

Evaluating Model-Task Fit in Arabic Word Sense Disambiguation

Yousef Younes
Universität Ulm
Germany
yousef.younes@uni-ulm.de

Abdelhalim Dahou
GESIS - Leibniz Institute for
the Social Sciences
Germany
Abdelhalim.dahou@
gesis.org

Brigitte Mathiak
GESIS - Leibniz Institute for
the Social Sciences
Germany
brigitte.mathiak@
gesis.org

Abstract

Word Sense Disambiguation (WSD) is the computational task of identifying a word’s intended meaning within a specific context when the word exhibits lexical ambiguity. It remains a core and challenging problem in Natural Language Processing (NLP). We formulate the problem in two ways: binary classification (context-gloss alignment) and multi-choice selection (identifying the correct sense from candidate glosses) using Large Language Models (LLMs) and Small Language Models (SLMs). Our results on the KSAA-CAD shared task dataset show a critical task-model dependency, where fine-tuned SLMs excel in binary classification (mBERT achieving 83.74% F1), while LLMs show strong performance on multi-choice (fine-tuned Gemma2 7B attaining 84.26% F1). Notably, Camel-BERT-MSA shows robust cross-task consistency, securing second place in both evaluations. This insight underscores that the effectiveness of a model in WSD depends heavily on the specific task. It also emphasizes the value of aligning model architecture with task requirements, offering useful insights for applying WSD models in real-world settings.

1 Introduction

The Word Sense Disambiguation (WSD) task aims to identify the intended meaning of a word based on its context. This task is crucial because many words in natural language are polysemous—having multiple related meanings (e.g., “paper” as a material vs. a publication)—or homonymous—having multiple unrelated meanings (e.g., “bark” as the sound a dog makes vs. the outer layer of a tree). Solving the polysemy and homonymy problems is important for language understanding tasks such as information retrieval, question answering, sentiment analysis, text summarization, and machine translation (Rahman and Borah, 2021; Pu et al., 2018), where accurately interpreting a word’s in-

tended meaning is crucial for overall system performance.

The WSD task helps resolve both polysemous and homonymous ambiguities by using context to select the correct meaning. Distributional word embeddings like Word2Vec (Church, 2017) and GloVe (Pennington et al., 2014) assign a single vector representation to each word, thereby failing to capture this semantic variation. To address this limitation, contextual word embeddings, such as those derived from transformer-based models like BERT (Devlin et al., 2019), were introduced. These embeddings dynamically generate different representation vectors for the same word depending on its context, enabling more effective handling of semantic variation (Devlin et al., 2019; Abdul-Mageed et al., 2020).

Although the WSD task has been extensively studied in English (Huang et al., 2019; Navigli, 2009), it remains particularly challenging for morphologically rich languages like Arabic. In Arabic, polysemy is highly prevalent: approximately 43% of words with diacritics have multiple meanings, and this figure rises sharply to 72% when diacritics are removed (Alshammari et al., 2024). Beyond its inherent difficulty, Arabic WSD is also significantly underexplored. This is illustrated by the recent KSAA-CAD shared task (Alshammari et al., 2024), where only one submission was made to the WSD subtask—and it was disregarded due to its performance being 64% below the baseline. Another shared task ArabicNLU (Khalilia et al., 2024) provide dataset of just 100 sentences and received just three submissions, without outperforming the baseline.

Existing approaches to the WSD problem adopt context-gloss pair classification (Huang et al., 2019) and relevance ranking (Yap et al., 2020). In this work, we frame the Arabic WSD problem as both a binary classification task and a multi-choice ((MC)) selection task. We run a

set of experiments using Small Language Models (SLMs), e.g., mBERT (Pires et al., 2019), and Large Language Models (LLMs), e.g., Mistral (Albert Q. Jiang, 2023), on the WSD dataset introduced in the KSAA-CAD shared task. The dataset is described in Section 4.1.

Our study is guided by the following research questions and corresponding contributions:

1. **RQ1. Do multilingual models outperform Arabic monolingual models, and how does task format impact cross-lingual transfer?** We compare multilingual models (e.g., mBERT, Gemma2, and others) against Arabic-specific monolingual models (e.g., AraBERT) under identical fine-tuning protocols. Performance is evaluated on both binary and multi-choice task formulations to isolate cross-lingual transfer effects.
2. **RQ2. How does task formulation affect model performance on WSD?** We conduct paired evaluations where each model executes both: (a) *binary classification*: context-gloss alignment; (b) *Multi-choice selection*: identifying the correct sense from candidate glosses. Task superiority is determined based on the F1-score metric on the dev set.
3. **RQ3. How do LLMs compare to SLMs on Arabic WSD under fine-tuning and zero-shot settings?** We benchmark open-source LLMs against SLMs under two conditions:
 - *Fine-tuned*: Models trained on our Arabic WSD training set.
 - *Zero-shot*: LLMs evaluated without task-specific training.
4. **RQ4. Can open-source LLMs match proprietary models in zero-shot Arabic WSD with identical prompts?** Top-performing open-source LLMs from RQ3 are compared against proprietary LLMs (e.g., GPT-4, Claude 3) using identical prompts and zero-shot settings. Closed-source models are accessed via API with standard hyperparameters.

Through our experiments, we found that fine-tuned multilingual models consistently outperform Arabic monolingual models. Additionally, task formulation significantly impacts performance, with some models showing clear preferences for binary classification or multi-choice formats. Finally, in

the zero-shot setting, proprietary models (e.g., GPT-4, Claude 3) outperform open-source models, highlighting the performance gap between closed and open models in Arabic WSD.

The rest of this paper is organized as follows. Section 2 overviews some works that discuss the WSD problem. Then, we talk about the methods that we use to solve the problem in Section 3. After that, the data, models, hyperparameters, and metrics used in the experiments are described in Section 4. The results are reported in Section 5 and discussed in Section 6. Finally, the paper concludes in Section 7.

2 Related Work

The WSD task has received significant attention from researchers working on English and other Indo-European languages (Huang et al., 2019). However, handling this task in morphologically rich languages like Arabic is more challenging due to the ambiguity and rich morphology present in these languages.

Huang et al. (Huang et al., 2019) fine-tuned BERT-based models to give a *yes* or *no* label for each context-gloss pair. Next, the gloss from the pair that has the highest score for *yes* is selected as the target gloss. Another work (Yap et al., 2020) formulated the WSD task as a relevance ranking task where each context has a list of 4-6 glosses. They fine-tuned BERT to give scores to each gloss, then selected the gloss with the highest scores as the target sense of the target word. In this work, we use both binary classification and multi-choice tasks to tackle the WSD problem.

Researchers use different ways to emphasize the target word in the context. For example, Huang et al. (Huang et al., 2019) surrounded the target word by quotation marks and also added the target word followed by a colon at the start of each gloss in the training instances. On the other hand, some researchers proposed to use special tokens to mark the start and end of the target word. These tokens could be the same as in (Yap et al., 2020), which used “[TGT]”, or different like in (Botha et al., 2020; Lei et al., 2017). In this paper, we use different tokens to mark the start and end of the target word.

The authors in (El-Razzaz et al., 2021) fine-tuned AraBERTv2 (Antoun et al., 2020) and AR-BERT (Abdul-Mageed et al., 2020) in binary classification on a small WSD dataset and claimed an

F1-score of 89%. However, the author of (Al-Hajj and Jarrar, 2021), investigated the results and found that the reason behind this high score, is the great intersection between the training and testing sets. To account for that, they introduced a WSD dataset and used it to fine-tune three Arabic BERT-based models, AraBERTv02, CAMeLBERT (Inoue et al., 2021), and QARiB (Abdelali et al., 2021), to tackle the problem as a binary classification one. The AraBERTv02 model performed the best and produced an accuracy of 84% on their dataset. A recent study (Wael et al., 2024) showed that using transformer-based models with siamese networks is an effective choice for the Arabic WSD task.

3 Methodology

This study focuses on Arabic WSD by tackling this problem as a binary classification task and also as a multi-choice task. We use small language models like mBERT (Pires et al., 2019) and Camel-BERT-MSA (Inoue et al., 2021) and also large language models like Mistral (Albert Q. Jiang, 2023), Llama3.1 (Grattafiori et al., 2024), and Gemma2 (Team et al., 2024). We test these models in both zero-shot and fine-tuned settings.

3.1 Binary Classification

This is the traditional way to handle the WSD task. Here every sample has a label, which is 1 to indicate that the given meaning is meant by the target word in the context and 0 otherwise. The top part of Figure 1 presents an example of a positive sample. During training, we inform the model about the target word using

```
``<token>TARGET_WORD</token>''
```

For LLMs, we use the following prompt:

```
< Given a context, a target word, and
a candidate sense (all provided
in Arabic), determine if the candidate
sense correctly
matches the meaning of the target
word in the given context.
The target word appears between <token>
and </token> tags in the context.
You MUST respond with exactly one number:
- Output "1" if the candidate sense
matches
the meaning of the word in the context.
- Otherwise, output "0".>
```

3.2 Multi-choice task

Here we have the context as the question and the different senses as possible choices of the target word. We instruct the model to focus on the target word using the same method used for the binary task and provide the possible meanings as choices for the model to select from. One possible problem here is how to specify the number of choices. We found that the best way is to use the highest number of possible choices for all the samples. Investigating the data, we found that in the worst case, a sample in the used dataset could have 43 choices, so we use that for all the samples in the training and development sets. Of course, a lot of samples do not have that number of choices, so we use "null" to fill in the missing choices as shown in the lower part of Figure 1. For LLMs, we use the following prompt:

```
<Your task is to choose **only one**
correct gloss that best matches the sense
of the word in the given context.
Please respond with **just the number**
of the correct answer (e.g., `0`, `1`,
`2`, ...). Do not include any text,
explanation, or justification.>
```

The high number of choices incurs a higher computational cost. However, we have tried to limit the number of choices using some similarity measure and found that doing so has a negative impact on the results. Because the system would then have two steps: one to select the most probable choices, say 4 or 5, and the other to make the final decision and choose the best choice. In other words, we would have two error sources instead of one.

4 Experimental Apparatus

4.1 Dataset

We use the data from the KSAA-CAD shared task (Alshammari et al., 2024). The dataset consists of two core components: the WSD context gloss mapping and dictionary data. The WSD context gloss mapping is divided into training, development, and testing sets containing 22,404, 2,801, and 2,801 samples, respectively. Each sample consists of a word, context, context ID, lemma ID, and corresponding gloss ID as depicted in part (a) of Figure 2. The dictionary contains 15,865 entries and is derived from the Contemporary Arabic Language Dictionary (Omar, 2008). Each dictionary entry contains lemma_id, gloss, and gloss_id as

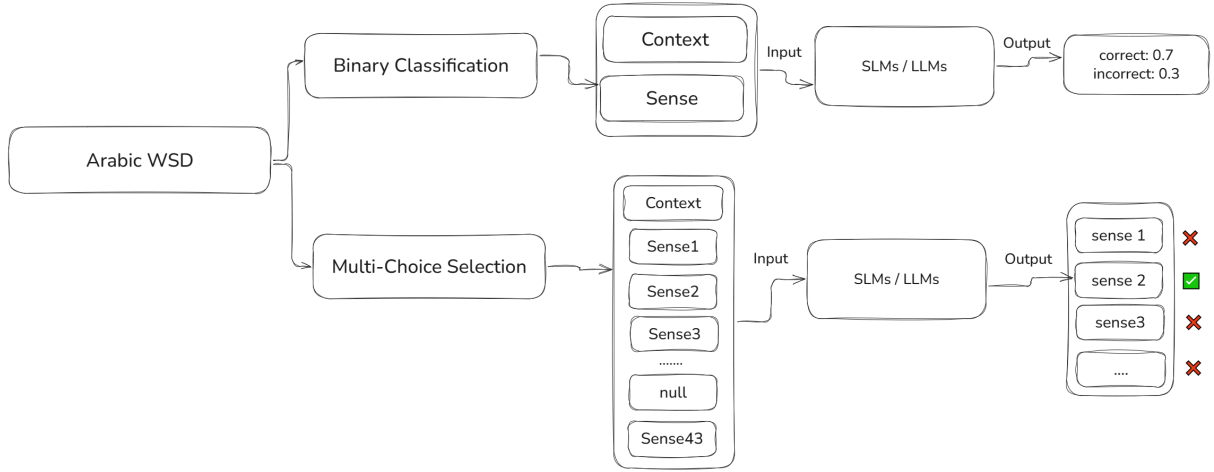


Figure 1: General workflow of the multi-choice and binary classification tasks for Arabic WSD.

shown in part (b) of Figure 2. The lemma_id is the link between the dictionary entry and the context gloss mapping. Unfortunately, the test set is not available, so we use the development set to compare the performance of the different models.

4.2 Models and Baselines

Baselines. We include two baselines from (Alshammari et al., 2024), which utilize transformer-based models in a binary classification setting, specifically CAMELBERT-MSA and AraBERTv2.

To address our research questions, we experiment with a range of models representing diverse architectures. As an Arabic monolingual encoder-only models, we use AraBERT, CAMELBERT-MSA, and ARBERT, bidirectional transformers pre-trained on MSA text and effective for Arabic contextual understanding. We also include multilingual encoder-only models such as mBERT (Pires et al., 2019), which is pre-trained on 104 languages, enabling cross-lingual transfer and multilingual understanding. This model serves as a comparative model to evaluate the effectiveness of multilingual versus monolingual representations in Arabic WSD. For decoder-only models, we employ instructed models including Llama3.1 (Grattafiori et al., 2024), Gemma2 (Team et al., 2024) with two variants in terms of size, Mistral (Albert Q. Jiang, 2023) version 0.3, and Qwen2.5 (Team, 2024). Additionally, we leverage advanced LLMs, including Claude 3.5 Sonnet (Anthropic, 2024)(version 2024-10-22), GPT-4o (Achiam et al., 2023) (version 2024-08-06), Gemini 1.5 Pro (Team et al., 2023), and DeepSeek V3 (Liu et al., 2024), which excel in text-related tasks due to extensive pre-training

on natural language.

4.3 Hyperparameter

For multi-choice small models’ fine-tuning, we trained for 3 epochs using a learning rate of 5×10^{-5} , a weight decay of 0.01, and a batch size of 2. In the other task, we set the batch size to 64, and used a learning rate of 3×10^{-5} , and models were trained for 3 epochs. All experiments were conducted using a fixed random seed of 42 to ensure reproducibility.

For instruction fine-tuning of Gemma-2B using QLoRA, we used a learning rate of 2×10^{-4} , a batch size of 4, weight decay of 0.001, and trained for 1 epoch with a maximum sequence length of 40. QLoRA parameters were set to $r = 4$, $\alpha = 16$, and dropout = 0.1. All training was performed with 8-bit quantization.

4.4 Metrics

We evaluate performance using accuracy, Macro-F1, and average precision/recall. Due to extreme class imbalance—particularly in the binary classification task (context-gloss alignment), where false labels substantially dominate true labels—we prioritize Macro-F1 across both tasks. Accuracy becomes particularly problematic here, as it fails to reflect model effectiveness. LLMs’ responses that have no output or do not match any class label are regarded as misclassifications.

5 Results

5.1 Binary Configuration

Table 1 shows the evaluation results for the binary classification approach. The highest perfor-

```
{
  "context_id": "context.301",
  "context": "يأتي برمجان اللغة العربية...",
  "word": "اللغة",
  "gloss_id": "gloss.205",
  "lemma_id": "ar.301",
}
```

(a)

```
{
  "lemma_id": "ar.301",
  "gloss_id": "gloss.205",
  "gloss": "... كل وسيلة لتبادل المشاعر والأفكار كالإشارات"
```

(b)

Figure 2: A training example of the KSAA-CAD dataset. Part (a) shows an entry from the context gloss mapping component. Part (b) displays a dictionary entry. This figure is reproduced from (Alshammari et al., 2024).

mance was achieved by fine-tuning mBERT model on Arabic data augmented with its English translation (En+Ar) reaching 83.74% Macro-F1 and the Arabic-only version (Ar) reaching 83.41%. Both significantly exceeded baseline models, including our experiment with the CAMELBERT-MSA (80.04% Macro-F1). Our experiment with CAMELBERT-MSA model also exceeded prior reported baselines achieving 94.17% accuracy.

Among zero-shot models, Qwen2.5 with 14B consistently achieved the best results, with Macro-F1 scores of 75.53% (fp16) and 75.57% (q8). It notably outperformed other zero-shot models, such as Gemma2 with 27B and Llama3.1. In contrast, Mistral-v0.3 stay far behind (13.63% fp16, 11.26% q8).

5.2 Multi-Choice Configuration

Results of the multi-choice approach are provided in Table 2. The multi-choice evaluation showed that the fine-tuned Gemma2 with 9B parameters achieved the highest Macro-F1 score (84.26%), surpassing both CAMELBERT-MSA (81%) and AraBERT (77.08%).

Considering the models under zero-shot settings, proprietary models outperformed open-source models. GPT-4 led proprietary models with a Macro-F1 score of (55.36%), followed by Gemini-1.5-Pro (52.18%) and Claude3.5-Sonnet (40.61%). For open-source models in zero-shot settings, Gemma2 with 27B parameters achieved the best performance (53.94% Macro-F1), substantially outperforming Qwen2.5 with 14B parameters (45.14%) and Mistral-v0.3 with 7B parameters (14.45%). All fine-tuned models except ARBERT, consistently exceeded models under zero-shot settings.

6 Discussion

6.1 Key Scientific Insights

Our evaluation clearly demonstrates that fine-tuned multilingual models significantly outperform monolingual Arabic-specific models, and that task formulation and training paradigms critically influence performance outcomes in Arabic WSD task as shown in Table 1 and Table 2. In addition, fine-tuned multilingual models, notably Gemma2 with 9B parameters and mBERT (En+Ar), consistently achieved higher Macro-F1 scores compared to monolingual Arabic models such as AraBERT and ARBERT.

Insights by Research Questions:

1. RQ1. Do multilingual models outperform Arabic monolingual models, and how does task format impact cross-lingual transfer?

Figure 3 confirms the results of Tables 1 and 2 that multilingual models significantly outperform Arabic-specific models under identical fine-tuning conditions. Gemma2 with 9B parameters and mBERT (En+Ar) consistently delivered higher Macro F1-scores compared to Arabic-specific models such as AraBERT and CAMELBERT-MSA, indicating strong cross-lingual transfer capability. Multilingual models leverage semantic patterns from diverse languages (e.g., English and Arabic), enriching Arabic representations beyond monolingual training.

2. RQ2. How does task formulation affect model performance on WSD?

The evaluations revealed that task formulation significantly impacts performance. Multi-choice selection task yielded substantially higher Macro-F1 scores across models compared to binary classification (context-gloss alignment). Gemma2:9B notably excelled in

Model	Accuracy	Macro F1	Precision	Recall	Provenance
<i>Baselines</i>					
CAMeLBERT	94.17	80.04	83.47	78.90	our experiment
CAMeLBERT	91.54	—	—	—	[x]
AraBERT	91.32	—	—	—	[x]
<i>Zero-shot LLMs (fp16)</i>					
Gemma-2:9B	75.58	68.77	67.59	75.91	
Gemma-2:27B	56.90	42.34	47.43	36.56	
Mistral-v0.3:7B	22.25	13.63	38.19	34.41	
Llama-3.1-8B	57.45	50.31	53.29	55.27	
Qwen2.5:7B	84.29	69.70	75.29	67.09	
Qwen2.5:14B	84.50	75.53	74.76	76.42	
<i>Zero-shot LLMs (q8)</i>					
Gemma-2:9B	73.14	67.66	67.59	77.54	
Gemma-2:27B	79.61	72.46	70.49	78.17	
Mistral-v0.3:7B	23.93	11.26	29.37	26.39	
Llama-3.1-8B	58.25	52.63	56.21	60.09	
Qwen2.5:7B	84.17	69.43	75.03	66.84	
Qwen2.5:14B	84.44	75.57	74.69	76.61	
<i>Fine-tuned Models</i>					
mBERT (En+Ar, B=64)	94.14	83.74	86.60	81.73	
mBERT (Ar, B=64)	94.02	83.41	86.85	80.90	

Table 1: Performance of language models on binary classification across different configurations

the multi-choice setup, achieving a Macro-F1 score of 84.26%, highlighting multi-choice as a superior task formulation for capturing contextual nuances in Arabic WSD. The multi-choice format provides a clear distinction among candidate senses, and modeling sense competition by ensuring rare senses contribute equally. Crucially, it supports words with diverse sense inventories (2-10 glosses), as shown in Table 4.

3. RQ3. How do LLMs compare to SLMs on Arabic WSD under fine-tuned and zero-shot settings?

Fine-tuned models consistently outperformed zero-shot counterparts, demonstrating the substantial benefits of task-specific training. The performance gap was especially pronounced, with fine-tuned Gemma2 surpassing the top-performing models in zero-shot settings, GPT-4, by 28.9 Macro-F1 points. This emphasizes that despite the reasoning capabilities of models under zero-shot settings, fine-tuning remains critical for high-quality Arabic WSD.

4. RQ4. Can open-source LLMs match proprietary models in zero-shot Arabic WSD with identical prompts?

Proprietary LLMs like GPT-4 outperformed the leading open-source model in zero-shot settings (Gemma2:27B), although the margin was relatively modest. GPT-4 (55.36% Macro-F1) surpassed Gemma2:27B (53.94%), suggesting that while open-source models are competitive, proprietary LLMs still maintain a slight performance advantage under identical zero-shot conditions and prompts.

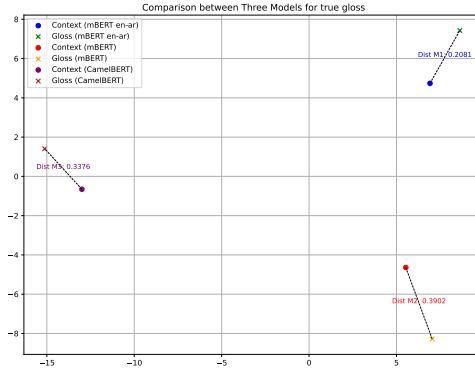
In summary, multilingual fine-tuned models substantially outperform monolingual counterparts and zero-shot models, with task formulation strongly influencing WSD performance outcomes. While proprietary models under zero-shot settings slightly outperform open-source models, fine-tuning significantly elevates overall performance.

6.2 Error Analysis

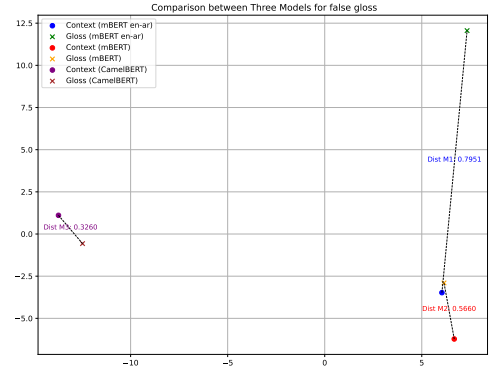
Analysis of the errors of the best performing model (Gemma2:9B) for multi-choice task in Table 3.

Model	Accuracy	Macro F1	Precision	Recall
<i>Zero-shot Closed Models</i>				
GPT-4	77.22	55.36	63.94	51.05
Claude3.5-Sonnet	77.12	40.61	44.80	37.94
Gemini-1.5-Pro	73.72	52.18	57.84	50.33
DeepSeek-V3	73.04	44.50	56.17	39.42
<i>Zero-shot Open Models</i>				
Gemma-2:9B	66.72	32.64	34.36	31.84
Gemma-2:27B	73.00	53.94	63.45	49.88
Mistral-v0.3:7B	44.55	14.45	18.07	13.07
Llama-3.1-8B	39.16	18.03	25.40	15.39
Qwen2.5:14B	68.26	45.14	50.86	44.29
Qwen2.5:7B	63.12	31.95	39.04	28.75
<i>Fine-tuned Models</i>				
Gemma-2:9B	90.14	84.26	87.84	82.84
CAMELBERT	87.00	81.00	86.84	78.63
AraBERT	84.64	77.08	80.78	75.56
mBERT	67.08	56.59	62.24	59.17
ARBERT	50.55	21.90	23.35	56.59

Table 2: Performance results for MC WSD on dev set across different models.



(a) True gloss comparison.



(b) False gloss comparison.

Figure 3: Cosine distance comparison of context and gloss embeddings across mBERT (En-Ar), mBERT, and CamelBERT for both true and false gloss predictions.

We found the following errors categories: Semantic confusion between subtly distinct glosses, surface-level biases favoring lexical cues over contextual meaning, domain mismatches misinterpreting, literal-figurative misinterpretations, and gloss issues from ambiguous distractors.

These repeated errors reveal how complex Arabic WSD can be, especially when dealing with words that demand nuanced contextual and cultural interpretation.

7 Conclusion

This work provides an evaluation of different methods and models for Arabic WSD. Multilingual models like mBERT and Gemma2 consistently outperform Arabic-specific models by 3–7% Macro-F1, confirming strong cross-lingual transfer. The multi-choice task formulation proves more effective than binary classification. Fine-tuned models delivers strong results overall, while proprietary LLMs maintained a 1.4–15% edge in zero-shot settings over open-source counterparts using identical prompts.

Error Category	Description	Example
Semantic Confusion	Model fails to distinguish between subtly different glosses due to close meaning or vague wording.	“تصيبهم” → gold: “afflicted and harmed” (نال منه وجرحه), predicted: “befell him” (نزلت به وحلت)
Surface-Level Bias	Model relies on surface cues like shared words or short/abstract glosses instead of deep semantic reasoning.	“أباطيل”/Falsehoods → distractor gloss included “خرافات”/myths from the context
Domain/Context Mismatch	Model fails to adapt to domain-specific usage (e.g., medical, religious, economic).	“الهياكل”/Structures in anatomy → predicted gloss was “huge from everything”
Literal vs. Figurative Misreading	Model interprets a figurative or abstract concept in a literal or common sense.	“الاستقبال”/Welcoming of Ramadan → predicted “greet”, gold was “begin anew”
Gloss Design Quality Issue	The gloss set itself contains poor quality distractors , vague definitions, or domain-inappropriate candidates.	“الأعمال”/business in economy → predicted “intentional act”, gold was “profession”

Table 3: General Error Categories in Arabic WSD Mutli-choice Model Predictions where the number of glosses is 7 and 9.

#Gloss	mBERT	Gemma2:9B	Camel-Bert	AraBERT	ARBERT	#Samples
2	77.18	91.63	88.58	86.49	45.13	526
3	63.87	90.97	84.51	83.51	45.88	388
4	62.77	90.05	87.96	84.66	45.09	333
5	52.98	85.83	82.85	80.70	33.99	240
6	47.70	85.92	78.93	77.37	28.00	187
7	59.60	88.49	83.66	81.77	39.78	174
8	53.69	82.41	75.40	65.75	24.95	96
9	59.47	90.58	84.64	78.71	20.91	105
10	45.49	73.91	83.80	69.83	25.42	51

Table 4: Fine-tuned models performance on MC configuration by gloss count in terms of macro F1 score.

In the future work, we will address semantic granularity through contrastive gloss pairing and integrate Arabic knowledge resources to mitigate cultural-contextual gaps as highlighted by our error analysis. In addition, exploring more sophisticated prompting strategies such as few-shot and Chain-of-Thought (CoT) could better leverage LLMs’ full capabilities.

Limitations

This study has three main limitations. First, the evaluation was limited to Arabic WSD using a single dataset, which may constrain generalizability to other languages or domains. Second, zero-shot evaluations utilized only one prompt engineering approach which may not fully exploit the potential of LLMs. Third, fine-tuning experiments were

conducted exclusively on open-source models (e.g., Gemma, mBERT). The high computational and financial costs associated with fine-tuning closed-source proprietary models hinder direct performance comparisons with such models.

References

- Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *arXiv preprint arXiv:2102.10684*.
- Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: Deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Moustafa Al-Hajj and Mustafa Jarrar. 2021. [ArabGloss-BERT: Fine-tuning BERT on context-gloss pairs for WSD](#). In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 35–43, Held Online. INCOMA Ltd.
- Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Albert Q. Jiang, Alexandre Sablayrolles. 2023. Mistral 7b. <https://arxiv.org/abs/2310.06825>.
- Waad Alshammari, Amal Almazrui, Asma Al Wazrah, Rawan Almatham, Muneera Alhoshan, and Abdulrahman Alosaimy. 2024. [KSAA-CAD shared task: Contemporary Arabic dictionary for reverse dictionary and word sense disambiguation](#). In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 677–685, Bangkok, Thailand. Association for Computational Linguistics.
- AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.
- Jan A. Botha, Zifei Shan, and Daniel Gillick. 2020. [Entity Linking in 100 Languages](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, Online. Association for Computational Linguistics.
- Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.
- 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.
- Mohammed El-Razzaz, Mohamed Waleed Fakhr, and Fahima A Maghraby. 2021. Arabic gloss wsd using bert. *Applied Sciences*, 11(6):2567.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- 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.
- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in arabic pre-trained language models. *arXiv preprint arXiv:2103.06678*.
- Mohammed Khalilia, Sanad Malaysha, Reem Suwaileh, Mustafa Jarrar, Alaa Aljabari, Tamer Elsayed, and Imed Zitouni. 2024. Arabicnlu 2024: The first arabic natural language understanding shared task. *arXiv preprint arXiv:2407.20663*.
- Wenqiang Lei, Xuancong Wang, Meichun Liu, Ilija Ilievski, Xiangnan He, and Min-Yen Kan. 2017. Swim: A simple word interaction model for implicit discourse relation recognition. In *IJCAI*, pages 4026–4032.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.
- Ahmed Mukhtar Omar. 2008. Dictionary of contemporary arabic language.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

- Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. [Integrating weakly supervised word sense disambiguation into neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 6:635–649.
- Nazreena Rahman and Bhogeswar Borah. 2021. [A method to disambiguate a word by using restricted boltzmann machine](#). In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 388–398, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLP AI).
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.
- Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2412.15115*.
- Tasneem Wael, Eman Elrefai, Mohamed Makram, Sahar Selim, and Ghada Khoriba. 2024. Pirates at arabicnlu2024: Enhancing arabic word sense disambiguation using transformer-based approaches. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 372–376.
- Boon Peng Yap, Andrew Koh, and Eng Siong Chng. 2020. [Adapting BERT for word sense disambiguation with gloss selection objective and example sentences](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 41–46, Online. Association for Computational Linguistics.