17	42.79	72.83	57.12	68.37	65.75	41.10	67.34	61.76
Mistral-7B-Instruct-v0.2	75.06	74.86	55.62	74.27	70.57	69.06	65.86	44.18	69.65	63.08
vicuna-7b-v1.5	72.23	74.55	52.54	78.03	68.33	68.00	68.16	41.94	68.49	62.11
GPT-3.5-turbo	76.90	77.48	57.26	73.41	72.34	71.90	71.09	47.03	78.03	66.42
GPT-4-turbo	83.60	82.72	66.40	82.08	79.49	74.88	72.25	50.66	74.85	69.00
Mean	70.75	72.99	51.65	75.16	66.91	67.67	66.51	42.68	70.72	62.30

Table 3: F1 scores obtained by models when evaluated against our disambiguation benchmark for LLMs in the *one-shot* scenario (*All-MiniLM-L6-v2* is the sentence embedder used for DG_σ setting). Source: (Martelli et al., 2024)

4.5 Results

Overall, baseline performance proves challenging to surpass, particularly for non-commercial LLMs. Specifically, the random baseline achieves an F1 score of 17.01 in both WSD_σ and DG_σ settings, and 50.00 in WiC_σ. In contrast, the Most Frequent Sense (MFS) baseline scores 65.18 in both WSD_σ and DG_σ.

Among all models, GPT-4-turbo, which features the highest number of parameters, is the best-performing model. Among the non-commercial models, Mistral-7B-Instruct-v0.2 and the smaller Phi-3-mini-128k-instruct stand out as the leading performers. We present the results for WSD_σ and DG_σ across zero-shot, one-shot, and few-shot settings in Tables 2, 3, and 4, respectively. The results for WiC_σ are shown in Table 5.

Across our evaluation settings, WiC_σ appears to be the most challenging, with the best-performing non-commercial models barely surpassing the random baseline. Interestingly, WSD_σ is the setting where the presence of annotated examples in prompts proves most beneficial. Analyzing the mean values across all parts of speech in WSD_σ, we observe a notable increasing trend: 57.37 in zero-shot, 66.91 in one-shot, and 67.87 in few-shot. This improvement with annotated examples is not seen in DG_σ, where we observe a decrease in average performance from one-shot to few-shot. Similarly, in WiC_σ, performance peaks in the one-shot setting with an average score of 56.73.

Consistent with other studies (Campolungo et al., 2022; Barba et al., 2021c; Loureiro et al., 2021), we find that verbs are more challenging than nouns, while adverbs perform

the best. This is likely due to the highly polysemous nature of verbs and the relatively lower ambiguity of adverbs.

4.5.1 WSD_σ

We observe a significant performance gap among models, with the lowest score of 28.71 reported by falcon-7b-instruct in the zero-shot setting and the highest at 79.49 achieved by GPT-4-turbo in the one-shot setting. This performance gap narrows significantly between the one-shot and few-shot settings.
Notably, falcon-7b-instruct shows the greatest benefit from the addition of examples in the prompt, reaching an F1 score of 60.22 in the few-shot scenario. Unexpectedly, some models, such as Mistral-7B-Instruct-v0.2, show little to no improvement from enriched prompts with additional annotated instances. Generally, the most significant performance increase is observed from zero-shot to one-shot, with only limited improvement from one-shot to few-shot.
Interestingly, smaller models challenge their larger counterparts with remarkable performances. For instance, almost all smaller models outperform falcon-7b-instruct across all sub-settings (see Tables 2, 3, 4).

Impact of MFS annotated examples Building on research that examines the "Most Frequent Sense Bias" phenomenon (Maru et al., 2022), where WSD systems tend to predict the most frequent sense, we conducted a preliminary empirical study indicating that the choice of annotated instances impacts performance. This led us to analyze whether including annotated MFS target words in one-shot and few-shot prompts biases the system towards predicting the most frequent senses. Preliminary findings suggest that this is generally the case, with only a few outliers where the outcome remains unaffected. However, the underlying mechanisms are not yet fully understood, and further studies are required to clarify these observations.

4.5.2 DG_σ

Unlike WSD_σ, here, we do not observe significant performance gaps between sub-settings. In all DG_σ sub-settings, all open-source models do not outperform the MFS baseline (see Table 2, 3, 4). The overall performance in this setting depends significantly on the ability of sentence embedders to capture the semantics of the generated definitions, potentially leading to errors not attributable to the language models themselves. Preliminary manual error analysis indicates that some misclassifications result from the sentence embedders' inability to understand the semantics of the generated definitions. Future work will focus on adopting more efficient commercial sentence embedders, which were not utilized in this study due to financial constraints.

	WSD_σ					DG_σ				
Model	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1	67.48	74.13	50.00	80.34	64.99	62.90	65.86	37.28	69.65	57.78
Phi-3-mini-128k-instruct	73.65	71.30	51.99	70.80	68.27	65.13	66.80	43.64	70.23	60.70
stablelm-2-1_6b-chat	67.62	71.72	47.51	73.12	63.84	66.04	65.34	41.52	69.94	60.55
h2o-danube2-1.8b-chat	68.02	72.56	48.48	72.83	64.40	67.72	64.39	40.55	69.07	61.16
Llama-2-7b-chat-hf	70.60	72.35	49.93	80.05	66.57	62.48	64.81	38.98	68.20	57.71
Meta-Llama-3-8B-Instruct	72.88	75.07	52.30	78.61	68.75	66.41	64.71	40.73	69.07	60.47
falcon-7b-instruct	62.60	68.48	45.94	76.01	60.22	66.81	64.60	39.70	68.20	60.41
Mistral-7B-Instruct-v0.2	74.11	74.76	55.81	76.30	70.13	66.18	65.23	41.52	70.53	60.65
vicuna-7b-v1.5	72.16	74.97	53.57	77.74	68.56	66.46	66.91	41.94	70.80	61.14
GPT-3.5-turbo	76.06	75.60	57.38	73.41	71.62	71.65	71.41	43.28	75.43	65.33
GPT-4-turbo	83.48	82.40	66.22	79.47	79.22	74.25	72.87	50.54	77.45	68.88
Mean	71.69	73.94	52.64	76.24	67.87	66.91	66.63	41.78	70.77	61.34

Table 4: F1 scores obtained by models when evaluated against our disambiguation benchmark for LLMs in the *few-shot* scenario (*All-MiniLM-L6-v2* is the sentence embedder used for DG_σ setting). Source: (Martelli et al., 2024)

4.5.3 WiC_σ

Consistent with the findings of Brown et al. (2020a), our results indicate that models continue to struggle with this task, achieving an average F1 score across all sub-settings that is comparable to the random baseline. On average, we observe an improvement in performance in the one-shot setting; however, performance unexpectedly decreases in the few-shot setting. Notably, GPT-3.5-turbo and GPT-4-turbo exhibit significant performance drops of 10.7 and 33.40, respectively, in the few-shot scenario.

4.5.4 Finetuning

We investigate the impact of fine-tuning on the disambiguation capabilities of LLMs. To this end, we fine-tuned four LLMs: Meta-Llama-3-8B-Instruct, Mistral-7B-Instruct-v0.2, TinyLlama-1.1B-Chat-v1.0, and Phi-3-mini-128k-instruct.
Does finetuning help boosting performances? Our observations indicate that models benefit significantly from fine-tuning, with an average performance increase of 6.35 F1 points across all tasks and sub-settings. The evaluation setting that gains the most from fine-tuning is DG_σ, with an average improvement of 10.15 F1 points. Notably, the sharpest performance increase, 31.01 F1 points, is observed in WSD_σ after fine-tuning TinyLlama-1.1B-Chat-v1.0. Across all sub-settings and tasks, the highest improvement is seen in the zero-shot setting, with an average increase of 14.22 F1 points. This is expected since models are fine-tuned with "zero-shot" instances. Surprisingly, the only instance where fine-tuning does not lead to a performance increase is in the WiC_σ zero-shot setting, where TinyLlama-1.1B-Chat-v1.0

	zero-shot					one-shot					few-shot				
Model	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1	50.00	47.91	49.32	60.00	49.80	49.62	52.08	50.00	50.00	50.10	49.81	52.77	50.00	40.00	50.00
Phi-3-mini-128k-instruct	58.33	56.94	53.35	63.33	56.80	53.21	52.08	53.35	40.00	52.70	55.30	52.70	53.35	40.00	53.90
stablelm-2-1_6b-chat	51.89	47.91	54.02	66.66	52.40	50.94	52.08	50.33	40.00	50.60	47.91	47.91	51.67	50.00	49.10
h2o-danube2-1.8b-chat	53.21	52.77	50.33	36.66	51.80	54.54	54.16	55.03	36.66	54.10	50.56	54.16	52.01	40.00	51.20
Llama-2-7b-chat-hf	21.78	23.61	20.13	23.33	21.60	56.06	52.08	53.02	43.33	54.20	50.94	52.08	48.99	56.66	50.70
Meta-Llama-3-8B-Instruct	49.43	52.08	52.68	60.00	51.10	48.67	43.75	48.32	56.66	48.10	50.18	47.22	49.64	60.00	49.90
falcon-7b-instruct	21.04	15.97	21.14	33.33	20.90	52.84	50.00	50.33	63.33	52.00	51.89	49.30	52.68	63.33	52.10
Mistral-7B-Instruct-v0.2	61.93	60.41	59.73	60.00	61.00	63.63	66.60	62.08	50.00	63.20	51.32	50.00	49.32	53.33	50.60
vicuna-7b-v1.5	49.81	52.77	50.00	40.00	50.00	51.32	47.91	51.00	56.66	50.90	50.18	47.22	50.00	60.00	50.00
GPT-3.5-turbo	55.87	56.90	57.04	70.00	56.80	62.68	69.44	63.42	50.00	63.50	52.84	58.33	52.01	33.33	52.80
GPT-4-turbo	80.87	78.47	83.55	83.33	81.40	87.50	81.94	81.20	83.33	84.70	52.27	50	49.32	60	51.30
Mean	50.37	49.61	50.11	54.24	50.32	57.36	56.55	56.18	51.81	56.73	51.19	51.06	50.81	50.60	51.05

Table 5: F1 scores obtained by models when evaluated against our disambiguation benchmark for LLMs in WiC_σ setting. Source: (Martelli et al., 2024)

		WSD_σ					DG_σ					WiC_σ				
Model		**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v1.0	*zero-shot*	67.45	77.45	50.42	80.92	65.14	72.09	71.62	50.66	77.74	67.42	40.53	46.52	39.26	33.33	40.80
	one-shot	66.27	73.61	49.69	80.63	64.15	69.32	71.51	51.51	74.56	65.80	55.68	43.75	51.34	56.66	52.70
	few-shot	67.23	74.34	49.87	80.92	64.86	68.74	72.04	51.39	73.41	65.44	50.00	52.77	50.00	40.00	50.10
Phi-3-mini-128k-instruct	*zero-shot*	71.88	75.49	55.75	77.45	68.95	73.72	72.67	56.17	78.32	69.80	77.84	77.77	74.16	60.00	76.20
	one-shot	70.60	74.13	52.66	78.03	67.33	72.27	72.56	54.84	75.14	68.48	57.00	52.77	56.04	56.66	56.10
	few-shot	69.18	74.45	51.51	77.45	66.24	70.34	69.52	52.42	76.30	66.44	51.89	50.00	48.32	56.66	50.70
Meta-Llama-3-8B-Instruct	*zero-shot*	77.23	78.74	58.17	83.52	73.39	73.27	75.49	59.50	81.21	70.81	65.71	56.94	61.74	63.33	63.20
	one-shot	72.58	76.12	54.53	80.05	69.29	72.99	74.76	58.05	80.63	70.19	53.59	52.77	53.69	63.33	53.80
	few-shot	72.72	75.18	54.41	79.19	69.18	72.86	73.08	58.11	81.50	69.94	52.65	57.63	53.35	50.00	53.50
Mistral-7B-Instruct-v0.2	*zero-shot*	76.41	77.27	57.50	82.36	72.50	74.44	74.03	59.32	81.21	71.26	74.05	65.97	70.80	53.33	71.30
	one-shot	52.25	62.30	40.85	73.98	52.01	73.39	74.34	57.80	82.08	70.38	72.53	68.05	68.12	63.33	70.30
	few-shot	57.32	65.54	40.07	69.65	55.06	73.81	74.24	57.14	82.65	70.49	56.81	51.38	53.35	63.33	55.20
Mean		68.42	73.71	51.28	78.67	65.67	72.27	72.98	55.58	78.72	68.87	59.02	56.36	56.68	55.00	57.82

Table 6: F1 scores obtained by *finetuned* models when evaluated against our benchmark. Source: (Martelli et al., 2024)

experiences a decrease of 9 F1 points compared to its not finetuned performance. Results obtained by models after finetuning are reported in Table 6.

We analyzed and discussed the results of LLMantics benchmark for both fine-tuned and non-fine-tuned models across all evaluation settings. Our findings suggest that LLMs can partially disambiguate words in context: only commercial models consistently outperform the MFS and random baselines. Despite the limited exploration of fine-tuning (using default hyperparameters without adjustments and applying it to a small number of LLMs), our results indicate that it generally enhances the performance of all tested models.
DG_σ evaluation setting, with its inherent issues, warrants further review. The LLMantics benchmark aims to accommodate a wide range of models and evaluation settings, striving to be as comprehensive as possible. Future expansions could include additional semantic-related tasks. For example, in Section 5, we employed the new coarse-grained resources from Proietti et al. (2024) to evaluate LLMs on the

Homonymy Disambiguation task, thereby laying the foundation for what in the future might be the HD_σ evaluation setting.

5 Homonymy Disambiguation with Large Language Models

Homonymy Disambiguation is the task of disambiguating coarse-grained senses, namely homonyms. With the development of a coarse-grained resource (Proietti et al., 2024) and a Large Language Model semantic benchmark (Martelli et al., 2024), it was logical to evaluate LLMs on this task to assess their capabilities in handling homonymy disambiguation. We did not finetune any LLM in this sets of experiments. In the following sections, we discuss the datasets used, the models involved, and the resulting findings.

5.1 Experimental setup

Data We are unable to use the ALL test set (Raganato et al., 2017) in our evaluation, as in LLMantics, because it does not provide the coarse-grained sense distinctions required for HD as the *test* split of Proietti et al. (2024) (see Figure 13 in Section 3.2). Therefore, the *test* dataset, consisting of 17.206 instances, was modified to be compatible with the LLMantics pipeline, as the original data structures and fields differed from those used in LLMantics. In Table 7 there is an overview of all test datasets with some statistics. We report the number of instances, the average degree of ambiguity for each instance (which indicates the difficulty level of the dataset) and the accuracy of the Most Frequent Sense and random baselines. The MFS baseline always predicts the first sense of the target word, while the random baseline selects a sense randomly. Also these metrics provide an indication of the testing set's difficulty.

We have datasets that are used exclusively for the WSD task, others only for HD, and the *test* set Proietti et al. (2024) used for both tasks. We also report the statistics for the ALL dataset to facilitate a comparison with *test*. Despite differences in the number of instances, the characteristics of the datasets are quite similar, except for the MFS score. It appears that the *test* set contains a higher proportion of instances where the most frequent sense is the gold standard. This can be problematic for evaluating disambiguation systems, as models that always predict the first sense might achieve disproportionately high results, which does not accurately reflect their performance. The *test* dataset includes several subsets (see Section 3.2 for more details): $test_{FGA}$ (14.172 instances), which retains only words with at least one fine-grained candidate; $test_{HA}$ (2.265 instances), which excludes instances with only one homonymy cluster candidate; and $test_{HA_p}$ (609 instances), a subset of $test_{HA}$ that includes only items suitable for comparison with feature extraction methods (see Section 3.3 for more details). As evident, $test_{FGA}$ and $test_{HA}$ exhibit higher degrees of ambiguity and lower baseline scores compared to *test* for WSD task and HD task, respectively. This is because we excluded instances where the sense choice was unique.

Dataset	Task	*Number instances*	*Ambiguity degree*	*MFS baseline*	*Random baseline*
ALL	WSD	7253	5.87	65.18	38.49
test	WSD	17206	6.05	74.55	36.66
	HD	17206	1.18	98.49	92.86
test$_{FGA}$	WSD	14172	7.13	69.10	22.91
test$_{HA}$	HD	2265	2.37	88.56	47.28
test$_{HA_p}$	HD	609	2.06	81.11	48.44

Table 7: Statistics of test datasets employed in experiments.

Models We did not fine-tune the two selected Large Language Models, instead, we used them only for text generation inference. These models were chosen because they are ones of the latest available: **Meta-Llama-3-8B-Instruct**[16] by Meta (released in April 2024) and **gemma-2-9b-it**[17] by Google (released in June 2024). Both models have a comparable number of parameters, 8 billion and 9 billion, respectively. The latter model is not yet included in the LLMantics benchmark but will be added soon. Given the dynamic nature of the LLM field, where new models and features are continuously developed, LLMantics is designed to be adaptable, allowing the integration of new models and evaluation settings, as previously discussed. The hyperparameters used are identical to those described in LLMantics (Section 4.2). The only design choice that requires discussion pertains to the creation of prompts.

Prompts Building on the experiments conducted by Proietti et al. (2024) with pre-trained Language Models (particularly BERT, which was the best performer), my aim is to test the Llama-3-8b and gemma-9b LLMs in both WSD$_\sigma$ and HD$_\sigma$ settings. Additionally, as described in Proietti et al. (2024), we plan to evaluate how an LLM that predicts fine-grained senses (WSD system) performs in predicting coarse-grained ones (HD$_\sigma$ setting). In the context of LLMs, a WSD system uses prompts designed to output fine-grained senses. By leveraging the newly created resource, we can map these fine-grained senses to their corresponding homonyms. For instance, if the LLM predicts the fine-grained sense "*bound.a.01*" ("confined by bonds") for the word *bound*, the HD system will map this to the homonym "*bound.a.h.02*".
The prompt used for WSD task is the same as the one employed in LLMantics. However, to enable the disambiguation of homonyms (HD), some modifications are made. Table 8 presents examples of WSD and HD prompts along with their corresponding outputs. In WSD prompt, the model outputs the definition of the

[16] `https://huggingface.co/meta-llama/Meta-Llama-3-8B`
[17] `https://huggingface.co/google/gemma-2-9b-it`

Task	Example Prompt	Example Output
WSD	**Question**: given the following sentence: *The thing to lose sleep over is what people, having concluded that we are weaker than we are, are likely to do about it. The evidence suggests that foreign peoples believe the United States is weaker than the Soviet Union, and is bound to fall still further behind in the years ahead.* Choose the corresponding definition among: 1) confined by bonds; 2) held with another element, substance or material in chemical or physical union; 3) secured with a cover or binding; often used as a combining form; 4) (usually followed by 'to') governed by fate; 5) covered or wrapped with a bandage; 6) headed or intending to head in a certain direction; often used as a combining form as in 'college-bound students'; 7) bound by an oath; 8) bound by contract; 9) confined in the bowels; Answer by reporting the corresponding definition and do not motivate your answer.	**Answer**: 4) (usually followed by 'to') governed by fate.
HD	**Question**: given the following sentence: *The thing to lose sleep over is what people, having concluded that we are weaker than we are, are likely to do about it. The evidence suggests that foreign peoples believe the United States is weaker than the Soviet Union, and is bound to fall still further behind in the years ahead.* Each homonym cluster is identified by a key and a set of definitions. Choose the corresponding homonym key among: **bound.a.h.02** : ["confined by bonds", "held with another element, substance or material in chemical or physical union", "secured with a cover or binding; often used as a combining form", "(usually followed by 'to') governed by fate", "covered or wrapped with a bandage", "bound by an oath", "bound by contract", "confined in the bowels"]. **bound.a.h.01** : ["headed or intending to head in a certain direction; often used as a combining form as in 'college-bound students'"]. Answer by reporting the corresponding homonym key and do not motivate your answer.	**Answer**: bound.a.h.02.

Table 8: Prompt examples with their corresponding output for WSD and HD tasks referring to the same instance.

predicted sense. In contrast, for the HD one, we ask the system to output the key of the homonym. The latter is represented in the input prompt by the concatenation of all fine-grained sense definitions that constitute it.

5.2 Results

To summarize, we evaluate two of the most recent and powerful open-source LLMs in three different modalities:

- **WSD$_\sigma$ setting**: using WSD prompts to perform word sense disambiguation.
- **HD$_\sigma$ via WSD**: mapping the outputs of the WSD system to homonymy clusters for evaluation in the HD$_\sigma$ setting.
- **HD$_\sigma$ setting**: using HD prompts to disambiguate coarse-grained senses.

For each prompt type (WSD and HD), we also incorporate zero-shot, one-shot, and few-shot variations. These variations include prompts containing zero, one, or three

			Total		FGA	HA	HA_p
Model		**System**	WSD_σ	HD_σ	WSD_σ	HD_σ	HD_σ
BERT		WSD	81.77	99.23	78.13	94.13	96.39
		HD	–	99.16	–	93.64	96.06
Llama-3-8B-Instruct	*zero-shot*	WSD	63.34	99.41	55.49	95.58	92.77
		HD	–	98.82	–	91.08	83.41
Llama-3-8B-Instruct	*one-shot*	WSD	70.45	99.52	64.13	96.37	94.08
		HD	–	99.27	–	94.52	90.64
Llama-3-8B-Instruct	*few-shot*	WSD	72.49	99.55	66.61	96.64	94.41
		HD	–	98.80	–	90.94	84.89
gemma-2-9b-it	*zero-shot*	WSD	66.72	99.03	59.59	92.71	88.99
		HD	–	98.65	–	89.75	84.07
gemma-2-9b-it	*one-shot*	WSD	74.36	99.62	68.87	97.17	95.73
		HD	–	99.25	–	94.30	89.81
gemma-2-9b-it	*few-shot*	WSD	73.50	99.59	67.83	96.95	95.56
		HD	–	99.35	–	95.09	89.98

Table 9: Accuracy scores of BERT, Llama-3-8b and gemma-9b in WSD_σ and HD_σ evaluation settings on different test sets.

human-annotated examples, respectively. The datasets used for these evaluations include the complete *test* set and its previously introduced subsets.

The results for BERT, as presented in the first row, are entirely sourced from Proietti et al. (2024) (see Figure 15). It is immediately evident that BERT, with approximately 300 million parameters, significantly outperforms Llama-3-8b and gemma-9b, which have 8 and 9 billion parameters, respectively, in the WSD_σ setting. As illustrated in Tables 2, 3 and 4, it is likely that commercial LLMs such as GPT-3.5 and GPT-4 would yield comparable results with respect to BERT. However, due to financial constraints, we were unable to test these models with this data. These results align with the findings of Kocoń et al. (2023) and LLMantics (Martelli et al., 2024), namely that smaller but task-specialized models (in this case in WSD and HD) perform better than huge general-purpose models in text generation such as LLMs.

However, the scenario changes with the HD_σ setting: BERT is almost always outperformed by Llama-3-8b and gemma-9b in both the *test* and $test_{HA}$ datasets. The best performing model in HD_σ is gemma-9b using the HD_σ via WSD approach with one-shot prompt, achieving an accuracy of 99.62 in *test* and 97.17 in $test_{HA}$. Regarding $test_{HA_p}$, the score of 95.24 reached by the distance-based method (see Figure 14) is only surpassed by the fine-tuned BERT and gemma-9b models, though not by much. This observation is significant, as the computational cost of the distance-based method is much lower, yet it achieves a high accuracy score.

It should be noted that, as previously found by Proietti et al. (2024), the HD_σ via WSD prediction modality consistently outperforms direct homonym predictions.

Additionally, regarding LLMs, there is a performance gain when transitioning from zero-shot prompts to one- and few-shot ones.

6 Conclusions

In this thesis, it has been explored the critical importance of WSD and its pivotal role in enhancing the performance of various downstream applications such as Machine Translation and Information Retrieval. Despite significant advancements, WSD remains a challenging task due to the intrinsic complexity of word meanings and the fine granularity required to distinguish between them.

We reviewed various approaches to WSD, ranging from traditional knowledge-based methods to modern neural networks and pre-trained language models. Each method brings unique strengths and addresses specific facets of the WSD problem. However, challenges such as the scarcity of annotated data, the issue of fine-granularity, and the need for comprehensive evaluation benchmarks continue to impede progress.

In response to these challenges, we presented two significant contributions to the field. First, we developed a new coarse-grained resource aimed at simplifying sense granularity, making it more manageable for both humans and machines (Proietti et al., 2024). This resource is expected to enhance the applicability of WSD systems in practical scenarios by reducing the ambiguity of sense distinctions. Second, we introduced the LLMantics benchmark (Martelli et al., 2024), designed to evaluate the performance of LLMs in semantic tasks, including WSD. This benchmark is adaptable and constantly evolving, allowing the integration of new models and evaluation settings. Looking ahead, I plan to enhance LLMantics with additional LLMs and improve the Definition Generation evaluation setting (DG_σ), as it currently produces misclassifications due to limitations in the sentence embedders used, a well-known issue in the research community (Bevilacqua et al., 2020). Furthermore, integrating the Homonymy Disambiguation setting or adding other semantic-related evaluation settings like Lexical Substitution could further improve the benchmark's comprehensiveness and utility. These contributions aim to push the boundaries of current WSD research and provide robust tools for the community.

I would like to extend my gratitude to Professor Navigli for providing me with the opportunity of collaborating to such projects, partly due to the research grant I won, and to everyone at SapienzaNLP for their guidance and support over the past year. Their advice has been invaluable in helping me improve. I hope this dissertation has provided a comprehensive overview of the WSD landscape, highlighting its limitations and potential improvements, and I hope to continue contributing to WSD research with future works.

May the semantics be with you!

References

Abdin, M. (2024). Phi-3 technical report: A highly capable language model locally on your phone.

Agirre, E., de Lacalle, O. L., and Soroa, A. (2014). Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. (2023). Gqa: Training generalized multi-query transformer models from multi-head checkpoints.

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In Bender, E. M., Derczynski, L., and Isabelle, P., editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Barba, E., Pasini, T., and Navigli, R. (2021a). ESC: Redesigning WSD with extractive sense comprehension. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Barba, E., Procopio, L., Lacerra, C., Pasini, T., and Navigli, R. (2021b). Exemplification modeling: Can you give me an example, please? In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3779–3785. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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

Bellagente, M., Tow, J., Mahan, D., Phung, D., Zhuravinskyi, M., Adithyan, R., Baicoianu, J., Brooks, B., Cooper, N., Datta, A., Lee, M., Mostaque, E., Pieler, M., Pinnaparju, N., Rocha, P., Saini, H., Teufel, H., Zanichelli, N., and Riquelme, C. (2024). Stable lm 2 1.6b technical report.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer.

Bevilacqua, M., Maru, M., and Navigli, R. (2020). Generationary or "how we went beyond word sense inventories and learned to gloss". In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Bevilacqua, M. and Navigli, R. (2020). Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Bevilacqua, M., Pasini, T., Raganato, A., and Navigli, R. (2021). Recent trends in word sense disambiguation: A survey. In Zhou, Z.-H., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Blevins, T. and Zettlemoyer, L. (2020). Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020a). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020b). Language models are few-shot learners.

Bsharat, S. M., Myrzakhan, A., and Shen, Z. (2024). Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. (2016). Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64.

Campolungo, N., Martelli, F., Saina, F., and Navigli, R. (2022). DiBiMT: A novel benchmark for measuring Word Sense Disambiguation biases in Machine Translation. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4352, Dublin, Ireland. Association for Computational Linguistics.

Carpuat, M. and Wu, D. (2007). Improving statistical machine translation using word sense disambiguation. In Eisner, J., editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic. Association for Computational Linguistics.

Chia, Y. K., Hong, P., Bing, L., and Poria, S. (2023). Instructeval: Towards holistic evaluation of instruction-tuned large language models. *arXiv preprint arXiv:2306.04757*.

Chklovski, T. and Mihalcea, R. (2003). Exploiting agreement and disagreement of human annotators for word sense disambiguation.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators.

Conia, S. and Navigli, R. (2021). Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration. In Merlo, P., Tiedemann, J., and Tsarfaty, R., editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3269–3275, Online. Association for Computational Linguistics.

Dao, T., Fu, D., Ermon, S., Rudra, A., and Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Edmonds, P. and Cotton, S. (2001). SENSEVAL-2: Overview. In Preiss, J. and Yarowsky, D., editors, *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Gale, W., Church, K. W., and Yarowsky, D. (1992a). Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Newark, Delaware, USA. Association for Computational Linguistics.

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

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. (2023). A framework for few-shot language model evaluation.

Hadiwinoto, C., Ng, H. T., and Gan, W. C. (2019). Improved word sense disambiguation using pre-trained contextualized word representations. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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 5297–5306, Hong Kong, China. Association for Computational Linguistics.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning.

He, P., Liu, X., Gao, J., and Chen, W. (2021). Deberta: Decoding-enhanced bert with disentangled attention.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2020). Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Huang, L., Sun, C., Qiu, X., and Huang, X. (2019). GlossBERT: BERT for word sense disambiguation with gloss knowledge. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

Jawahar, G., Sagot, B., and Seddah, D. (2019). What does BERT learn about the structure of language? In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Kilgarriff, A. (1997). I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., Bielaniewicz, J., Gruza, M., Janz, A., Kanclerz, K., Kocoń, A., Koptyra, B., Mieleszczenko-Kowszewicz, W., Miłkowski, P., Oleksy, M., Piasecki, M., Łukasz Radliński, Wojtasik, K., Woźniak, S., and Kazienko, P. (2023). Chatgpt: Jack of all trades, master of none. *Information Fusion*, 99:101861.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2023). Large language models are zero-shot reasoners.

Kong, C., Chen, Y., Zhang, H., Yang, L., and Yang, E. (2022). Multitasking framework for unsupervised simple definition generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5934–5943.

Lacerra, C., Bevilacqua, M., Pasini, T., and Navigli, R. (2020). Csi: A coarse sense inventory for 85 *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8123–8130.

Levine, Y., Lenz, B., Dagan, O., Ram, O., Padnos, D., Sharir, O., Shalev-Shwartz, S., Shashua, A., and Shoham, Y. (2020). SenseBERT: Driving some sense into BERT. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667, Online. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Li, C., Wang, J., Zhang, Y., Zhu, K., Hou, W., Lian, J., Luo, F., Yang, Q., and Xie, X. (2023). Large language models understand and can be enhanced by emotional stimuli.

Li, J., Bao, Y., Huang, S., Dai, X., and Chen, J. (2020). Explicit semantic decomposition for definition generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 708–717.

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

Loureiro, D. and Jorge, A. (2019). Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Loureiro, D., Rezaee, K., Pilehvar, M. T., and Camacho-Collados, J. (2021). Analysis and evaluation of language models for word sense disambiguation.

Luan, Y., Hauer, B., Mou, L., and Kondrak, G. (2020). Improving word sense disambiguation with translations. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4055–4065, Online. Association for Computational Linguistics.

Mallery, J. C. (1988). Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers.

Martelli, F., Kalach, N., Tola, G., Navigli, R., et al. (2021). Semeval-2021 task 2: Multilingual and cross-lingual word-in-context disambiguation (mcl-wic). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 24–36.

Martelli, F., Lavalle, L., and Navigli, R. (2024). Llmantics: A novel benchmark for measuring lexical semantic capabilities of instruction-tuned large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. under revision.

Maru, M., Conia, S., Bevilacqua, M., and Navigli, R. (2022). Nibbling at the hard core of Word Sense Disambiguation. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4724–4737, Dublin, Ireland. Association for Computational Linguistics.

Maudslay, R. H. and Teufel, S. (2022). Homonymy information for English WordNet. In Kernerman, I. and Krek, S., editors, *Proceedings of Globalex Workshop on Linked Lexicography within the 13th Language Resources and Evaluation Conference*, pages 90–98, Marseille, France. European Language Resources Association.

McCarthy, D. and Navigli, R. (2007a). SemEval-2007 task 10: English lexical substitution task. In Agirre, E., Màrquez, L., and Wicentowski, R., editors, *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

McCarthy, D. and Navigli, R. (2007b). SemEval-2007 task 10: English lexical substitution task. In Agirre, E., Màrquez, L., and Wicentowski, R., editors, *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (2008). Wordnet: An on-line lexical database. *Communications of the ACM*, 38.

Miller, G. A., Leacock, C., Tengi, R., and Bunker, R. T. (1993). A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA. Association for Computational Linguistics.

Moro, A. and Navigli, R. (2015). SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In Nakov, P., Zesch, T., Cer, D., and Jurgens, D., editors, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Nair, S., Srinivasan, M., and Meylan, S. (2020). Contextualized word embeddings encode aspects of human-like word sense knowledge.

Navigli, R. (2006). Meaningful clustering of senses helps boost word sense disambiguation performance. In Calzolari, N., Cardie, C., and Isabelle, P., editors, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 105–112, Sydney, Australia. Association for Computational Linguistics.

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

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

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

Navigli, R. and Ponzetto, S. P. (2010). BabelNet: Building a very large multilingual semantic network. In Hajič, J., Carberry, S., Clark, S., and Nivre, J., editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225, Uppsala, Sweden. Association for Computational Linguistics.

Neale, S., Gomes, L., Agirre, E., de Lacalle, O. L., and Branco, A. (2016). Word sense-aware machine translation: Including senses as contextual features for improved translation models. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2777–2783, Portorož, Slovenia. European Language Resources Association (ELRA).

Ng, H. T. and Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, Santa Cruz, California, USA. Association for Computational Linguistics.

Ng, H. T., Lim, C. Y., and Foo, S. K. (1999). A case study on inter-annotator agreement for word sense disambiguation. In *SIGLEX99: Standardizing Lexical Resources*.

Noraset, T., Liang, C., Birnbaum, L., and Downey, D. (2016). Definition modeling: Learning to define word embeddings in natural language.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Pasini, T. (2020). The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4936–4942. International Joint Conferences on Artificial Intelligence Organization. Survey track.

Pasini, T. and Camacho-Collados, J. (2020). A short survey on sense-annotated corpora. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5759–5765, Marseille, France. European Language Resources Association.

Pasini, T., Raganato, A., and Navigli, R. (2021). Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13648–13656.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In Walker, M., Ji, H., and Stent, A., editors, *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, Louisiana. Association for Computational Linguistics.

Pilehvar, M. T. and Camacho-Collados, J. (2019). WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In Burstein, J., Doran, C., and Solorio, T., editors, *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, Minnesota. Association for Computational Linguistics.

Pilehvar, M. T. and Collier, N. (2017). Inducing embeddings for rare and unseen words by leveraging lexical resources. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 388–393, Valencia, Spain. Association for Computational Linguistics.

Pradhan, S., Loper, E., Dligach, D., and Palmer, M. (2007). SemEval-2007 task-17: English lexical sample, SRL and all words. In Agirre, E., Màrquez, L., and Wicentowski, R., editors, *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

Proietti, L., Perrella, S., Tedeschi, S., Vulpis, G., Lavalle, L., Sanchietti, A., Ferrari, A., and Navigli, R. (2024). Analyzing homonymy disambiguation capabilities of pretrained language models. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 924–938, Torino, Italia. ELRA and ICCL.

Pu, X., Pappas, N., Henderson, J., and Popescu-Belis, A. (2018). Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model.

Raganato, A., Camacho-Collados, J., and Navigli, R. (2017). Word sense disambiguation: A unified evaluation framework and empirical comparison. In Lapata, M., Blunsom, P., and Koller, A., editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.

Riccardi, N. and Desai, R. H. (2023). The two word test: A semantic benchmark for large language models. *arXiv preprint arXiv:2306.04610*.

Scozzafava, F., Maru, M., Brignone, F., Torrisi, G., and Navigli, R. (2020). Personalized PageRank with syntagmatic information for multilingual word sense disambiguation. In Celikyilmaz, A. and Wen, T.-H., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–46, Online. Association for Computational Linguistics.

Singer, P., Pfeiffer, P., Babakhin, Y., Jeblick, M., Dhankhar, N., Fodor, G., and Ambati, S. S. (2024). H2o-danube-1.8b technical report.

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

Soanes, C. and Stevenson, A., editors (2003). *Oxford Dictionary of English*. Oxford University Press.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., et al. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Stokoe, C., Oakes, M., and Tait, J. (2003). Word sense disambiguation in information retrieval revisited. pages 159–166.

Su, Y., Zhang, H., Song, Y., and Zhang, T. (2022). Rare and zero-shot word sense disambiguation using Z-reweighting. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4713–4723, Dublin, Ireland. Association for Computational Linguistics.

Tedeschi, S., Bos, J., Declerck, T., Hajič, J., Hershcovich, D., Hovy, E., Koller, A., Krek, S., Schockaert, S., Sennrich, R., Shutova, E., and Navigli, R. (2023). What's the meaning of superhuman performance in today's NLU? In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12471–12491, Toronto, Canada. Association for Computational Linguistics.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

TUGGY, D. (1993). Ambiguity, polysemy, and vagueness. *Cognitive Linguistics*, 4(3):273–290.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 54(433):443–460.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.

Vial, L., Lecouteux, B., and Schwab, D. (2019). Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In Vossen, P. and Fellbaum, C., editors, *Proceedings of the 10th Global Wordnet Conference*, pages 108–117, Wroclaw, Poland. Global Wordnet Association.

Vickrey, D., Biewald, L., Teyssier, M., and Koller, D. (2005). Word-sense disambiguation for machine translation. In Mooney, R., Brew, C., Chien, L.-F., and Kirchhoff, K., editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 771–778, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Voita, E., Sennrich, R., and Titov, I. (2019). The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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, China. Association for Computational Linguistics.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2020). Superglue: A stickier benchmark for general-purpose language understanding systems.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupała, G., and Alishahi, A., editors, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Weaver, W. (1949). Translation. In *Machine Translation of Languages: Fourteen Essays*. MIT Press, Cambridge, MA.

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., and Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Wiedemann, G., Remus, S., Chawla, A., and Biemann, C. (2019). Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings.

Wilks, Y., Fass, D., ming Guo, C., McDonald, J. E., Plate, T. A., and Slator, B. M. (1990). Providing machine tractable dictionary tools. *Machine Translation*, 5:99–154.

Xu, B., Yang, A., Lin, J., Wang, Q., Zhou, C., Zhang, Y., and Mao, Z. (2023). Expertprompting: Instructing large language models to be distinguished experts.

Zhang, P., Zeng, G., Wang, T., and Lu, W. (2024). Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*.

Zhong, Z. and Ng, H. T. (2012). Word sense disambiguation improves information retrieval. In Li, H., Lin, C.-Y., Osborne, M., Lee, G. G., and Park, J. C., editors, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282, Jeju Island, Korea. Association for Computational Linguistics.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., and Chi, E. (2023). Least-to-most prompting enables complex reasoning in large language models.

A Prompt Versions

		zero-shot					one-shot					few-shot				
Model		**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1		35.30	41.78	18.58	54.33	33.25	63.83	72.87	49.87	80.92	62.66	66.70	72.67	49.63	80.34	64.29
	v1.1	35.13	44.71	19.12	56.64	33.77	59.88	69.84	45.46	80.05	58.87	66.39	72.25	48.84	79.47	63.79
	v1.2	43.60	50.36	24.39	66.18	41.19	64.90	72.46	49.39	79.76	63.07	65.81	72.25	47.82	79.76	63.22
	v1.3	34.69	39.79	20.33	51.15	32.88	67.55	74.03	50.30	80.92	65.11	66.02	73.50	49.87	80.63	64.02
Phi-3-mini-128k-instruct		73.53	69.31	52.84	69.36	68.06	74.67	72.25	52.17	70.52	69.03	73.65	71.30	51.99	70.80	68.27
	v1.1	62.60	54.76	41.70	50.80	56.25	72.51	70.78	50.24	72.54	67.21	71.81	72.35	50.24	71.67	66.96
	v1.2	68.81	65.96	49.57	65.60	63.90	74.04	71.93	50.96	69.65	68.30	73.83	72.04	51.39	71.38	68.37
	v1.3	74.76	66.38	48.36	66.76	67.26	75.46	74.13	51.99	71.38	69.75	78.27	78.42	56.65	74.85	73.21
stablelm-2-1_6b-chat		55.18	53.19	32.68	63.00	50.17	65.37	66.07	44.24	69.65	60.85	67.62	71.72	47.51	73.12	63.84
	v1.1	47.88	52.56	29.47	61.84	44.97	63.16	66.70	41.16	67.05	58.80	68.69	72.98	48.12	71.38	64.70
	v1.2	55.58	53.71	31.90	59.24	50.11	64.16	66.07	41.20	66.18	59.25	66.67	67.64	46.00	72.83	62.38
	v1.3	59.60	57.69	34.56	62.71	53.79	63.97	61.88	40.67	65.89	58.48	67.44	68.27	50.12	68.20	63.64
h2o-danube2-1.8b-chat		56.53	63.56	39.04	64.16	53.83	67.16	72.35	50.60	75.43	64.46	68.02	72.56	48.48	72.83	64.40
	v1.1	59.20	60.80	40.49	68.78	56.40	60.32	64.92	44.67	67.34	57.70	68.97	74.03	48.24	73.98	65.15
	v1.2	56.83	64.81	39.34	61.27	54.11	68.27	73.82	50.72	74.27	65.29	67.34	72.98	48.54	73.12	64.08
	v1.3	52.95	56.43	39.28	57.51	50.51	66.51	68.06	50.06	73.41	63.29	70.58	74.97	51.08	77.16	67.03
Llama-2-7b-chat-hf		59.93	60.73	42.73	69.07	56.55	67.41	67.95	45.46	73.12	62.76	70.60	72.35	49.93	80.05	66.57
	v1.1	18.90	22.51	11.74	27.45	18.15	63.26	66.49	43.40	71.09	59.50	69.51	73.50	49.93	77.74	65.97
	v1.2	55.46	61.88	38.19	67.34	52.94	58.83	62.61	41.52	71.96	56.01	67.48	71.26	48.54	76.58	64.09
	v1.3	66.39	66.91	46.48	70.52	62.12	73.00	70.57	50.42	76.87	67.72	73.39	73.92	51.99	80.34	68.92
Meta-Llama-3-8B-Instruct		64.88	63.66	40.43	68.49	59.32	70.86	74.65	50.78	76.87	67.07	72.88	75.07	52.30	78.61	68.75
	v1.1	14.55	14.86	09.98	24.27	14.02	68.95	70.26	47.03	70.23	64.19	71.67	75.49	50.78	74.85	67.57
	v1.2	61.25	66.07	41.46	70.23	57.81	71.62	75.18	52.11	79.19	68.01	72.74	76.23	51.81	78.61	68.71
	v1.3	70.51	68.06	47.88	72.83	65.14	75.09	74.76	54.53	80.05	70.60	76.32	78.21	53.99	80.05	71.66
falcon-7b-instruct		30.53	34.45	16.82	47.10	28.71	59.34	66.17	42.79	72.83	57.12	62.60	68.48	45.94	76.01	60.22
	v1.1	00.11	00.00	00.18	00.00	00.11	58.34	65.86	44.06	67.05	56.50	60.65	67.01	42.79	70.52	57.89
	v1.2	43.37	46.49	35.71	58.09	42.47	62.48	70.57	46.24	78.03	60.59	60.88	67.43	42.67	71.38	58.10
	v1.3	31.48	34.34	15.13	47.39	28.89	59.44	65.02	42.25	74.56	56.98	61.95	68.90	42.97	73.98	59.12
Mistral-7B-Instruct-v0.2		71.74	70.47	51.99	68.20	66.91	75.06	74.86	55.62	74.27	70.57	74.11	74.76	55.81	76.30	70.13
	v1.1	67.62	66.70	46.79	65.60	62.66	74.44	73.71	53.38	71.96	69.43	73.74	75.07	55.26	78.03	69.91
	v1.2	65.79	67.74	48.91	68.20	62.31	75.20	74.55	55.20	74.27	70.52	74.00	75.81	55.26	77.45	70.13
	v1.3	76.72	77.80	58.77	75.14	72.70	78.37	77.80	56.84	74.56	73.21	78.58	78.53	58.53	78.03	73.98
vicuna-7b-v1.5		68.72	69.31	43.88	67.63	63.09	72.23	74.55	52.54	78.03	68.33	72.16	74.97	53.57	77.74	68.56
	v1.1	06.00	07.22	24.81	15.89	05.83	71.09	70.05	50.06	71.67	66.19	71.95	75.07	50.00	77.45	67.62
	v1.2	66.90	70.15	46.48	71.96	62.92	72.25	74.03	52.90	76.58	68.28	72.69	75.60	52.11	79.47	68.71
	v1.3	72.32	69.94	47.76	69.94	66.30	75.34	74.45	53.63	75.43	70.28	76.18	77.90	55.99	79.19	71.95

Table 10: F1 scores obtained by models when evaluated against our disambiguation benchmark for open-source LLMs in WSD_σ setting with different prompts (*v1.1*, *v1.2* and *v1.3*). For having a visual comparison, the first line of results of each model are the ones obtained by prompt version *v1*. Source: (Martelli et al., 2024)

		zero-shot					one-shot					few-shot				
Model		**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1		62.27	61.15	39.04	69.07	57.16	60.30	63.45	37.46	66.47	59.36	62.90	65.86	37.28	69.65	57.78
	v1.1	00.11	00.00	00.00	00.28	00.08	61.30	62.09	36.19	69.36	56.07	59.32	57.59	34.80	64.73	53.77
	v1.2	57.60	58.63	35.04	66.18	53.01	63.18	59.79	36.44	69.36	56.94	62.11	63.66	36.92	69.36	56.92
Phi-3-mini-128k-instruct		50.88	47.32	37.53	50.57	47.35	66.55	65.96	43.03	73.12	61.43	65.13	66.80	43.64	70.23	60.70
	v1.1	67.00	63.76	41.16	71.09	60.88	66.25	68.27	42.37	70.23	61.27	65.41	68.58	43.64	71.38	61.16
	v1.2	54.58	57.59	35.71	64.45	51.15	67.23	68.79	43.28	73.98	62.30	66.88	69.73	42.79	73.41	62.08
stablelm-2-1_6b-chat		65.88	63.76	41.46	72.25	60.34	66.51	63.35	42.19	69.65	60.70	66.04	65.34	41.52	69.94	60.55
	v1.1	58.27	58.01	35.89	68.49	53.63	66.41	65.02	40.37	70.80	60.51	65.88	62.82	40.92	67.34	59.86
	v1.2	65.58	64.18	39.16	68.78	59.53	67.41	65.02	42.49	70.80	61.58	66.55	65.02	43.22	69.94	61.20
h2o-danube2-1.8b-chat		66.11	62.72	41.34	71.67	60.29	67.18	63.03	39.4	69.36	60.41	67.72	64.39	40.55	69.07	61.16
	v1.1	64.20	62.40	40.92	68.49	58.87	66.86	61.67	40.98	67.05	60.29	67.11	64.29	41.52	67.34	60.92
	v1.2	65.13	62.72	40.01	70.23	59.34	67.58	63.03	41.16	69.94	61.07	66.72	65.02	40.79	67.91	60.65
Llama-2-7b-chat-hf		63.67	61.67	37.16	67.91	57.57	64.00	65.23	40.31	71.38	59.12	62.48	64.81	38.98	68.20	57.71
	v1.1	43.20	47.22	26.81	51.73	40.41	63.69	64.29	40.73	68.20	58.76	63.16	64.50	38.61	70.52	58.10
	v1.2	63.09	62.09	37.89	69.65	57.53	66.48	63.87	41.52	67.63	60.51	64.11	63.97	40.19	69.36	58.89
Meta-Llama-3-8B-Instruct		66.79	65.34	42.31	66.79	61.17	67.69	67.53	42.25	69.36	61.96	66.41	64.71	40.73	69.07	60.47
	v1.1	58.44	55.39	35.10	65.02	53.04	67.06	66.49	40.37	67.63	60.94	65.72	67.22	67.91	40.25	60.22
	v1.2	67.65	65.54	42.43	69.65	61.72	67.34	67.43	44.55	67.63	62.18	66.44	64.81	42.79	67.91	60.91
falcon-7b-instruct		66.25	64.39	40.01	69.36	60.18	68.37	65.75	41.10	67.34	61.76	66.81	64.60	39.70	68.20	60.41
	v1.1	65.51	63.35	39.76	69.07	59.53	64.53	62.40	40.55	63.87	58.76	62.74	63.35	40.13	63.00	57.68
	v1.2	65.93	62.40	40.79	67.91	59.83	68.23	62.82	40.79	64.73	61.10	67.74	64.29	39.46	66.18	60.77
Mistral-7B-Instruct-v0.2		69.46	69.28	43.46	67.91	63.04	69.06	65.86	44.18	69.65	63.08	66.18	65.23	41.52	70.53	60.65
	v1.1	68.58	65.23	41.64	73.12	62.22	69.62	68.06	45.21	71.38	63.94	68.86	66.70	43.46	70.80	62.88
	v1.2	69.23	65.96	42.79	68.20	62.73	70.60	67.74	45.09	73.41	64.55	68.34	67.43	44.67	73.69	63.09
vicuna-7b-v1.5		67.07	66.70	42.07	70.23	61.47	68.00	68.16	41.94	68.49	62.11	66.46	66.91	41.94	70.80	61.14
	v1.1	07.83	07.64	03.75	05.78	06.78	66.86	65.86	41.88	69.94	61.18	66.60	69.10	41.10	69.94	61.28
	v1.2	66.79	65.23	41.16	69.36	60.87	67.32	66.91	42.49	68.78	61.68	67.86	69.42	40.85	71.67	62.09

Table 11: F1 scores obtained by models when evaluated against our disambiguation benchmark for open-source LLMs in DG_σ setting (*All-MiniLM-L6-v2* as sentence embedder) with different prompts (*v1.1* and *v1.2*). For having a visual comparison, the first line of results of each model are the ones obtained by prompt version *v1*. Source: (Martelli et al., 2024)

Model		zero-shot					one-shot					few-shot				
		NOUN	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**	**NOUN**	**ADJ**	**VERB**	**ADV**	**ALL**
TinyLlama-1.1B-Chat-v0.1		50.00	47.91	49.32	60.00	49.80	49.62	52.08	50.00	50.00	50.10	49.81	52.77	50.00	40.00	50.00
	v1.1	00.18	00.00	00.33	00.00	00.20	50.00	49.30	49.66	60.00	50.10	49.81	52.77	50.00	40.00	50.00
	v1.2	06.81	06.25	06.71	03.33	06.60	50.00	53.47	50.00	40.00	50.20	49.81	52.77	50.00	50.00	50.00
	v1.3	50.56	47.22	49.66	63.33	50.20	49.81	52.77	50.00	40.00	50.00	50.00	52.77	49.32	36.66	49.80
Phi-3-mini-128k-instruct		58.33	56.94	53.35	63.33	56.80	53.21	52.08	53.35	40.00	52.70	55.30	52.70	53.35	40.00	53.90
	v1.1	03.40	02.08	03.35	00.00	03.10	57.76	55.55	54.69	46.66	56.20	61.36	52.77	54.02	50.00	57.60
	v1.2	15.90	22.22	09.06	10.00	14.60	57.76	51.38	56.04	43.33	55.90	55.87	52.77	52.34	43.33	54.00
	v1.3	47.53	36.80	39.59	30.00	43.10	61.55	56.25	55.70	46.66	58.60	70.64	67.36	64.09	53.33	67.70
stablelm-2-1_6b-chat		51.89	47.91	54.02	66.66	52.40	50.94	52.08	50.33	40.00	50.60	47.91	47.91	51.67	50.00	49.10
	v1.1	00.18	00.00	00.00	00.00	00.01	40.90	35.41	40.26	50.00	40.20	49.05	46.52	49.66	63.33	49.30
	v1.2	53.97	56.94	57.71	63.33	55.80	51.70	48.61	49.66	60.00	50.90	49.24	51.38	51.00	53.33	50.20
	v1.3	53.97	57.63	54.36	40.00	54.20	48.10	50.00	54.36	60.00	50.60	55.30	57.63	47.65	53.33	53.30
h2o-danube2-1.8b-chat		53.21	52.77	50.33	36.66	51.80	54.54	54.16	55.03	36.66	54.10	50.56	54.16	52.01	40.00	51.20
	v1.1	38.06	43.75	40.93	36.66	39.70	57.19	52.77	53.69	36.66	54.90	49.43	50.69	54.69	40.00	50.90
	v1.2	49.81	52.08	50.00	40.00	49.90	50.18	52.77	50.33	40.00	50.30	51.13	53.47	50.33	40.00	50.90
	v1.3	56.43	51.38	54.36	66.66	55.40	53.40	49.30	50.33	60.00	52.10	62.31	61.80	59.06	50.00	60.90
Llama-2-7b-chat-hf		21.78	23.61	20.13	23.33	21.60	56.06	52.08	53.02	43.33	54.20	50.94	52.08	48.99	56.66	50.70
	v1.1	04.16	04.86	05.03	03.33	04.50	47.53	52.08	46.97	53.33	48.20	49.81	54.16	52.68	36.66	50.90
	v1.2	43.75	43.75	39.26	20.00	41.70	49.62	54.16	54.36	50.00	51.70	48.67	50.69	52.68	63.33	50.60
	v1.3	39.77	44.44	40.93	33.33	40.60	59.84	49.30	51.00	53.33	55.50	63.06	61.11	53.35	50.00	59.50
Meta-Llama-3-8B-Instruct		49.43	52.08	52.68	60.00	51.10	48.67	43.75	48.32	56.66	48.10	50.18	47.22	49.64	60.00	49.90
	v1.1	13.44	09.02	13.42	16.66	12.90	50.37	48.61	49.32	66.66	50.30	50.18	47.22	50.00	60.00	50.00
	v1.2	00.18	00.69	01.00	00.00	00.50	47.91	47.91	48.99	53.33	48.40	50.94	48.61	50.00	56.66	50.50
	v1.3	50.18	47.22	50.00	60.00	50.00	43.93	45.83	47.98	63.33	46.00	60.41	57.63	55.70	36.66	57.90
falcon-7b-instruct		21.04	15.97	21.14	33.33	20.90	52.84	50.00	50.33	63.33	52.00	51.89	49.30	52.68	63.33	52.10
	v1.1	49.62	52.08	49.66	40.00	49.70	50.18	50.00	48.65	53.33	49.80	51.70	47.91	51.34	56.66	51.20
	v1.2	49.81	52.08	50.00	40.00	49.90	50.75	48.61	47.98	63.33	50.00	48.29	53.47	51.34	56.66	50.20
	v1.3	52.46	57.63	43.95	46.66	50.50	51.89	62.50	51.34	70.00	53.80	54.54	61.80	52.01	60.00	55.00
Mistral-7B-Instruct-v0.2		61.93	60.41	59.73	60.00	61.00	63.63	66.60	62.08	50.00	63.20	51.32	50.00	49.32	53.33	50.60
	v1.1	49.43	47.91	48.99	30.00	48.50	67.61	62.50	66.10	53.33	66.00	50.37	50.00	48.99	56.66	50.10
	v1.2	58.33	52.77	56.37	63.33	57.10	68.18	68.75	69.12	63.33	68.40	51.89	47.91	50.67	50.00	50.90
	v1.3	70.26	66.66	66.10	36.66	67.50	67.23	68.75	67.78	50.00	67.10	67.61	72.22	69.46	73.33	69.00
vicuna-7b-v1.5		49.81	52.77	50.00	40.00	50.00	51.32	47.91	51.00	56.66	50.90	50.18	47.22	50.00	60.00	50.00
	v1.1	00.00	00.00	00.00	00.00	00.00	50.18	47.22	50.00	60.00	50.00	50.18	47.22	50.00	60.00	50.00
	v1.2	49.81	52.77	50.00	40.00	50.00	50.75	47.91	51.00	60.00	50.70	50.18	47.22	50.00	60.00	50.00
	v1.3	53.03	50.00	56.04	60.00	53.70	50.18	47.22	49.32	60.00	49.80	50.18	47.22	50.33	60.00	50.10

Table 12: F1 scores obtained by models when evaluated against our disambiguation benchmark for open-source LLMs in the WiC$_\sigma$ setting with different prompts (*v1.1*, *v1.2* and *v1.3*). For having a visual comparison, the first line of results of each model are the ones obtained by prompt version *v1*. Source: (Martelli et al., 2024)