Word Sense Linking: Disambiguating Outside the Sandbox

Andrei Stefan Bejgu[1,2]**, Edoardo Barba**[1]**, Luigi Procopio**[3]
Alberte Fernández-Castro[4] **and Roberto Navigli**[1]
[1]Sapienza NLP Group, Sapienza University of Rome
[2]Babelscape, Italy   [3]Litus AI
[4]Roma Tre University
{lastname}@diag.uniroma1.it, bejgu@babelscape.com

Abstract

Word Sense Disambiguation (WSD) is the task of associating a word in a given context with its most suitable meaning among a set of possible candidates. While the task has recently witnessed renewed interest, with systems achieving performances above the estimated inter-annotator agreement, at the time of writing it still struggles to find downstream applications. We argue that one of the reasons behind this is the difficulty of applying WSD to plain text. Indeed, in the standard formulation, models work under the assumptions that a) all the spans to disambiguate have already been identified, and b) all the possible candidate senses of each span are provided, both of which are requirements that are far from trivial. In this work, we present a new task called Word Sense Linking (WSL) where, given an input text and a reference sense inventory, systems have to both identify which spans to disambiguate and then link them to their most suitable meaning. We put forward a transformer-based architecture for the task and thoroughly evaluate both its performance and those of state-of-the-art WSD systems scaled to WSL, iteratively relaxing the assumptions of WSD. We hope that our work will foster easier integration of lexical semantics into downstream applications.

1 Introduction

Leveraging the advances in pretrained transformer architectures (Devlin et al., 2019), Word Sense Disambiguation (WSD) systems have nowadays reached performances on several evaluation benchmarks that are on par with their estimated inter-annotator agreement (Bevilacqua and Navigli, 2020; Barba et al., 2021b). However, despite these advances, the task is still well known for struggling to find downstream applications. We argue that one of the possible causes is the difficulty of applying WSD in unconstrained settings due to its heavy assumptions. Indeed, three requirements need to be met for a generic state-of-the-art system to perform WSD on some input text: R1) a sense inventory, that is, a semantic resource providing a comprehensive list of all the senses of interest, must be provided, R2) the list of spans to disambiguate in the input text must have already been identified, and R3) an oracle that pairs each span to its set of possible senses, realized through a manually curated word-to-sense mapping, must be available.

While the first condition is intrinsic to the disambiguation objective and, thus, unavoidable, we argue that the second two (i.e. R2 and R3) are system-specific assumptions that can be relaxed. We formulate this idea into a new task called Word Sense Linking (WSL), which more accurately reflects the conditions of downstream applications such as Neural Machine Translation (Liu et al., 2018; Iyer et al., 2023), Information Extraction (Moro and Navigli, 2013; Delli Bovi et al., 2015) and the enrichment of contextual representations in neural models (Peters et al., 2019). In WSL, given an input text and a reference sense inventory, systems have to identify which spans to disambiguate and link them to their most suitable meaning in the sense inventory.

Similarly to Entity Linking (Broscheit, 2019), WSL can be split into three simpler subtasks, with traditional WSD taking place after two initial stages of Concept Detection (CD), that is, the identification of the spans to be disambiguated in the input text (R2), and Candidate Generation (CG), the generation of a list of sense candidates for each target span (R3). For example, given some reference sense inventory and the sentence *"Bus drivers drive buses for a living."*, Concept Detection might identify `[bus drivers, drive, buses, living]` as the spans to disambiguate, while Candidate Generation might provide `[vehicle, electrical conductor]` as the sense candidates for *buses*.

In this work, we first formally introduce the task of Word Sense Linking, along with its three com-

14332

ponents, and, then, put forward a novel architecture for the task, based on the retriever-reader paradigm (Karpukhin et al., 2020). We thoroughly study the behavior of our architecture and that of state-of-the-art WSD systems, once they have been scaled to WSL, as we iteratively relax the sandboxed assumptions of WSD, starting with R2 and continuing with R3. Our analysis highlights a number of important – yet neglected – challenges when it comes to performing disambiguation in unconstrained settings. In particular, straightforward and natural extensions of WSD systems to WSL result in large performance drops. In contrast, our model demonstrates considerably more robustness and consistently outperforms such extensions of WSD systems by a large margin. The contributions of this work are therefore as follows:

- We introduce the task of Word Sense Linking, which we believe to better represent the settings of downstream applications for WSD.

- We introduce for the first time a Word Sense Linking evaluation dataset, enriching the *de-facto* standard WSD benchmark (Raganato et al., 2017).

- We put forward a novel flexible architecture for this task and evaluate, in multiple settings, both its behavior and that of state-of-the-art WSD systems scaled to WSL.

- Overall, our findings underline several crucial yet neglected challenges when scaling WSD systems to a real-world scenario.

We release code, data, and model weights at https://github.com/Babelscape/WSL.

2 Word Sense Linking

Word Sense Linking is the task of identifying and disambiguating all the spans of an input text with their most suitable senses chosen from a reference inventory. Formally, let t be the input text, with $t_1, \ldots, t_{|t|}$ being its words, and I the reference inventory, containing a set of senses. Then, a WSL system can be represented as a function f that takes as input the tuple (t, I) and outputs a list of triples $[(s_1, e_1, g_1), \ldots, (s_n, e_n, g_n)]$ where each triple (s_i, e_i, g_i), $i \in [1, n]$, represents a disambiguated span, with s_i and e_i being the start and end token index of the span, and $g_i \in I$ representing the corresponding sense chosen from the

inventory. Conceptually, WSL can be divided into three subtasks, namely, Concept Detection, Candidate Generation, and Word Sense Disambiguation.

Concept Detection (CD) is the task of identifying the spans to disambiguate in an input text t given a reference inventory I. A Concept Detection system can be represented as a function that takes as input the tuple (t, I) and outputs a list of pairs $[(s_1, e_1), \ldots, (s_n, e_n)]$, each marking the boundaries of a span to disambiguate. Specifically, $\forall i \in [1, n]$, s_i and e_i are the start and end token index of the i-th span to disambiguate.

Candidate Generation (CG) is the task of generating a set of possible candidate senses that an input span occurring in some text t can assume given the reference inventory I. A Candidate Generation system can be represented as a function that takes as input t, I and the start and end token indices (s, e) of the span, and outputs a set of sense candidates $PC \subseteq I$, that the span can assume.

Word Sense Disambiguation (WSD) aims at identifying, for each target span occurring in an input text t, the most suitable sense among a set of possible candidates. A WSD system can be represented as a function that takes as input t, I, a list of span indices $[(s_1, e_1), \ldots, (s_n, e_n)]$ and corresponding sets of possible candidates $[PC_1, \ldots, PC_n]$. It then outputs a list of triples $[(s_1, e_1, g_1), \ldots, (s_n, e_n, g_n)]$ where, $\forall i \in [1, n]$, each span (s_i, e_i) is paired with the sense g_i chosen by the system among the candidates provided in PC_i.

Finally it is easy to see that a WSL system can be built by concatenating, in cascade, a Concept Detection, a Candidate Generation, and a Word Sense Disambiguation module. This structure not only facilitates the extension of WSD systems to WSL but also serves as a flexible framework rather than a strict recipe. In fact, as we will show in the next section, the architecture we put forward does not follow this flow and inverts the CD and CG steps.

3 Model

The flow *CD-then-CG* presents an intrinsic limitation: each span identified by Concept Detection ought to correspond to the occurrence of a specific sense candidate among those produced by Candidate Generation and, yet, CG occurs after CD. That
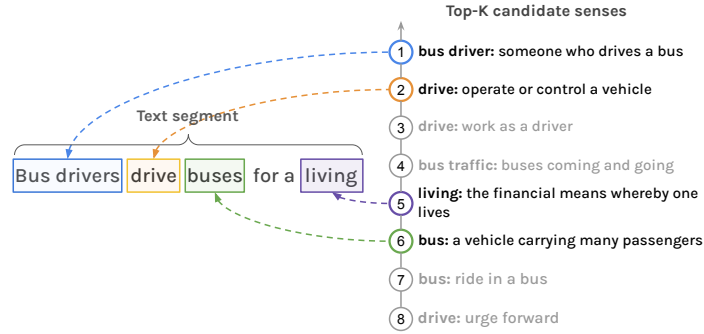
Figure 1: Our WSL process. First, the retriever identifies the top-k candidate senses (Candidate Generation). Then, the reader identifies the spans to be disambiguated (Concept Detection) and pairs each of these with their most suitable sense (Word Sense Disambiguation).

is, we need to identify the span we will, later on, link to a specific sense without actually knowing this sense. To overcome this limitation, inspired by the work presented by Zhang et al. (2022b) for Entity Linking, we invert the CD and CG steps, and propose a novel flexible architecture for WSL based on the retriever-reader paradigm. In this section, we first outline its formulation (Section 3.1) and, then, discuss its two components, namely the retriever (Section 3.2) and the reader (Section 3.3).

3.1 Formulation

Starting from the input text t and the reference inventory I, our formulation is as follows. First, we perform CG on the entire input text, producing an ordered list of unique candidates $PC(t) \subseteq I$, each coming from I and likely to represent the meaning of some arbitrary span in t. Then, moving to CD and using vector representations contextualized on both t and $PC(t)$, we let a classifier identify the start and end token indices of every span in t; this operation results in a list of n tuples $[(s_1, e_1), \ldots, (s_n, e_n)]$. Finally, we perform WSD on the identified spans, pairing each (s, e) with its most suitable sense $g^* \in PC(t)$.[1] Figure 1 reports a visual outline of this process.

We implement our formulation with transformer-based architectures that operate on textual inputs. However, this makes it necessary for our inputs to have such representations and, while t inherently satisfies this requirement, the same does not hold for the senses in I. To overcome this issue, for each sense $g \in I$, we define a textual representation, which we build by concatenating its lemmas[2], provided through the mapping from word to possi-

ble senses, and its textual definition in I.[3]

Finally, we note that t can be arbitrarily long. Consequently, providing the entire t as input to our formulation can be challenging, both computationally and from a modeling perspective. To overcome this, we adopt a sliding window approach, with window size w and stride τ, and actually feed, to the previous steps, single chunks of t, rather than the entire sequence. Once all chunks have been processed for t, we retain only *conflict-free predictions*, i.e., every triple (s, e, g^*) such that the processing of every chunk that includes (s, e) in its window always results in assigning the same g^* to (s, e). Henceforth, with no loss of generality, we replace our input, and the meaning of notation t, with a generic chunk, rather than the entire sequence.

3.2 Retriever

We implement the initial Candidate Generation step via dense passage retrieval (Karpukhin et al., 2020), using a transformer-based retriever consisting of an encoder E to produce a dense representation of text passages and senses. Starting from the input text t and the inventory I, we use E to compute a vector representation v_t for t, and v_g for every sense $g \in I$. Then, we use the dot product $v_t \cdot v_g$ to rank all the senses in I and, finally, extract the top k among these. The resulting $g^1, \ldots, g^k$ senses constitute our sense candidate set $PC(t)$ for t.

To train our model, we use a multi-label variant of noise contrastive estimation (Zhang et al., 2022b, NCE), maximizing the following objective:

$$\sum_{g \in \Gamma(t)} \log \frac{\exp(v_t \cdot v_g)}{\exp(v_t \cdot v_g) + \sum_{g' \in \mathrm{N(t)}} \exp(v_t \cdot v_{g'})}$$

---

[1] We extend the $\in$ operator to work on lists as well.

[2] If a mapping between senses and words is available.

[3] Although we did not explicitly mention it before, it is customary for sense inventories to provide, for each of their senses, a textual definition (*gloss*) that defines its meaning.

where $\Gamma(t) \subset I$ is the set of gold senses appearing in t and $N(t) \subset I \setminus \Gamma(t)$ represents a selection of negative examples. Considering the crucial role of $N(t)$ in retriever-based architectures (Karpukhin et al., 2020), we adopt a strategy aimed at selecting adequately informative negative examples, while, at the same time, accounting for the presence of unannotated tokens. Starting from $I \setminus \Gamma(t)$, we first discard the senses that never appear in the training set or whose part of speech does not match any sense-annotated span in t. Then, denoting by $I^R(t)$ the set of resulting senses in I for t, we define $N(t)$ as the union of the two sets $N_1(t)$, a collection of hard negatives (Gillick et al., 2019) in $I^R(t)$, and $N_2(t)$, a subsample of $I^R(t)$ that aims at counterbalancing the bias towards the most frequent senses in the training corpus. Specifically, we build $N_1(t)$ by selecting the ν_1 senses in $I^R(t)$ to which the retriever assigns the highest score, and $N_2(t)$ constructed using ν_2 gold senses from other samples in the same mini-batch.

3.3 Reader

Having identified the sense candidates $PC(t) = g^1, \ldots, g^k$, we now describe the remaining Concept Detection and WSD steps, which we formulate as a multi-task multi-label classification problem. To this end, we concatenate t and $g^1, \ldots, g^k$ into a single sequence m, prepending a special symbol $[S_i]$ for each sense candidate g^i:

$$m = t \, [S_1] \, g^1 \, \ldots \, [S_k] \, g^k \qquad (1)$$

We feed m to a transformer encoder, producing a series of vector representations for the tokens in m; let $h_1, \ldots, h_{|t|}$ be the representations corresponding to the tokens in t, and h_g the one associated with the special symbol of a generic sense candidate $g \in PC(t)$. With these contextualized representations, we realize CD and WSD as follows.

We begin with CD, identifying the spans to disambiguate. Specifically, we apply two classification heads, namely H_{start} and H_{end}, on each representation to determine whether the corresponding token is a start or end of some span in t; both heads consist of two linear transformations with a ReLU activation in between. However, their behavior – and input – differ: while H_{start} receives single token representations, H_{end} operates at span level. That is, once H_{start} has identified some index s as a span start, to determine whether some other index $e \geq s$ is its end, H_{end} takes as input $[h_s; h_e]$,

that is, the concatenation of h_s and h_e. Once completed, this step results in a list of span indices $[(s_1, e_1), \ldots, (s_n, e_n)]$.

Then, moving to WSD, we let each extracted span (s, e) identify the sense $g^* \in PC(t)$ it refers to. We start by computing the following vectors:

$$h'_{se} = [f_1(h_s); f_2(h_e)]$$
$$h'_g = [f_1(h_g); f_2(h_g)] \quad \forall g \in PC(t)$$

where f_1 and f_2 are both functions consisting of two identical but independent linear transformations, interleaved by a ReLU activation. Then, we pair the span (s, e) with the sense $g^* = \operatorname{argmax}_{g \in PC(t)} h'_{se} \cdot h'_g$. We replicate this strategy for all the extracted spans, hence eventually producing as output $[(s_1, e_1, g_1^*), \ldots, (s_n, e_n, g_n^*)]$. Our reader is trained by jointly maximizing three cross-entropy objectives, respectively over i) the gold start indices in t, ii) the gold end indices in t and iii) the gold sense for every span in t.

We note that our architecture presents a number of interesting properties. First, since $h_1, \ldots, h_{|t|}$ are vectors contextualized over the entire input sequence m (Equation 1), which includes $g^1, \ldots, g^k$, the choice of the extracted spans does indeed take into account the sense candidates available. Second, this contextualization is not limited to $h_1, \ldots, h_{|t|}$ but, in fact, applies to $h_{g^1}, \ldots, h_{g^k}$ as well. This means that the sense candidates are also contextualized on each other, allowing for better representations, as in Barba et al. (2021a). Third, computationally speaking, the only demanding operation corresponds to the encoding of $h_1, \ldots, h_{|t|}$. However, since this operation depends only on t and $PC(t)$, we can disambiguate all the spans in t in a single forward pass, which makes it very fast and hence suited for downstream applications.

4 Evaluation

We now investigate the effectiveness of our model and how it relates to the R2 and R3 assumptions of WSD (Section 1). To this end, we first outline its implementation details (Section 4.1) and the novel dataset introduced for the WSD and WSL evaluation (Section 4.2), which will remain unchanged throughout our experiments. Then, we evaluate it on plain English WSD (Section 4.3), so as to set an initial reference for what regards its disambiguation capabilities. Finally, we move to WSL and gradually relax these sandboxed assumptions,

dropping R2 (Section 4.4) and examining different relaxations on R3 (Section 4.5).

4.1 Model Details

Formulation Hyperparameters We use $w = 16$ and $\tau = 8$ for our sliding window approach. This ensures that each window captures sufficient context for an effective disambiguation while maintaining a manageable overlap between consecutive windows.

Retriever We employ mean pooling over $E5_{base}$ (Wang et al., 2022), a pretrained transformer-based architecture, for our retriever encoder E. The entire system is trained with a batch size of 16 input texts for 150,000 steps using AdamW (Loshchilov and Hutter, 2019), with 20% warm-up and $2 \cdot 10^{-5}$ learning rate, and setting $k = 100$. At training time, we construct the sense candidate set by including the gold senses, $\nu_1 = 5\%$ of negatives from $N_1(t)$ and the remaining ν_2 from $N_2(t)$.

Reader We use the base version of DeBERTa-v3 (He et al., 2021a) as the transformer-based encoder in our reader. The four linear transformations, namely two for span detection and two for Word Sense Disambiguation, present the same intermediate dimensionality, that is, 768; instead, the final dimension is 1 for the span detection mappings, as they are classification heads, but 768 for the WSD mappings. The overall system is trained with a batch size of 4096 tokens for 100,000 steps using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4} and a layerwise decay rate of 0.8.

4.2 WSL Benchmark

The framework presented by Raganato et al. (2017) represents the de facto standard benchmark for WSD and is based on the WordNet sense inventory (Miller et al., 1990). Specifically, it is composed of Senseval-2 (Edmonds and Cotton, 2001, **SE02**), Senseval-3 (Snyder and Palmer, 2004, **SE03**), SemEval-2007 (Pradhan et al., 2007, **SE07**), SemEval-2013 (Navigli et al., 2013, **SE13**) and SemEval-2015 (Moro and Navigli, 2015, **SE15**). The original task datasets might include spans of text that contain content words but were not assigned any specific meaning by annotators from among those contained in the reference sense inventory. This could have occurred for a variety of reasons, such as in the case of SE13, where only nouns were chosen to be annotated. Whereas for

the WSD setting the absence of this information does not pose a problem for evaluation, for WSL it renders evaluation impossible. Specifically, we cannot in such case measure the precision of a WSL system as we cannot discern between wrongly predicted spans and annotation omissions. To address this issue and overcome the lack of a WSL-specific dataset, we introduce a dedicated evaluation resource aimed at bridging the annotation gap, not just in terms of precision, but also in terms of recall. We comprehensively annotated the standard WSD evaluation datasets, increasing the annotation count from 7253 to 11623 annotations, and resulting in the complete coverage of all the content words.

This substantial increase in annotations allows for a comprehensive benchmark, facilitating future research in the field by providing a more robust framework for evaluating the precision, recall, and F1 of WSL systems.

Annotation process We have annotated the standard WSD evaluation dataset (i.e., ALL, the one utilized in the previously presented WSD evaluation framework (Raganato et al., 2017)). This process aimed to ensure a comprehensive and accurate representation of terms and their meanings, using WordNet as the sense inventory. We selected these guidelines for the annotators:

1. **Multiwords:** Annotators marked terms such as "lung cancer" as single entities, focusing on those recognized in WordNet with contextually coherent meanings.

2. **Sub-words:** Annotators also marked the individual components of multiword expressions, but only when the sub-words had coherent meanings within the sentence context. This dual-level annotation strategy captured both the general and specific meanings of terms.

3. **Non-content Words**: Annotators excluded non-content words, such as auxiliary verbs, from annotation. These words are essential for grammatical structure but do not carry any semantic weight.

Employing WordNet as the sense inventory facilitated a uniform and precise approach to annotation across the entire dataset and maintained consistency with WSD tasks, ensuring alignment with established standards and methodologies in the field.

Due to the complexity of the task, a single expert linguist, who is also an author of this paper

and has a robust background in the annotation of lexical-semantic tasks, conducted the majority of the annotation work. Nonetheless, to validate the reliability and consistency of the annotation process, we conducted an inter-annotator agreement evaluation. This involved a subset of the dataset, constituted by 20 sentences independently annotated by three different expert linguists,[4] resulting in 222 distinct annotations. The agreement among annotators was measured using Cohen's kappa statistic, which yielded a score of 0.83, interpreted as an almost perfect level of agreement, showing a high degree of annotation consistency. These results underscore the robustness and reliability of our annotation methodology, despite the inherent challenges of subjective interpretation and the detailed demands of manual annotation. Further statistics about the newly introduced dataset can be found in Appendix B.

4.3 Word Sense Disambiguation

Setting We evaluate our model on all-word English WSD using WordNet as our sense inventory. WordNet provides a comprehensive and structured database of English word senses, making it a widely accepted benchmark for WSD. By utilizing WordNet, we ensure that our evaluation aligns with established standards in the field, facilitating comparison with previous work and other state-of-the-art models.

Comparison Systems We compare our model with recent state-of-the-art systems for WSD, which we divide into two different categories. On the one hand, we consider systems that frame WSD as a sequence-level classification problem, that is, systems that disambiguate a single span at a time. We include in our evaluation: Barba et al. (2021a, **ESCHER**), the first approach reformulating WSD as a text extraction problem; Zhang et al. (2022a, **KELESC**), a knowledge-enhanced version of ESCHER, incorporating additional information coming from WordNet; Song et al. (2021, **ESR**), an architecture framing WSD as a binary classification problem; and Barba et al. (2021b, **ConSeC**), the system that, thanks to iterative disambiguation, holds the state of the art on the reference benchmark at the time of writing.

On the other hand, we report models that frame WSD as a token-level classification problem, thus

---

	Models	Params	SE07	ALL	ALL$_{FULL}$
Sequence	ESCHER	400M	76.3	80.7	81.2
	KELESC	400M	76.7	81.2	81.4
	ESR	350M	77.0	81.1	81.3
	ConSeC	400M	**77.4**	**82.0**	**82.5**
Token	WMLC	340M	72.2	77.6	78.1
	EWISER	340M	71.0	78.3	78.9
	BEM	220M	74.5	79.0	79.7
	Our Model	295M	**75.2**	**80.2**	**80.8**

Table 1: WSD results for sequence-level and token-level classifiers.

disambiguating all the spans in a sentence together. We include: Blevins and Zettlemoyer (2020, **BEM**), a bi-encoder system incorporating gloss knowledge; Bevilacqua and Navigli (2020, **EWISER**), a classifier modeling the relational knowledge in WordNet using a Personalized PageRank approach; and Conia and Navigli (2021, **WMLC**), a classification formulation for WSD that leverages the relational knowledge in WordNet at training time. We use this division to highlight not only the different designs of our comparison systems but, also, their corresponding trade-offs. Indeed, while sequence-level classifiers typically achieve higher performances, token-level classifiers emphasize speed and are considerably more usable in downstream applications.

Data We evaluate our model using the framework presented by Raganato et al. (2017). Specifically, we train our system on SemCor (Miller et al., 1993), perform model selection on SE07, and test on the other available datasets. We measure performances in terms of F1 score and, as in previous works, report this score on the concatenation of each evaluation dataset (identified as **ALL**). Finally, to set a reference for the WSL setting, we evaluate our model and the comparison systems on the novel benchmark presented in Section 4.2 under the WSD assumptions (identified as **ALL**$_{FULL}$).

Our Model Behavior Comparison systems operate in this setting assuming the availability of an oracle for both CD and CG. To enable a fair comparison, we i) limit the candidates retrieved from the retriever module only to those that the spans to disambiguate can assume in WordNet and ii) force the reader to select for each span only the senses that it can assume in WordNet.

Results Table 1 shows how all the systems under comparison fare on the standard framework.

As a first result, we note that our model, a token-level classifier as all spans are disambiguated together, outperforms all the other token-level systems, almost reaching sequence-level ESCHER's performance. Furthermore, while the difference in comparison to ConSeC, i.e., the best-performing model, is significant in terms of F1 score (1.8 points), the speed comparison reveals a compelling advantage: using ALL as a reference, our model can process the 7253 instances in less than 17 seconds, while ConSeC requires a total of 73 seconds.[5] These results suggest that our model provides a solid foundation for our setting, showing not only strong performance on WSD, but also that it is flexible enough to support CD and CG. Finally, and most importantly, we can see that the performances obtained by the comparison systems on the ALL_{FULL} dataset align with those on the standard benchmark, allowing us to study how they change when relaxing the WSD constraints .

4.4 Word Sense Linking: Dropping the Concept Detection Oracle

Setting Here, we drop the R2 condition. In the context of WSL, this means that the system is required to identify spans of text that need to be disambiguated. By removing this oracle, we simulate a more challenging and practical setting where the system has to detect the spans to disambiguate before proceeding to candidate generation and disambiguation.

Comparison Systems Given the task's novelty, there are no direct comparison systems. We take advantage of this opportunity to assess the complexity of Concept Detection and its impact on ConSeC and BEM, that is, the best sequence-level and token-level classifiers described in the previous section. Specifically, to establish baselines, we implement two natural straightforward solutions for CD and pipeline each of these with our reference WSD systems. Our first solution consists of a simple heuristic approach: given an input text, we find all the longest spans such that the corresponding lemma is linked, through the mapping from word to possible senses, to any sense in WordNet. As an alternative to this strategy, we consider a supervised implementation to determine whether each token represents the start, inside, or outside of a span to

| | **SE07** | | | ALL_{FULL} | | |
Models	P	R	F1	P	R	F1
BEM_{SUP}	67.6	40.9	51.0	74.8	50.7	60.4
BEM_{HEU}	70.8	51.2	59.4	76.6	61.2	68.0
ConSeC_{SUP}	76.4	46.5	57.8	78.9	53.1	63.5
ConSeC_{HEU}	**76.7**	55.4	64.3	**80.4**	64.3	71.5
Our Model	73.8	**74.9**	**74.4**	75.2	**76.7**	**75.9**

Table 2: WSL results with no CD oracle.

Models	Lemmas	P	R	F1	Δ F1
ConSeC_{HEU}	all	80.4	64.3	71.5	–
ConSeC_{HEU}	one	71.6	56.4	63.1	-8.4
ConSeC_{HEU}	no	0.0	0.0	0.0	-71.5
Our Model	all	75.2	76.7	75.9	–
Our Model	one	70.4	73.1	71.7	-4.2
Our Model	no	68.5	62.5	65.4	-10.5

Table 3: WSL analysis on CG oracle.

disambiguate. We train[6] a variant of the architecture of Mueller et al. (2020) widely used in Named Entity Recognition tasks using BERT-base as an encoder. We denote the four variants by BEM_{SUP}, BEM_{HEU}, ConSeC_{SUP} and ConSeC_{HEU}.

Data To address the issue of missing annotations in the training phase of our WSL system, specifically when using SemCor, which is known for its incomplete annotations, we propose a mitigation strategy. We use ConSeC_{HEU} to identify and annotate all the missing spans in SemCor, leading to the creation of SemCor_C. This procedure results in 133,727 new annotations, in addition to the original 226,036 already in SemCor. We note that the usage of these annotations proves to be crucial for our model but irrelevant, if not actually harmful in some cases, for both BEM and ConSeC. Further details are available in Appendix C. Therefore, in what follows, the results we report always refer to the usage of SemCor_C for our model and of solely SemCor for our comparison systems.

Our Model Behavior Differently from the previous section, here we run the model in a WSL setting. The retriever identifies the sense candidates and the reader their corresponding spans. However, as the other systems use a CG oracle, for a fair comparison, for a given identified span, we limit the reader to selecting only the senses it can assume.

---

[5]To perform the comparison, we use the code made available by the authors at `https://github.com/SapienzaNLP/consec` using an RTX 3090 for both experiments.

[6]We use the span information contained in SemCor for training and SemEval-2007 for model selection.

Results   Table 2 reports the behavior of our model and the comparison systems. We note that all systems experience, in this unconstrained setting, a large drop in performance, with scores diverging markedly from those reported for WSD in Table 1 on $\mathbf{ALL}_{FULL}$ (e.g., 82.5 vs 71.5 for ConSeC). All the same, our model behaves significantly better, attaining 75.9 (5 point drop) and surpassing the best alternatives, i.e., ConSeC_{HEU}, by almost 5 points. Interestingly, while our model does, in fact, lose to ConSeC_{HEU} in terms of precision (80.4 vs 75.2), it displays large improvements in recall (64.3 vs 76.7).

To better understand these results, we investigate how CD approaches fare in terms of span recall.[7] What we find is that, while our model achieves 92.1, the two baselines lag behind by a considerable margin, with HEU and SUP reaching 80.6 and 68.2, respectively. Besides showing the effectiveness of our model on CD, this has several implications. First, tackling CD in a supervised fashion, in the current regime of WSL data, without taking into account the reference inventory, is not sufficient for good performance. Second, while HEU is a better baseline, neglecting the semantics, i.e. choosing the longest span with at least one valid sense, is deleterious. Paired together, these findings suggest that hybrid schemes like ours, where both training examples and the inventory are jointly employed, represent a better alternative. Finally, while HEU achieves nearly acceptable performance in English, the same does not hold for other languages where word inflection is a more complex phenomenon, and the mapping from word to senses lacks coverage compared to English[8]: for instance, testing span recall in Italian, using XL-WSD (Pasini et al., 2021), results in 70.7, 10 points less than English.

4.5 Word Sense Linking: Relaxing the Candidate Generation Oracle

Setting and Comparison Systems   Here, we drop the R2 condition and examine different relaxations of R3. That is, compared to the setting of the previous section, we assume the mapping from word to possible senses, which we used up to this point as our Candidate Generation oracle, to be either incomplete or absent. Such scenarios are realistic as they mimic low-resource language settings in which dedicated WordNet and the related word-to-sense mappings are incomplete. We compare our model with ConSeC_{HEU}.

Data   To highlight the importance of having a complete mapping between senses and their lemmas, we analyze the performance of both systems in three different settings, that is, when, for each synset, we have at our disposal: i) all the lemmas with which it can be expressed, ii) only its most frequent lemma, or iii) no lemma at all.

In the first setting, where all possible lemmas are available, the system can utilize all the lexical variations for disambiguation. The second setting, which restricts the information to only the most frequent lemma, simulates a scenario with limited lexical resources, requiring the system to rely only on the most common representation of each sense. This may impact the system's ability to accurately disambiguate less frequent word forms. The third setting, where no lemma information is provided, forces the system to rely solely on the definition, and this is the most challenging scenario.

Our Model Behavior   Our model behavior is unchanged compared to the previous section. The only difference across the three settings regards the textual representation of each sense in I, which now consists of its definition postpended to i) all its lemmas, ii) only its most frequent lemma, or iii) no lemma at all.

Results   As evidenced by Table 3, both systems perform worse when leveraging an incomplete mapping. However, our model shows a more robust behavior in such a setting compared to ConSeC_{HEU}. Indeed, as it performs the CG step independently from the mapping (with the retriever module), and takes advantage of the lemmas only to enhance sense representations, it is less reliant on this mapping. In contrast, ConSeC_{HEU} requires the sense mapping in order to retrieve the candidates for the WSD step.

More in detail, when limiting to only the main lemma for each sense, ConSeC_{HEU} can only predict senses that are expressed by their primary lemma. As expected, this results in a decrease of 8.4 F1 points. Our model, instead, is able to predict all senses, with a lower impact of 4.2 F1 points. Moving to the scenario where no lemmas are available, our system is still able to recognize and disambiguate spans, but its performance drops significantly by 10.5 F1 points. Yet, compared

---

[7]This refers to the accuracy of systems in identifying spans in ALL, irrespective of their associated senses.

[8]This refers to the resources available to the research community.

with $ConSeC_{HEU}$, this is still a huge improvement: indeed, $ConSeC_{HEU}$ is completely unable to perform the task as it lacks candidates.

Our objective is to evaluate our models' effectiveness in scenarios with limited CG oracles, typical of most mid- and low-resource languages in WSD. The results we report not only underline the robustness of our model but also highlight the inadequacy of modern WSD systems.

5 Related Work

Since this work is the first to introduce Word Sense Linking, unsurprisingly, there is no previous literature that covers it. However, what is surprising is the total absence, to the best of our knowledge, of studies that examine how recent state-of-the-art WSD systems scale to real-world scenarios. This is especially puzzling when we consider the growing interest that WSD has been receiving. Indeed, thanks to the introduction of pretrained transformer-based language models (Devlin et al., 2019; Lewis et al., 2020; He et al., 2021b), this task has been witnessing renewed attention, with the research community focusing on challenging directions such as unseen prediction, cross-inventory generalization, and data efficiency, inter alia. Among the alternatives explored, the usage of definitions, which we also follow in this work, has proved to be particularly effective (Huang et al., 2019; Blevins and Zettlemoyer, 2020; Barba et al., 2021a), achieving unprecedented performances and allowing sense representations to be disentangled from their occurrences in the training corpus.

Yet, in spite of this trend and its promising results, to the best of our knowledge, no assessment of how these models might actually be applied in general real-world scenarios has been made, thereby overlooking the relevance of Concept Detection and Candidate Generation. We suspect that this oversight is due mainly to two reasons: i) the performances on WSD benchmarks being too scarce, at least until recently, for any downstream application, and ii) the research community assuming that adequate word sense mappings are always available for CG and that a simple heuristic approach could solve CD. In our CD approach, we chose the greedy strategy, a decision supported by Martínez-Rodríguez et al. (2020), who observed its widespread use in span identification within texts. However, our study clearly shows that is not a trivial task. Both heuristic and supervised techniques report definitely suboptimal behaviors.

Finally, to provide some background on Candidate Generation, the task has generally been approached by striving to enumerate all the possible senses a word can assume. However, this is a prohibitively challenging endeavor, especially when wishing to scale across languages. While the research community has put forward a number of studies and mitigating strategies (Taghizadeh and Faili, 2016; Al Tarouti and Kalita, 2016; Khodak et al., 2017; Neale, 2018), the resulting resources are still incomplete.

Arguably closest to our work, especially to our WSL model, is Zhang et al. (2022b), who address Entity Linking by initially generating candidates, followed by Mention Detection (akin to our Concept Detection) and Entity Disambiguation (similar to our Word Sense Disambiguation). However, their reader architecture differs significantly from ours: they link one candidate at a time, whereas our model can simultaneously process all spans in an input sequence. A comparison with our model can be found in Appendix D.

6 Conclusion

In this work, we challenge the assumptions behind Word Sense Disambiguation (WSD) and introduce a novel task called Word Sense Linking (WSL). WSL requires a system to identify and disambiguate all the spans in an input text using only the information contained in a reference inventory, offering a scenario that is more aligned with practical downstream applications than the conventional WSD approach. Along with the WSL formalization, we discuss a first comprehensive study in this direction, presenting a novel retrieved-reader architecture for the task, a complete and comprehensive benchmark for WSL systems, and an analysis of its performances and those of state-of-the-art WSD systems in multiple settings. Our findings highlight several important yet overlooked challenges that arise when scaling to unconstrained settings. In particular, natural expansions of WSD systems to WSL appear to be quite brittle, resulting in large performance drops. Conversely, our proposed architecture appears to be considerably more robust, achieving superior performances across all WSL settings. Looking ahead, we plan to investigate the expansion of WSL to a multilingual setting and analyze the usage of WSL systems in downstream applications.

7 Limitations

This work has two inherent limitations: first, due to space constraints, we have deferred the evaluation of the model in a multilingual setting to future work. Potential challenges include the necessity for an extension of the sense inventories and the availability of training resources, which are requirements that go beyond the scope of the current study. Second, the lack of WSL-specific annotated data meant that we had to rely on datasets designed for Word Sense Disambiguation for training our models. Although these datasets offered valuable insights and exhibited promising results on our WSL-specific benchmarks, the prevalence of annotation gaps could hinder the performance of WSL systems. The effort to develop such datasets would be extensive, mirroring the significant undertaking required for our WSL-specific evaluation benchmark.

References

Feras Al Tarouti and Jugal Kalita. 2016. Enhancing automatic Wordnet construction using word embeddings. In *Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP*, pages 30–34, San Diego, California. Association for Computational Linguistics.

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

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

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

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

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China. Association for Computational Linguistics.

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

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543.

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

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

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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

Vivek Iyer, Edoardo Barba, Alexandra Birch, Jeff Pan, and Roberto Navigli. 2023. Code-switching with word senses for pretraining in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12889–12901, Singapore. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Mikhail Khodak, Andrej Risteski, Christiane Fellbaum, and Sanjeev Arora. 2017. Automated WordNet construction using word embeddings. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 12–23, Valencia, Spain. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Frederick Liu, Han Lu, and Graham Neubig. 2018. Handling homographs in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1336–1345, New Orleans, Louisiana. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

José-Lázaro Martínez-Rodríguez, Aidan Hogan, and I. Lopez-Arevalo. 2020. Information extraction meets the semantic web: A survey. *Semantic Web*, 11:255–335.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. Introduction to WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics.

Andrea Moro and Roberto Navigli. 2013. Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

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

David Mueller, Nicholas Andrews, and Mark Dredze. 2020. Sources of transfer in multilingual named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8093–8104, Online. Association for Computational Linguistics.

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

Steven Neale. 2018. A survey on automatically-constructed WordNets and their evaluation: Lexical and word embedding-based approaches. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. XL-WSD: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proc. of AAAI*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word

representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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

Yang Song, Xin Cai Ong, Hwee Tou Ng, and Qian Lin. 2021. Improved word sense disambiguation with enhanced sense representations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4311–4320, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nasrin Taghizadeh and Hesham Faili. 2016. Automatic wordnet development for low-resource languages using cross-lingual wsd. *Journal of Artificial Intelligence Research*, 56:61–87.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training.

Guobiao Zhang, Wenpeng Lu, Xueping Peng, Shoujin Wang, Baoshuo Kan, and Rui Yu. 2022a. Word sense disambiguation with knowledge-enhanced and local self-attention-based extractive sense comprehension. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4061–4070, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022b. EntQA: Entity linking as question answering. In *International Conference on Learning Representations*.

A Retriever Performances

In this section, we show the results of the Retriever module using the same hyperparameters for training as those described in Section 4.1 and making just one change to the configuration showing the impact on the performances in terms of recall top-100 (R@100). We report the results of our experiments in Table 4. The baseline retriever achieves a 96.5 R@100, confirming our thesis that performing the Candidate Generation step is possible without knowing the spans a priori. Moreover, these results set a remarkable upper bound for the Reader module performances. The architecture of our baseline model is based on $bert - base - uncased$ (Devlin et al., 2019) initialized with the weights from Sentence-Transformers (Reimers and Gurevych, 2019) and in particular using the weights of $E5_{base}$ (Wang et al., 2022). In this baseline setting, the textual representation of the senses of I is the standard one, namely, that composed by the concatenation of all its available lemmas and its textual definition. We can see that initializing our weights from the generic $bert - base - uncased$ yield to 7.8 points performance loss in recall shows that sentence-embedding pretraining is useful. Furthermore, using a more parameter-efficient architecture (33M, MiniLM-L6) compared to our reference one (109M, bert-base-uncased) still leads to competitive results (92.5). Finally, we can see that the standard setting where the senses of I are represented with all the lemmas yields the best results; we gain $+4.0$ points over the textual representation composed by just one lemma and $+11.2$ points gain compared to the settings when lemmas are not used at all.

B Dataset Details

We introduced a substantial addition of new instances across various datasets, achieving an overall 60% increase as shown in Figure 4.

Before our annotation process, significant gaps were present in the POS tags across various datasets, with certain categories, such as verbs, adjectives, and adverbs in the semeval2013 dataset, and adjectives and adverbs in the semeval2007 dataset, being completely absent. This is evident from the missing data points in Figure 3. Filling

Models	Params	ALL R@100 (Δ)
baseline	109M	96.5
- bert-base-uncased	109M	88.7 (-7.8)
- $E5_{small}$	33M	94.2 (-2.3)
- just main lemma	109M	92.5 (-4.0)
- no lemma	109M	85.3 (-11.2)

Table 4: Results in terms of the ablation study on the Retriever Module. Each row represents a change made to the baseline model and the corresponding impact on performance.

these gaps is crucial for constructing robust and comprehensive evaluation benchmarks. Incomplete datasets can lead to evaluations that fail to measure the true capabilities of language processing systems in real-world scenarios. Our annotation efforts were, therefore, critical in ensuring that all POS categories were fully represented, thereby enhancing the validity and reliability of subsequent system evaluations. Moreover, we preserved the distribution across POS tags as shown in Figure 2.

C Evaluation of SemCor with ConSeC$_{HEU}$ Annotations

Given the known limitations of SemCor due to its incomplete annotations, we have devised a mitigation strategy. By employing ConSeC$_{HEU}$, we aimed to identify and annotate all missing spans within SemCor, thereby creating an enhanced version, SemCor$_C$. This process, as detailed in Table 5, resulted in the addition of 133,727 new annotations to the existing 226,036 in SemCor. Subsequently, we assessed the performance of leading Word Sense Disambiguation systems ESC, BEM, and ConSeC using this enriched dataset. However, as indicated in Table 6, the enhancements in SemCor$_C$ did not necessarily translate into improved performance of the WSD models; with BEM achieving an F1 score of 79.0, ESCHER at 80.7, and ConSeC at 82.0. Upon transitioning to the SemCor$_C$ dataset, a noticeable decline in performance is observed for each model: BEM experiences a slight reduction to 78.8 (a 0.2 decrease), ESCHER to 80.3 (a 0.4 decrease), and ConSeC exhibits a more significant drop to 81.2 (a 0.8 decrease). We argue that this outcome originates from the 'silver' quality of the newly added annotations.

However, despite the quality of the new annotations in SemCor$_C$ not matching the 'gold' standard of the original dataset, it plays an important role

in our WSL setting. This integration improves the model's capacity for the production of accurate representation and identification of spans in the text.

D Comparison with EntQA

In this section, we present a detailed comparison between EntQA (Zhang et al., 2022b), and our model in the context of a WSL unconstrained environment. While EntQA represents a significant benchmark in the field, particularly for tasks similar to our WSL model, such as Entity Linking through the pipeline: candidate generation, Mention Detection (comparable to our Concept Detection), and Entity Disambiguation (analogous to our Word Sense Disambiguation), the two architectures diverge particularly in the reader part. Unlike EntQA, which processes candidates sequentially, our model can simultaneously handle all spans within an input sequence. We used the same Retriever for both reader models. In particular, we re-implemented the EntQA reader model.

The performance outcomes, as shown in Table 7, show that our model outperforms EntQA. Our model obtained an F1 score of 75.9 compared to EntQA's 72.0. This superiority in performance underscores the effectiveness of our model, especially in terms of recall and overall F1 score. We argue that, in our model, contextualizing all the candidates of a sentence together plays a crucial role in the performance gain. Moreover, our model not only outperforms EntQA but it is also more efficient. By processing all candidates together, our model significantly reduces processing times, in contrast to EntQA's sequential candidate method. This efficiency is quantitatively evident as our system processes the ALL$_{FULL}$ dataset in merely 17 seconds, a substantial improvement over the 63 seconds required by EntQA. To perform the comparison, we evaluated the same machine using an RTX 3090 for both experiments. This speed-up not only demonstrates our model's enhanced performance in terms of speed but also reinforces its practicality for integration into downstream tasks, further establishing our method's advantage in the field of WSL unconstrained settings.

E Qualitative analysis

In this section, we present a qualitative analysis of the model's output. We identified instances that highlight mismatches and inherent data discrepancies rather than direct errors in model processing.

	Dataset	Sentences	Tokens	Instances	New Instances
Train	SemCor	37176	820410	226036	-
	SemCor$_C$	37176	820410	359763	-
Evaluation	semeval2007	135	3219	455	941 (+206%)
	semeval2013	306	8533	1644	2194 (+133%)
	semeval2015	138	2643	1022	157 (+15%)
	senseval2	242	5829	2282	444 (+19%)
	senseval3	352	5640	1850	634 (+34%)
	all	1173	25864	7253	4370 (+60%)

Table 5: Statistics for training and evaluation corpora. The columns represent the number of sentences, the total number of tokens, the number of annotated terms, and the number of newly annotated instances added in each dataset.

Models	SemCor	SemCor$_C$
BEM	79.0	78.8 (-0.2)
ESCHER	80.7	80.3 (-0.4)
ConSeC	82.0	81.2 (-0.8)

Table 6: WSD F1 score results on the SemCor$_C$ the dataset containing the silver annotations annotations from ConSec$_{HEU}$.

	SE07			**ALL**$_{FULL}$		
Models	P	R	F1	P	R	F1
ConSeC$_{HEU}$	**76.7**	55.4	64.3	**80.4**	64.3	71.5
EntQA	75.1	64.7	69.5	78.4	66.5	72.0
Our Model	73.8	**74.9**	**74.4**	75.2	**76.7**	**75.9**

Table 7: Our model comparison with EntQA in the WSL task tested on ALL$_{FULL}$ dataset.

These cases highlight the complexities of matching model interpretations with established lexical databases such as WordNet. Some examples include:

- **Lexical Variants and Inventory Gaps:** The model can identify terms that are absent in the sense inventory, highlighting a gap between model recognition capabilities and standardized lexical entries. Some examples are shown in Table 8. For instance, in the sentence *"training and development of ageing workers in both the work place and the community,"* the model accurately identifies the span [work place] and annotates it with the sense *[a place where work is done]*. This instance reveals a mismatch due to the lexical variant "work place" not being directly mapped to its standard form "workplace" in WordNet.

- **Named Entities and Sense Attribution:** The model tends to abstract named entities into

broader conceptual categories, as shown in Table 9. For instance, in the sentence the sentence *"Trouble is following hard on the heels of the uproar around Josef Ackermann, CEO of Deutsche Bank,"* the model categorizes [Josef Ackermann] as *[the corporate executive responsible for the operations of the firm;]*. Although WordNet contains some named entities, specific ones like "Josef Ackermann" may not be explicitly available in the inventory. Moreover, in the Semcor training samples, often named entities are annotated with generic synsets like *person* or *location*, illustrating the challenge of capturing the full specificity of named entities within existing sense inventories.

These examples highlight WSL's challenges with lexical variations and named entity interpretation, emphasizing the importance of refining sense inventories and training methods for better alignment with lexical standards.

Example Text	WSL disambiguation
Training and development of ageing workers in both the work place and the community.	a place where work is done
In the amount USD 45 billion (nearly EUR 30 billion) in one go .	the basic monetary unit of most members of the European Union
Auditors found crookery the first day on the job.	verbal misrepresentation intended to take advantage of you in some way
Played on the 23rd of November against Ajax in European Champions League	- any number of entities (members) considered as a unit; - an active diversion requiring physical exertion and competition
Ctrl Q Quit Shuts the program.	cease to operate or cause to cease operating

Table 8: This table showcases examples of model's disambiguation capabilities and lexical recognition gaps, showing specific instances where the model accurately identifies and annotates lexical variants not directly mapped in standard sense inventories

Example Text	WSL disambiguation
Trouble is following hard on the heels of the uproar around Josef Ackermann, CEO of Deutsche Bank.	the corporate executive responsible for the operations of the firm;
In his program, François Hollande confines himself to banalities.	a human being
The World Labor Organisation estimates that for example in Germany..	an international alliance involving many different countries
Friendly game today, at 3:05 pm at the National Stadium in San Jose.	location, a point or extent in space
The two justices have been attending Federalist Society events for years.	any number of entities (members) considered as a unit

Table 9: This table showcases examples of how the model abstracts named entities into broader conceptual categories. Each row shows the model's disambiguation of specific named entities.
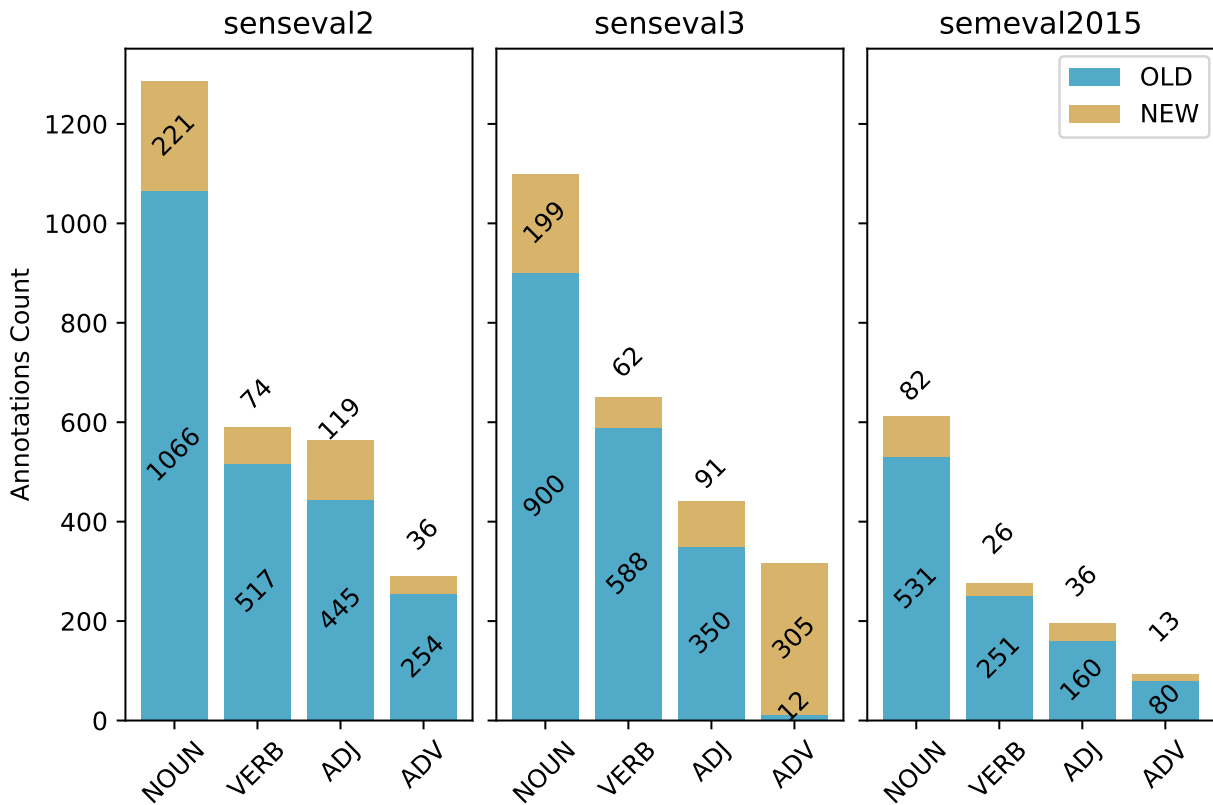
Figure 2: The counts of four POS categories (NOUN, VERB, ADJ, ADV) for three different datasets (Senseval2, Senseval3, Semeval2015). Each POS category is subdivided into 'OLD' (blue) and 'NEW' (orange) data points, indicating the frequency of each annotation before and after our comprehensive annotation process.
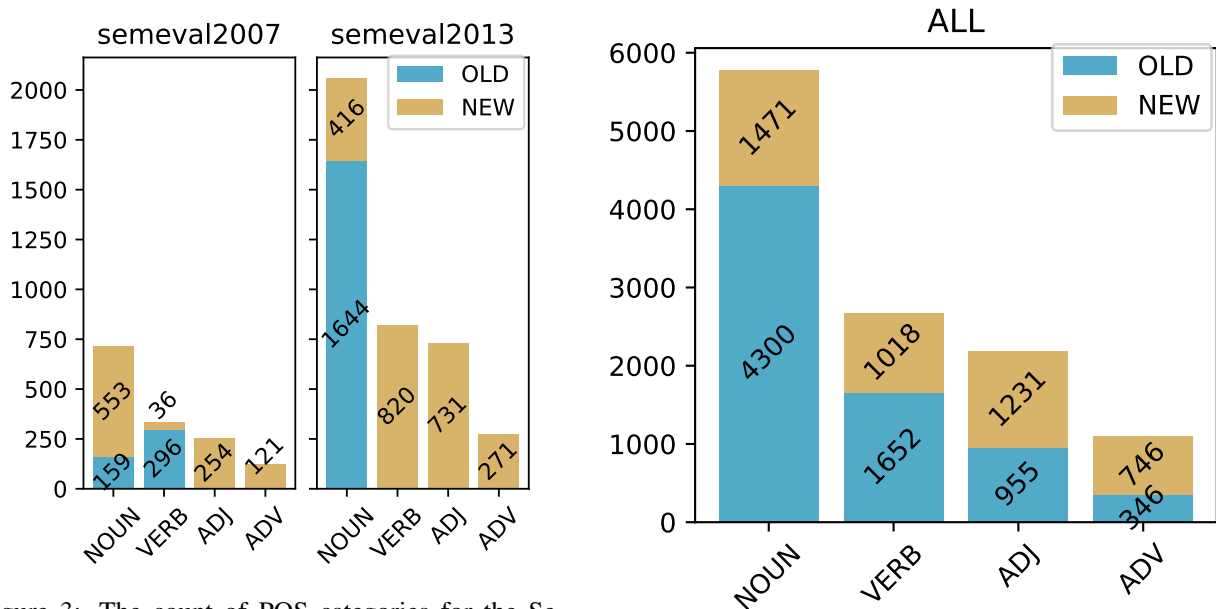

Figure 3: The count of POS categories for the Semeval2007 and Semeval2013 datasets. Notably, the original Semeval2007 dataset lacks annotations for ADJ and ADV categories, and Semeval2013 lacks annotations for VERB, ADJ, and ADV, as indicated by the absence of 'OLD' (blue) bars for these categories. The 'NEW' (orange) bars represent the counts post-annotation.

Figure 4: The counts of four POS categories within the 'ALL' dataset, which aggregates data across multiple sources. The 'OLD' (blue) bars represent the original annotation counts, while the 'NEW' (orange) bars indicate the increased counts following our extensive annotation process.