Insights from Transfer Learning Experiments with Word-in-Context and Word Sense Disambiguation Models

Alp Mujko, Dominik Schlechtweg
Institute for Natural Language Processing, University of Stuttgart
first.last@ims.uni-stuttgart.de

Abstract

We investigate the relationship between Word-in-Context (WiC) and Word Sense Disambiguation (WSD) by examining how training on one task (or both) affects performance on the other. Using established English datasets we train a sentence transformer with target-word highlighting and contrastive loss. Models are evaluated on WiC and WSD benchmarks across single-task, joint, and combined dataset configurations. Results show that joint training consistently improves or maintains WiC performance, particularly in low-resource settings, while WSD benefits mainly when annotated data is limited. Cross-task experiments demonstrate strong transfer: WSD-trained models generalize effectively to WiC, and WiC-trained models outperform baselines on WSD, indicating shared context-sensitive lexical representations. Combining multiple WiC datasets further enhances accuracy and stability. These findings highlight the complementary nature of WiC and WSD and demonstrate that unified training strategies can yield more robust and generalizable sense disambiguation models. The results provide practical guidance for designing datasets and models in multilingual and low-resource contexts, emphasizing the value of leveraging shared semantic representations.

Keywords: Word-in-Context, Word Sense Disambiguation, Joint Training, Cross-Task Learning, Transfer Learning

1. Introduction

Understanding word meaning in context is a core challenge in Natural Language Processing (NLP), essential for applications like machine translation, information retrieval, and text processing (Ide and Véronis, 1998). Two main tasks address this: Word-in-Context (WiC, Pilehvar and Camacho-Collados, 2019; Raganato et al., 2020), a binary classification task that determines if a word has the same meaning in two contexts, and Word Sense Disambiguation (WSD, Navigli, 2009), which assigns the most appropriate sense label from a predefined inventory. Both tasks capture lexical semantic nuances but differ in definition and have typically been studied separately (Cassotti et al., 2023; Barba et al., 2021). As NLP advances, examining the relationship between multitask models and approaches such as cross-task and transfer learning is increasingly important.

Recent advances in sentence transformers (Reimers and Gurevych, 2019) and the availability of new benchmarks (Wang et al., 2019) have significantly improved the modeling of contextual word meaning. This progress presents an opportunity to re-examine the relationship between WiC and WSD. Despite their conceptual overlap, the potential benefits of joint training or leveraging shared representations between these tasks remain largely unexplored, leaving a critical gap in our understanding of how to build more robust lexical-semantic models.

In this work, we bridge this gap by systematically investigating the effects of joint training on WiC and WSD using a unified sentence transformer. We combine several prominent WiC datasets (Pilehvar and Camacho-Collados, 2019; Martelli et al., 2021; Cassotti et al., 2023) with the FEWS WSD dataset (Blevins et al., 2021) and test various data integration strategies. By comparing models trained on individual versus combined data, we assess whether shared training leads to mutual benefits. Our results provide new insights into the relationship between WiC and WSD and support more generalizable lexical semantic models.

2. Related Work

Recent approaches, such as XL-LEXEME (Cassotti et al., 2023) or XL-DURel (Yadav and Schlechtweg, 2025), leverage a bi-encoder Sentence-BERT (SBERT) architecture (Reimers and Gurevych, 2019) that explicitly marks the target word in each sentence. This design enables efficient, comparable contextual representations and supports large-scale, multilingual applications, achieving state-of-the-art results on cross-lingual and diachronic WiC and lexical semantic change detection tasks (Schlechtweg et al., 2020, 2025).

While SBERT and similar bi-encoder models are standard for sentence-level semantic tasks, their application to supervised WSD, where a model must assign a sense from a predefined inventory, remains, to our knowledge, unexplored in the literature. In contrast, GlossBERT uses a BERT cross-

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10009–10019
11–16 May 2026. ©ELRA Language Resources Association (ELRA), 2026

encoder to score context–gloss pairs with WordNet definitions (and examples) and achieved state-of-the-art English all-words WSD when trained on SemCor within the Raganato et al. (2017) framework (Huang et al., 2019).

Several studies have examined the link between WiC and WSD. Hauer and Kondrak (2022) establish their theoretical equivalence, showing that methods for one task can be adapted to the others, though without empirical validation. In fact, the original WiC dataset (Pilehvar and Camacho-Collados, 2019) was constructed by repurposing sense-annotated data from traditional WSD resources, pairing example sentences for the same word and framed the task as a binary decision on sense identity, thus directly bridging WSD and WiC. Loureiro and Jorge (2019) demonstrate that WSD systems using contextual embeddings perform well on WiC without task-specific training, while Škvorc and Robnik-Šikonja (2025) show that dictionary examples and large language models can generate WiC-style data for low-resource languages, which can then be used for both WiC and WSD tasks. These works highlight the shared underlying representations and the potential for transfer learning between WiC and WSD.

Despite the established theoretical and empirical links between WiC and WSD, a systematic investigation into the effects of joint training remains a critical gap in the literature. Building on the success of bi-encoder architectures for WiC, this paper provides the first comprehensive analysis of how training on WiC, WSD, and their combination impacts performance. We specifically use an SBERT-based model to examine knowledge transfer and determine whether a unified training strategy can yield more robust and generalizable lexical semantic representations.

3. Tasks

This work addresses two core lexical semantic tasks: Word-in-Context and Word Sense Disambiguation.

Word-in-Context: Given two sentences with the same target word, decide if the sense is the same. For *"She can **book** a flight online"* and *"He read a fascinating **book**"*, the output is 0 (different senses).

Word Sense Disambiguation: Given a sentence and a predefined sense inventory, select the correct sense for a target word. For *"She can **book** a flight online"* with senses (1) To arrange in advance; (2) A written work, the output is 1.

Both tasks require assessing the semantic relationship between two text sequences. WiC compares a word's usage across two contexts, while WSD compares a usage against candidate sense definitions. This parallel structure motivates our unified modeling approach, which treats both as a sequence-comparison problem.

4. Hypotheses

Building on the theoretical links between WiC and WSD (Hauer and Kondrak, 2022), we investigate their practical interplay through joint and cross-task training. This leads us to formulate two central hypotheses.

Hypothesis 1: We hypothesize that jointly training models on both WiC and WSD data will lead to improved or at least non-detrimental performance on each individual task, compared to training on a single task alone.

Hypothesis 2: We further hypothesize that models trained exclusively on one task can generalize to the other, providing insight into the transferability and shared representations between WiC and WSD.

Through these experiments, our aim is to clarify whether joint or cross-task training offers tangible benefits for sense disambiguation, particularly in settings with limited annotated resources.

5. Data

For our experiments, we used several established datasets for both WiC and WSD tasks, focusing exclusively on English data.

The WiC data used in this work come from three main sources. The first is the **Original WiC dataset** (hereafter referred to as **Pil-WiC**) (Pilehvar and Camacho-Collados, 2019), which provides 5,428 training, 638 development, and 1,400 test English sentence pairs in tab-separated format. In each pair, a single target word appears in both sentences, accompanied by a label indicating whether the word has the same sense in both contexts.

The second source is **MCL-WiC** (Martelli et al., 2021), providing 8,000 training, 1,000 development, and 1,000 test English sentence pairs in JSON format. Similar to the original WiC dataset, each pair includes the same target word in both sentences.

Finally, we used the **XL-Lexeme** dataset (Cassotti et al., 2023), providing 13,428 training, 570 development, and 2,400 test English sentence pairs. This dataset was created by merging the training data from MCL-WiC, AM2ICO (Liu et al., 2021), and XL-WiC (Raganato et al., 2020). As with the other WiC datasets, each pair contains the same target word in both sentences.

For the WSD task, we use the **FEWS** dataset (Blevins et al., 2021). For training, FEWS originally provides 101,459 positive English instances in JSON format. To create a balanced (binary) training set of 202,918 instances, we pair each positive instance with a randomly selected, valid but contextually incorrect sense from the provided dictionary file (which lists all possible senses for each target word in the dataset), resulting in an equally distributed set of (sentence, gloss, label) triplets. For development and testing, we utilize the dataset's zero-shot splits. Applying the same balancing strategy used for the training data, we construct a zero-shot development set of 10,000 instances, equally distributed between positive and negative labels. The zero-shot test set comprises 5,000 target word usages, where each usage is paired with its possible candidate senses.

In some experiments, FEWS is downsampled to match the size of each WiC dataset, while in others, we use the full training dataset (over 200,000 elements) to study the effect of larger-scale WSD data. To ensure a balanced training signal in our joint-training experiments, the downsampling strategy groups the FEWS dataset by unique `line_id` identifiers, which represent specific target word usages. We then randomly sample these usage groups until the required dataset size is reached. Because a fixed random seed is utilized during the sampling process, the resulting downsampled WSD subsets are deterministic and identical across multiple experimental runs of the same size.

To better understand dataset relationships, we compute a sentence overlap matrix shown in Appendix A, which illustrates the proportion of shared sentences between datasets. Overlap is observed only within individual datasets, reflecting reused sentences across different data elements, but not reused data pairs. Due to the composition of XL-Lexeme, which integrates examples from multiple WiC sources, there is notable overlap between XL-Lexeme and both Pil-WiC and MCL-WiC, while no overlap occurs directly between Pil-WiC and MCL-WiC.

6. Model

Our experimental setup uses a Sentence Transformer model (Reimers and Gurevych, 2019) built upon the `xlm-roberta-large` architecture (Conneau et al., 2020). By applying a mean pooling layer to the model's final hidden state, we generate sentence embeddings that enable efficient encoding for semantic similarity and classification tasks. The model was trained using Contrastive Loss with a margin of 0.5. Additional model parameters and hyperparameters are detailed in Appendix B.

To help the model focus on relevant information, we insert special tokens `<t>` and `</t>` around the target word in each sentence, following Cassotti et al. (2023), as well as around the entire definition in the WSD data. The goal of this marking is to encourage the model to attend specifically to the relevant spans. We extend the model's vocabulary with these special tokens, in addition to standard special tokens (such as `[CLS]` and `[SEP]`), ensuring that the model can recognize and process them appropriately during training and inference.

At inference time, predictions are made by comparing sentence embeddings. For the WiC task, the model computes the cosine similarity between the generated embeddings of the two sentence contexts. The final binary label is determined using an optimal similarity threshold derived from the respective development set. For the WSD task, inference is framed as a nearest-neighbor retrieval problem. The model generates a vector embedding for the context sentence containing the marked target word, as well as separate embeddings for each candidate sense definition provided by the predefined inventory. We compute the cosine similarity between the context embedding and each candidate definition embedding, selecting the sense that yields the highest similarity score.

To contextualize model performance, we include two baselines: an untrained version of the model and random selection. For random selection, WiC instances are assigned a label randomly between 0 and 1, and for WSD instances, the label is chosen randomly among the n candidate definitions. We do not include a majority class baseline for either task; the standard WiC test sets are perfectly balanced (making a majority baseline functionally identical to random chance at .500), and our WSD evaluation relies exclusively on the zero-shot splits of FEWS. Because zero-shot target senses are completely unseen during training, calculating a traditional Most Frequent Sense (MFS) or majority class baseline is impossible.

7. Experiments

For each of the three main WiC datasets, we train models using: (a) the WiC dataset alone, (b) the WiC dataset combined with a downsampled FEWS WSD dataset (with FEWS sizes matched to the respective WiC set), and (c) the WiC dataset combined with the full FEWS dataset. Additionally, we create a **combined WiC training set** of size 15K, merging all three WiC datasets into a single training corpus (with duplicates removed), and train models on this merged set, both alone and in combination with FEWS.

For comparison, we also train models on FEWS alone, using both the full dataset (200K elements) and the corresponding downsampled versions (5K,

8K, 13K or 15K elements).

All models were evaluated using the accuracy metric on both test sets: the corresponding WiC test set (cf. Pilehvar and Camacho-Collados, 2019) and, across all training configurations, the same zero-shot test split for WSD.

During training, development sets are used to monitor validation loss and trigger early stopping with a patience of two epochs. Crucially, the composition of the development set strictly mirrors the nature of the training data for each specific experiment. For single-task models, evaluation is performed exclusively on the corresponding task's dev set (e.g., the standard WiC dev set for WiC-only training, or a binarized FEWS dev set for WSD-only training). In joint training configurations, the development set is constructed by concatenating the full WiC dev set with a proportionally downsampled, binarized FEWS dev set. This dynamic alignment ensures that the early stopping criterion strictly optimizes for the task or combination of tasks the model is actively learning.

Importantly, all models were trained using the same training parameters; only the training data was varied between runs. Although it is likely that better results could be achieved by tuning hyperparameters for each configuration individually, we deliberately kept the training setup fixed in order to isolate and investigate the impact of the data itself on model performance.

Each experiment was performed three times, and the reported results correspond to the mean accuracy across the three runs.

This experimental setup allows us to systematically assess the effects of joint training, dataset size, and cross-task transfer on sense disambiguation performance.[1]

8. Results and Discussion

Table 1 reports mean accuracies on two evaluation sets: WiC_* and WSD. The table is organized into three blocks, each representing a WiC dataset: Pil-WiC, MCL-WiC, and XL-Lexeme. Within each block, rows correspond to different training configurations, including single-dataset and combined-dataset training. For WiC_*, each model is tested on the WiC dataset corresponding to the block it belongs to, so comparisons are made fairly within each group. WSD is evaluated on a shared, standardized test set for all models, regardless of their training data. Bold values indicate the highest accuracy within each block. This organization emphasizes how varying combinations of WiC and WSD training data influence performance on both WiC and WSD evaluations.

[1]We report use of computational resources and AI assistance in Appendix C.

	Train	WiC_*	WSD
Pil-WiC	P + F_{200K}	$\mathbf{.714}^{+.006}_{-.004}$	$.665^{+.005}_{-.003}$
	P + F_{5K}	$.695^{+.014}_{-.008}$	$.552^{+.026}_{-.020}$
	P	$.681^{+.008}_{-.005}$	$.475^{+.004}_{-.004}$
	F_{200K}	$.693^{+.006}_{-.007}$	$\mathbf{.670}^{+.010}_{-.006}$
	F_{5K}	$.675^{+.009}_{-.012}$	$.539^{+.005}_{-.009}$
	Untrained	$.529^{+.000}_{-.000}$	$.309^{+.000}_{-.000}$
	Random	$.492^{+.010}_{-.005}$	$.262^{+.010}_{-.009}$
	C + F_{200K}	$.711^{+.019}_{-.011}$	$.660^{+.003}_{-.002}$
	C + F_{15K}	$.721^{+.010}_{-.013}$	$.617^{+.003}_{-.002}$
	F_{15K}	$.688^{+.010}_{-.013}$	$.604^{+.010}_{-.011}$
	C	$.703^{+.016}_{-.015}$	$.503^{+.024}_{-.033}$
MCL-WiC	M + F_{200K}	$.891^{+.009}_{-.013}$	$.663^{+.004}_{-.007}$
	M + F_{8K}	$.890^{+.005}_{-.010}$	$.575^{+.004}_{-.003}$
	M	$.889^{+.003}_{-.005}$	$.477^{+.007}_{-.010}$
	F_{200K}	$.860^{+.002}_{-.002}$	$\mathbf{.670}^{+.010}_{-.006}$
	F_{8K}	$.829^{+.030}_{-.020}$	$.527^{+.011}_{-.006}$
	Untrained	$.653^{+.000}_{-.000}$	$.309^{+.000}_{-.000}$
	Random	$.479^{+.011}_{-.020}$	$.262^{+.010}_{-.009}$
	C + F_{200K}	$.885^{+.001}_{-.003}$	$.660^{+.003}_{-.002}$
	C + F_{15K}	$.894^{+.004}_{-.005}$	$.617^{+.003}_{-.002}$
	F_{15K}	$.855^{+.012}_{-.019}$	$.604^{+.010}_{-.011}$
	C	$\mathbf{.906}^{+.011}_{-.007}$	$.503^{+.024}_{-.033}$
XL-Lexeme	X + F_{200K}	$.782^{+.002}_{-.003}$	$.669^{+.011}_{-.013}$
	X + F_{13K}	$\mathbf{.792}^{+.003}_{-.004}$	$.609^{+.007}_{-.005}$
	X	$.790^{+.005}_{-.006}$	$.472^{+.004}_{-.004}$
	F_{200K}	$.754^{+.006}_{-.004}$	$\mathbf{.670}^{+.010}_{-.006}$
	F_{13K}	$.759^{+.003}_{-.003}$	$.605^{+.010}_{-.015}$
	Untrained	$.585^{+.000}_{-.000}$	$.309^{+.000}_{-.000}$
	Random	$.508^{+.010}_{-.014}$	$.262^{+.010}_{-.009}$
	C + F_{200K}	$.781^{+.012}_{-.007}$	$.660^{+.003}_{-.002}$
	C + F_{15K}	$.791^{+.008}_{-.007}$	$.617^{+.003}_{-.002}$
	F_{15K}	$.758^{+.008}_{-.014}$	$.604^{+.010}_{-.011}$
	C	$.786^{+.008}_{-.007}$	$.503^{+.024}_{-.033}$

Table 1: Abbreviations: P = Pil-WiC, M = MCL-WiC, X = XL-Lexeme, F = FEWS (with subscript denoting sample size, e.g., F_{200K}), and C = Combined WiC dataset. Entries such as "P + F_{5K}" indicate joint training on Pil-WiC and FEWS_{5K}. Mean test accuracies are averaged over three runs for all training configurations and baselines. Each block corresponds to a WiC dataset (Pil-WiC, MCL-WiC, XL-Lexeme), with results reported on the dataset's own WiC test set (WiC_*) and the shared WSD evaluation benchmark. **Bold** values mark the highest mean accuracy per test set within each group. These results illustrate the effects of joint training, data scale, and cross-dataset combination on performance across WiC and WSD tasks. Superscript and subscript values indicate the maximum and minimum observed accuracy deviations from the mean across the three runs.

8.1. Joint Training Effects (Hypothesis 1)

WiC Task Performance: Across all datasets, joint training with smaller WSD subsets consistently improves WiC accuracy compared to single-task WiC training. Gains are modest but reliable: Pil-WiC improves from .681 to .695 (+.014), MCL-WiC from .889 to .890 (+.001), and XL-Lexeme from .790 to .792 (+.002). With the full $FEWS_{200K}$, results are mixed: Pil-WiC achieves its best WiC accuracy of .714 (+.033), MCL-WiC sees only a marginal increase to .891, (+.002), while XL-Lexeme declines slightly to .782 (–.008).

The previous results showed that joint training with WSD subsets can enhance WiC performance, but it remains unclear whether these gains arise from the semantic signal provided by WSD or simply from exposure to more data. To disentangle these effects, we examine models trained on the Combined WiC dataset (C), which merges all WiC sources without any WSD augmentation. This setup allows us to test whether increasing the amount and diversity of WiC data alone accounts for the improvements seen in joint training.

The Combined model achieves .703 on Pil-WiC, .906 on MCL-WiC, and .786 on XL-Lexeme, consistently outperforming single-WiC-dataset baselines except for XL-Lexeme, where performance remains slightly lower than the best single-WiC-source model. Crucially, when compared to the joint-training models discussed above, the Combined model actually outperforms joint training with small WSD subsets on Pil-WiC (.703 vs .695) and MCL-WiC (.906 vs .890). However, on XL-Lexeme, adding a WSD subset remains superior (.792 vs .786). Thus, expanding WiC coverage alone can match or exceed cross-task augmentation benefits, though WSD provides distinct semantic advantages for specific datasets.

When WSD data are added to the Combined WiC model, evaluating both the $FEWS_{15K}$ and $FEWS_{200K}$ augmentations, the results show that the supplementary semantic signal improves performance in exactly half of the evaluated cases (3 out of 6). For Pil-WiC, adding WSD data consistently improves upon the Combined baseline (.703), reaching .721 with the 15K subset and .711 with the full 200K set. For XL-Lexeme, adding the smaller 15K subset provides a moderate gain (.786 to .791), but the full 200K dataset causes a slight decline (.781). Conversely, for MCL-WiC, adding WSD data at either scale degrades performance compared to the Combined model alone, dropping from .906 to .894 (15K) and .885 (200K). These mixed results demonstrate that while WSD augmentation can still provide targeted benefits on top of a maximized WiC training set, it can also introduce task interference, particularly when the model is already performing near the ceiling.

WSD Task Performance: For the WSD task, joint training proves beneficial only when in-domain data is limited. Compared to their respective WSD-only baselines, Pil-WiC + $FEWS_{5K}$ provides an accuracy gain of +.013 (reaching .552), MCL-WiC + $FEWS_{8K}$ yields a substantial +.048 gain (reaching .575), and XL-Lexeme + $FEWS_{13K}$ offers a slight +.005 gain (reaching .609). Conversely, at full scale, cross-task augmentation becomes detrimental. To directly parallel the WiC analysis above, $FEWS_{200K}$ can be viewed as the WSD-side analogue of the Combined WiC dataset, the condition where all available task-specific data is utilized without cross-task augmentation. In this setting, larger data volume alone yields the highest absolute WSD score (.670). Adding WiC data at this scale causes performance to degrade slightly across the board: all joint-training configurations incorporating $FEWS_{200K}$ fall short of the single-task baseline, with scores ranging from .660 (when paired with the Combined WiC dataset) to .669 (when paired with XL-Lexeme). This confirms that for WSD, data scale dominates once sufficient coverage is reached.

Hypothesis 1 Evaluation: Our hypothesis that joint training would improve or not harm performance on WiC and WSD tasks is partially supported by the results in Table 1. The outcome is nuanced, depending heavily on the specific task and data scale.

For the WiC task, the hypothesis is largely supported. Joint training with WSD data, especially smaller subsets, consistently improves WiC accuracy (e.g., Pil-WiC + $FEWS_{5K}$ improves accuracy by +.014). The highest performance on the Pil-WiC test set (.721) is achieved with a joint training model (Combined + $FEWS_{15K}$), confirming that the WSD signal is beneficial even with diverse WiC data. However, the effect is not universally positive, as joint training with the full $FEWS_{200K}$ dataset slightly degrades performance on XL-Lexeme (–.008).

For the WSD task, the hypothesis is only supported when WSD data is limited. In these settings, adding WiC data consistently improves performance (e.g., MCL-WiC + $FEWS_{8K}$ boosts accuracy by +.048 over $FEWS_{8K}$ alone). However, the hypothesis is falsified in the large-data regime. The top WSD score (.670) is achieved with single-task training on the full $FEWS_{200K}$ dataset, which outperforms all joint training configurations. This indicates that at a large scale, an auxiliary task is detrimental to WSD performance.

In summary, Hypothesis 1 is only partially supported. The benefits of joint training are not universal but instead depend on a complex interplay between the primary task and data availability. For WiC, joint training is broadly beneficial, while for WSD, it is only helpful when in-task data is scarce.

Crucially, this observed difference may stem from the inherent size disparity between the available datasets. Because the WiC datasets, even in the Combined condition, are substantially smaller than the full WSD corpus, the WiC task may consistently benefit from joint training simply because it perpetually remains in a lower-resource regime.

These results may suggest a dynamic wherein the tasks provide asymmetric benefits. The simpler, binary WiC task may benefit from the fine-grained semantic supervision offered by WSD, which could in turn enrich the model's contextual representations. Conversely, the more complex WSD task may leverage the broader contextual exposure from WiC data as a form of regularization, an effect that is most pronounced when its own labeled data is sparse. However, when WSD data is abundant, the WiC task's simpler objective may act as a distraction, potentially preventing the model from learning the specific features required for fine-grained sense distinction and thus hindering its performance. Alternatively, rather than an inherent task asymmetry, these dynamics might simply reflect the disparity in training data volume. It is highly plausible that both tasks share identical underlying mechanics for joint training, but we only observe task interference on WSD because it is the only task with sufficient data ($FEWS_{200K}$) to reach the saturation point where auxiliary data becomes a detriment rather than a benefit.

8.2. Cross-Task Generalization (Hypothesis 2)

WiC Performance: Models trained solely on WSD data generalize effectively to the WiC task. The $FEWS_{200K}$-trained model achieves competitive scores across all datasets, even outperforming the in-domain WiC model on Pil-WiC (.693 vs. .681). However, this transferability does not uniformly scale with the amount of WSD data across all targets. For XL-Lexeme, smaller WSD subsets actually yield better WiC transfer performance than the full F_{200K} dataset (.759 with F_{13K} vs. .754 with F_{200K}). In contrast, for both the Pil-WiC and MCL-WiC datasets, WiC transfer performance scales directly with the volume of WSD training data, with MCL-WiC improving from .829 (F_{8K}) to .860 (F_{200K}).

WSD Performance: Conversely, WiC-trained models also generalize to WSD, albeit with lower absolute scores than WSD-specific training. Pil-WiC achieves .475, MCL-WiC reaches .477, and XL-Lexeme records .472. Although these numbers fall short of models trained even on small WSD subsets (e.g., $FEWS_{5K}$ achieving .539) and significantly behind the large-scale $FEWS_{200K}$ model (.670), they remain well above the untrained model (.309) and random baseline (.262). The Combined WiC dataset achieves the best transfer performance among the WiC-only models, reaching .503, likely due to its larger size and data diversity.

Hypothesis 2 Evaluation: The results provide strong support for Hypothesis 2, showing significant bidirectional generalization between WiC and WSD. All single-task models substantially outperform baselines on the opposite task, confirming a shared representational foundation.

Transfer is particularly strong from WSD to WiC. The $FEWS_{200K}$-trained model, for instance, scores .693 on Pil-WiC, outperforming the in-domain WiC model (.681). This highlights that representations from large-scale WSD training are highly effective for WiC. In contrast, WiC-trained models also generalize to WSD, with scores like .503 from the Combined dataset far exceeding baselines (.309 untrained, .262 random), which is notable given their much smaller training size.

Overall, the findings validate bidirectional transferability but reveal a distinct asymmetry. The superior performance of WSD-to-WiC transfer is likely driven by a dual effect of dataset scale and the nature of task supervision. First, consistent with our earlier discussion of joint training, the $FEWS_{200K}$ dataset provides a massive volume advantage over the available WiC corpora. Second, while both tasks operate on a binary objective, WSD requires the model to anchor instances to explicit, external sense definitions. This provides richer semantic supervision compared to WiC's relative context-matching objective, yielding robust representations that transfer highly effectively to the WiC task.

8.3. Effects in Resource-Constrained Settings

Joint training proves particularly valuable in resource-constrained environments, where combining smaller WiC and WSD datasets consistently yields superior performance compared to training on individual datasets. For example, MCL-WiC + $FEWS_{8K}$ outperforms both single-task models, achieving .890 on WiC_* (vs .889 for MCL-WiC alone and .829 for $FEWS_{8K}$) and .575 on WSD (vs .477 and .527). Similarly, XL-Lexeme + $FEWS_{13K}$ surpasses its components with .792 and .609, compared to XL-Lexeme alone (.790, .472) and $FEWS_{13K}$ alone (.759, .604). Pil-WiC + $FEWS_{5K}$ also demonstrates consistent gains, reaching .695, and .552, exceeding both Pil-WiC alone (.681 and .475) and $FEWS_{5K}$ alone (.675, .539). These improvements are most pronounced on WSD, while gains on WiC_* are modest but positive.

The benefits of joint training diminish as WSD

training data increases. With $FEWS_{200K}$, single-task WSD training delivers the highest WSD score overall (.670), and joint training can slightly reduce WiC accuracy for some datasets (e.g., XL-Lexeme at .782), while others improve or remain stable (PiL-WiC at .714, MCL-WiC at .891).

Overall, the effectiveness of joint training in low-resource settings highlights the value of supplementing small, task-specific datasets. However, because our experiments introduce both new textual contexts and an auxiliary objective simultaneously, it is difficult to definitively disentangle whether these gains stem from complementary task supervision, or simply from the sheer increase in overall training data volume. Regardless of the exact underlying mechanism, these findings confirm that merging available cross-task data offers a highly practical strategy for enhancing performance when annotated resources are scarce.

8.4. Statistical Significance and Limitations

To ensure full transparency regarding performance variance, we report the maximum and minimum accuracy bounds alongside the mean for all configurations (Table 1), and we release the exact accuracy scores for every individual run as a supplementary CSV file in our public repository [2]. To address the significance of our discussed accuracy gains, we conducted independent two-sample t-tests on key configuration pairs, provided in Appendix D. However, we note important limitations regarding this statistical analysis. Because our experiments are computationally constrained, each configuration is averaged over exactly three unseeded runs ($n = 3$). At this sample size, statistical tests suffer from notably low statistical power, making it difficult to detect true performance gains as statistically significant (e.g., small, consistent gains of .002 are naturally not statistically significant with $n = 3$). Furthermore, standard t-tests assume data normality and independence, assumptions that are difficult to confidently verify with only three observations. We also refrain from performing multiple comparison corrections (e.g., Bonferroni) as it would be overly conservative given the already underpowered sample size. Therefore, while we report these p-values for completeness, our broader conclusions heavily rely on the consistency of the observed directional trends across different datasets and evaluation setups.

[2]Supplementary CSV files are available at: https://github.com/alpmu/wic-wsd-transfer-insights

9. Conclusion

In this work, we systematically investigated the relationship between Word-in-Context and Word Sense Disambiguation by analyzing how different combinations of training data affect model performance across both tasks. We evaluated models trained on WiC data, WSD data, and various joint configurations, using both downsampled and full-scale datasets, and assessed performance on standard WiC and standard WSD benchmarks.

Our results demonstrate that joint training with both WiC and WSD data generally improves or maintains WiC performance, particularly when the datasets are relatively small, though gains are not uniform across all datasets. For WSD, the strongest results are achieved with large-scale, single-task training, while joint training provides clear benefits primarily in low-resource scenarios, helping models generalize better when annotated data is limited. We also found strong cross-task generalization: models trained on one task could transfer knowledge to the other, with the full WSD-trained model even outperforming an in-domain WiC-trained model in some cases.

These findings suggest that WiC and WSD share underlying semantic representations, and that leveraging data from both tasks is highly advantageous in resource-constrained scenarios. Furthermore, the exceptional strength of WSD-to-WiC transfer indicates that the explicit sense-anchoring required by WSD provides richer semantic supervision than WiC's relative context matching. However, our large-scale experiments also reveal a limit to multi-task benefits: once a primary task reaches sufficient data saturation, introducing an auxiliary objective can act as a distraction and slightly degrade performance. Overall, our study highlights the value of unified approaches to sense disambiguation and provides practical insights for designing models in multilingual and low-resource contexts.

In the future, it would be interesting to test variations of the particular model implementations we chose in this study. For instance, special tokens were applied in word usages to mark the target word and in sense definitions to mark the entire definition. One could test variations where the target word is added to the definitions with the special tags and an additional connecting construction. Further, we would like to relate our results to large language models. These are usually optimized for specific tasks by prompt optimization, which may include few-shot examples. We wonder whether including WiC and WSD examples into the same prompt helps models to generalize better to one of the tasks.

Limitations

We tested our hypotheses on the relationship between WiC and WSD using a narrow set of resources and methods. Specifically, we worked with three WiC datasets, one of which was largely a combination of the other two, and a single WSD dataset, and relied solely on a bi-encoder architecture. This limited scope may affect the generalizability of our findings, as it remains unclear whether different datasets, larger or more diverse data sources, or alternative architectures such as cross-encoders would produce similar patterns of joint training benefits and cross-task transfer. Future work should broaden both the dataset coverage and model architectures to better understand the robustness and limits of these effects.

Acknowledgements

We thank the reviewers for constructive criticism and insightful feedback that helped to improve this paper.

10. Bibliographical References

Edoardo Barba, Luigi Procopio, and Roberto Navigli. 2021. 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.

Terra Blevins, Mandar Joshi, and Luke Zettlemoyer. 2021. FEWS: Large-scale, low-shot word sense disambiguation with the dictionary. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 455–465, Online. Association for Computational Linguistics.

Pierluigi Cassotti, Lucia Siciliani, Marco DeGemmis, Giovanni Semeraro, and Pierpaolo Basile. 2023. XL-LEXEME: WiC pretrained model for cross-lingual LEXical sEMantic changE. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1577–1585, Toronto, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Bradley Hauer and Grzegorz Kondrak. 2022. WiC = TSV = WSD: On the equivalence of three semantic tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2478–2486, Seattle, United States. Association for Computational Linguistics.

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

Nancy Ide and Jean Véronis. 1998. Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40.

Qianchu Liu, Edoardo Maria Ponti, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2021. AM2iCo: Evaluating word meaning in context across low-resource languages with adversarial examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7151–7162, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Federico Martelli, Najla Kalach, Gabriele Tola, and Roberto Navigli. 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, Online. Association for Computational Linguistics.

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

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset

for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. 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.

Alessandro Raganato, Tommaso Pasini, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2020. XL-WiC: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, Online. 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.

Dominik Schlechtweg, Tejaswi Choppa, Wei Zhao, and Michael Roth. 2025. CoMeDi shared task: Median judgment classification & mean disagreement ranking with ordinal word-in-context judgments. In *Proceedings of Context and Meaning: Navigating Disagreements in NLP Annotation*, pages 33–47, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Sachin Yadav and Dominik Schlechtweg. 2025. XL-DURel: Finetuning sentence transformers for ordinal word-in-context classification. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 338–351, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Tadej Škvorc and Marko Robnik-Šikonja. 2025. Solving word-sense disambiguation and word-sense induction with dictionary examples.

A. Dataset Statistics

Figure 1 shows statistics on dataset overlap on the sentence level.

B. Hyperparameters

The same training parameters were used for all models to ensure comparability. Table 2 lists the hyperparameters applied in all experiments.

Hyperparameter	Value
Loss function	Contrastive (margin = 0.5)
Number of epochs	15
Batch size	24
Learning rate	1×10^{-5}
Weight decay	0.001
Warmup ratio	0.1
Batch sampler	NO DUPLICATES
Early stopping patience	5
Metric for best model	Cosine Accuracy

Table 2: Main hyperparameters used for model training.

C. Computational Resources & AI Assistance

All experiments were executed on a Linux-based server running Fedora 42, equipped with NVIDIA RTX A6000 GPUs (48 GB VRAM each) and dual Intel Xeon CPUs. Each run utilized a single GPU. The total computational time for all experiments was approximately 50–60 GPU-hours. The preparation of this text and coding were supported by OpenAI's ChatGPT and GitHub Copilot.

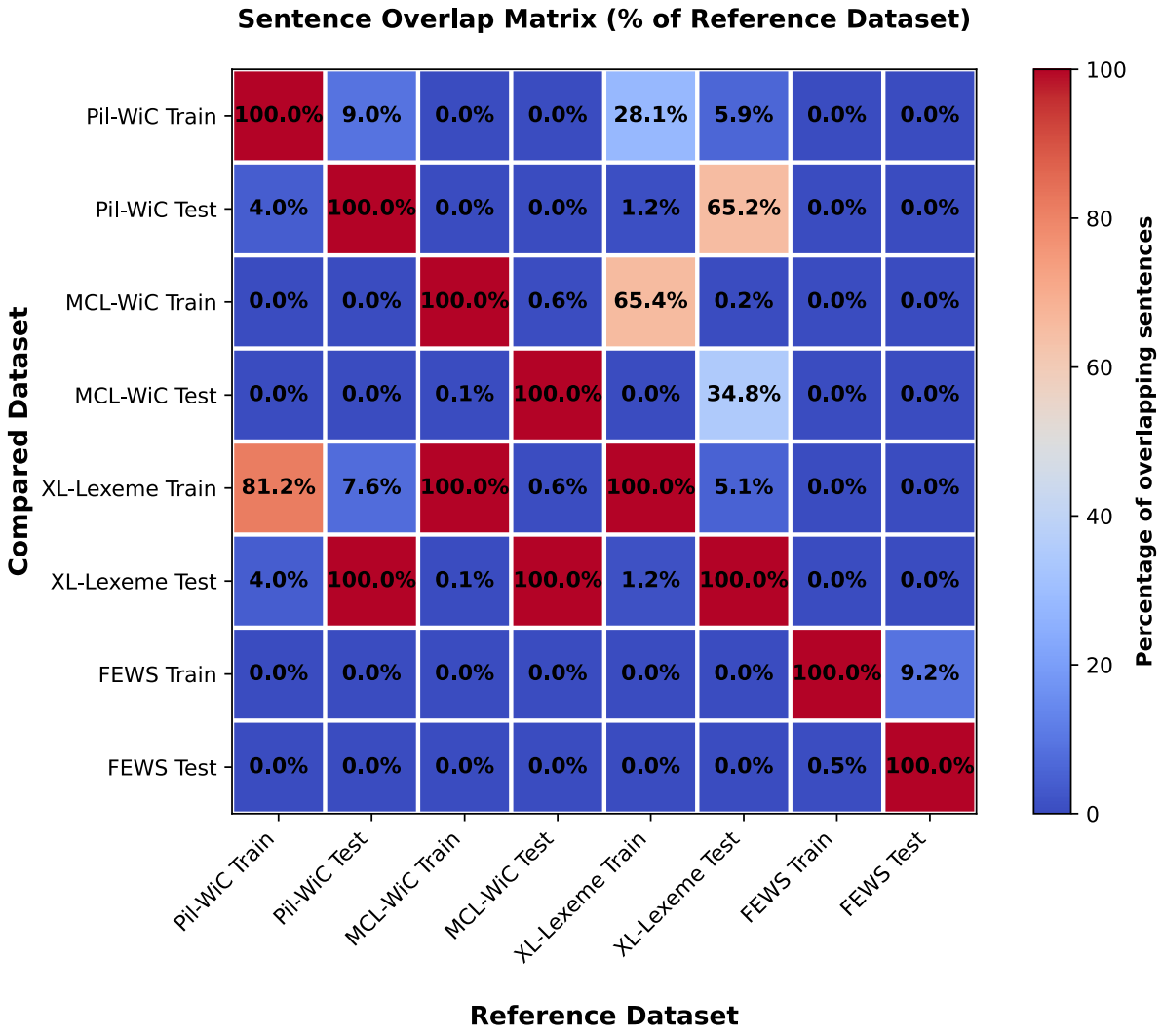

Figure 1: Sentence overlap matrix among datasets used in the experiments. Each cell indicates the percentage of sentences from the dataset on the x-axis that also occur in the dataset on the y-axis. The matrix includes WiC datasets (Pil-WiC, MCL-WiC, XL-Lexeme) and the FEWS WSD dataset (train and test splits).

D. Statistical Significance of the Results

To supplement the main findings, Table 3 provides the independent two-sample t-tests for key configuration pairs. The table reports the mean accuracy of each configuration, the absolute difference in means (Δ Mean), the t-statistic, and the corresponding p-value. Please note that due to computational constraints, each configuration represents an average of three runs ($n = 3$). As a result, the statistical power is low, and these metrics should be interpreted alongside the directional trends discussed in the main text.

Test Set	Config A	Config B	Mean A	Mean B	Δ Mean	t-stat	p-value
Adding Limited WSD Data							
Pil-WiC	P + F_{5K}	P	.695	.681	+0.014	+1.74	0.171
MCL-WiC	M + F_{8K}	M	.890	.889	+0.001	+0.18	0.870
XL-Lexeme	X + F_{13K}	X	.792	.790	+0.002	+0.59	0.590
Adding Full WSD Data							
Pil-WiC	P + F_{200K}	P	.714	.681	+0.033	+6.20	0.004*
MCL-WiC	M + F_{200K}	M	.891	.889	+0.002	+0.32	0.771
XL-Lexeme	X + F_{200K}	X	.782	.790	-0.008	-2.42	0.099
Merging WiC Datasets							
Pil-WiC	C	P	.703	.681	+0.022	+2.21	0.120
MCL-WiC	C	M	.906	.889	+0.017	+2.68	0.083
XL-Lexeme	C	X	.786	.790	-0.004	-0.82	0.464
WSD Signal vs. WiC Volume							
Pil-WiC	C	P + F_{5K}	.703	.695	+0.008	+0.69	0.528
MCL-WiC	C	M + F_{8K}	.906	.890	+0.016	+2.07	0.109
XL-Lexeme	C	X + F_{13K}	.786	.792	-0.007	-1.36	0.268
Augmenting Combined WiC							
Pil-WiC	C + F_{15K}	C	.721	.703	+0.018	+1.59	0.193
Pil-WiC	C + F_{200K}	C	.711	.703	+0.008	+0.63	0.564
MCL-WiC	C + F_{15K}	C	.894	.906	-0.012	-1.92	0.158
MCL-WiC	C + F_{200K}	C	.885	.906	-0.021	-3.59	0.060
XL-Lexeme	C + F_{15K}	C	.791	.786	+0.005	+0.84	0.447
XL-Lexeme	C + F_{200K}	C	.781	.786	-0.005	-0.64	0.560
Adding WiC (Low-Res WSD)							
FEWS zero shot	P + F_{5K}	F_{5K}	.552	.539	+0.013	+0.87	0.461
FEWS zero shot	M + F_{8K}	F_{8K}	.575	.527	+0.048	+7.88	0.008*
FEWS zero shot	X + F_{13K}	F_{13K}	.609	.605	+0.005	+0.56	0.616
Adding WiC (High-Res WSD)							
FEWS zero shot	P + F_{200K}	F_{200K}	.665	.670	-0.005	-0.84	0.463
FEWS zero shot	M + F_{200K}	F_{200K}	.663	.670	-0.007	-1.14	0.327
FEWS zero shot	X + F_{200K}	F_{200K}	.669	.670	-0.001	-0.13	0.901
Zero-Shot: WSD → WiC							
Pil-WiC	F_{200K}	P	.693	.681	+0.012	+2.09	0.106
Transfer Scaling: WSD → WiC							
XL-Lexeme	F_{13K}	F_{200K}	.759	.754	+0.005	+1.35	0.273
MCL-WiC	F_{8K}	F_{200K}	.829	.860	-0.030	-1.99	0.183
Zero-Shot: WiC → WSD							
FEWS zero shot	P	F_{5K}	.475	.539	-0.064	-12.96	$< 0.001^*$
FEWS zero shot	P	F_{200K}	.475	.670	-0.195	-34.93	$< 0.001^*$
WiC → WSD Baselines							
FEWS zero shot	P	Untrained	.475	.309	+0.166	+69.57	$< 0.001^*$
FEWS zero shot	C	Untrained	.503	.309	+0.194	+11.42	0.008*

Table 3: Independent two-sample t-tests for key comparisons discussed in Section 8. Δ Mean represents the absolute test accuracy difference between Config A and Config B. Due to computational constraints restricting the sample size ($n = 3$), statistical power is notably low. Signifier * denotes $p < 0.01$.