ZYC at SemEval-2026 Task 5: Application of BERT-based Contextual Embeddings Similarity for WSD

Sunny Zhou
sunny.t.zhou@gmail.com

Jordan Youner
jordan.youner@gmail.com

Dean Cahill
dpcahill98@gmail.com

Abstract

We investigate contextual embedding manipulation for Word Sense Disambiguation (WSD) as part of SemEval-2026 Task 5. We propose four approaches built on BERT-like pretrained models, experimenting with the informativeness of similarity calculations and classification methods. We introduce scratch-trained cross-attention mechanisms inspired by GLiNER to compute similarity between definition or synonym representations and the full context. Our best performance achieved 57% accuracy with a Spearman correlation of 0.20. Our results suggest that finetuning strategy and training curriculum matter more than pretrained model choice for this novel task, and we identify several directions for future improvement. View our code base at: https://github.com/heliosraz/SemEval52026

1 Introduction

Polysemy is an extremely common phenomenon in natural language, and creates a great deal of complexity for accurately modeling language. In this paper, as a part of SemEval 2026 Task 5 (Gehring et al., 2026), we attempt to tackle this topic. We investigate the efficacy of contextual embedding manipulation strategies on this task by using four approaches, based on BERT and SBERT-like models. (1) Our naive approach is a scales plausibility score between contextual embeddings of the target word in both an ambiguous context and an unambiguous example sentence, based on cosine similarity. (2) We train a classification head to predict the score from the similarity between the ambiguous context and definition using SBERT.

Our remaining two approaches introduce cross-attention as a mechanism for calculating similarity. In both, we adopt Zaratiana et al. 2024's method of training a similarity head from embeddings generated from sequence pairs. (3) in Definition Cross-Attention (DXA), this is done with the ambiguous word's definition, and the full instance context. (4) in Synonym Entity Recognition (SER), we represent the senses of a target word indirectly as categories with WordNet (Miller, 1994) synonyms and compare them to the context.

Our final submission reflects the performance of our four approaches tested with a variety of pretrained models. Each approach builds on the last under the assumption of capturing contextual knowledge. Our approaches achieve a max accuracy of 57% with a Spearman's correlation of 0.20 on the test set. The results suggest limitations within our method, and we propose a variety of improvements for the future.

2 Related Work

2.1 Embedding Similarity for WSD

WSD is commonly tackled as a constituent problem to be addressed in other, more comprehensive tasks, such as Neural Machine Translation. Sense similarity has been a common approach for tackling WSD for decades (Lesk, 1986). The rise of embedding spaces as a way to reason about relative meaning has provided for fruitful exploration into this notion of similarity, and others have produced various forms of embeddings designed to capture this semantic information. Suzuki et al. 2018 developed concept embeddings for all-words WSD, fully disambiguating entire documents by p redicting similarity between target words and their surrounding contexts across different senses.

Gloss knowledge has also been shown to influence model performance on WSD (Luo et al., 2018). When it comes to using models like BERT for word sense disambiguation, we must choose how we leverage both contextual and gloss information (i.e correct word sense). Given a context and a set of possible meanings, one might expect that the proper meaning has a higher similarity. In other words, we can treat WSD as a classification

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2960–2966

task on context-sense pairs based on embedding similarity. Huang et al. 2019 utilize BERT in such a manner. They build a set of context-possible gloss pairs with the Semeval 2007 WSD dataset and the senses present in WordNet, labeling pairs with yes or no, and they achieve high performance on contemporary benchmarks. Our approaches are informed by this notion of leveraging the similarity between these embeddings. In particular, one of our approaches involves Reimers and Gurevych 2019's Sentence Transformer (SBERT) model to experiment with pooling the sense embeddings as another way to capture this gloss knowledge.

2.2 Model Training Practices

It is well known that BERT and variations utilize certain pretraining objectives, such as **Masked Language Modeling (MLM)**, **Next Sentence Prediction (NSP)**, and Permuted Lanugage Modeling (PLM). These methods are shown to enable BERT to generalize over task specific training later (Liu et al., 2019; He et al., 2021; Song et al., 2020; Devlin et al., 2019).

Howard and Ruder 2018 develop a system for Universal Language Model Fine-Tuning (ULMFiT), which enables generalized transfer learning across various text classification tasks. A language model is pretrained on general-domain data, finetuned on task-specific data with the same pretraining task, and then a general classifier block is added. The finetuned classification uses a gradual unfreezing of each layer, starting from the final one and unfreezing the preceding layer with each epoch. This approach prevents catastrophic forgetting, which is common in small dataset finetuning.

Our approach in DXA and SER is inspired by this staged training. In both strategies, we separate the training of the attention head and the classifier block, and we utilize gradual unfreezing when training the classifier. DXA, in particular, implements the MLM pretraining from BERT and classifier finetuning from ULMFiT.

3 Task Description

This task asks us to produce a plausibility score [1, 5] for a semantic sense that matches human annotators' judgments. A precontext and an ambiguous sentence (and sometimes an ending) is given for each instance. It is presented as a discrete choice to the annotators, and thus we choose to approach this problem as a classification task wherein the predicted class is a given score; our goal is to match the prediction distribution to the annotators' distribution.

4 Data and Evaluation Metric

The task organizers provided AmbiStory, a presplit dataset containing 2279 raw instance in train, 587 in dev, and 929 in test (Gehring et al., 2026). To provide the word in the ambiguous sentence enough information to work with, we combined the pre-context, ambiguous sentence, and endings to form the complete story, which we call full context. Additionally, an average and a standard deviation of the human judgments were given for each instance. We use this to calculate the annotator's unnormalized normal distribution across a 5-point plausibility score.

We further split the train data by crossing "judged_meaning" with the set "full_context", "ending", and "example_sentence" and the set "full_context", "ending" with itself. Each of these pairs were labeled as "target" and "source" respectively. This yielded a more general dataset of 12159 instances used for pretraining in DXA. The full context and ending self crossed pairs and the judged meaning and example sentence pairs were given an average of 5 and a standard deviation of 0. Since the models produce 0-indexed classifications, all of the averages are decreased by 1.

The task calculates correctness based on if a prediction is within a standard deviation of the average. To match this continuous metric, we calculate the average from the classification distribution to use as our prediction.

5 Methodology

5.1 Naive Similarity

We input the full context into BERT, which updates the embeddings with the ambiguous contextual information. The given example sentence is assumed to be an unambiguous use of the target homonym, capturing the same sense to the judged meaning, so cosine similarity measures the difference in sense usage. Similar to how functions (e.g. softmax) offer probability proxies, we use this calculation to proxy model's probability confidence that these two embeddings match meanings. Since we are comparing two very similar senses, we can assume that the cosine similarity is bounded by $[0, 1]$, instead of $[-1, 1]$. Scaling this confidence with 5 and

rounding up to the nearest integer, we map this to a plausibility score of $[0, 5]$.

5.2 Cross Context Embeddings (CCE)

The previous approach makes a critical assumption that sense captured by the example sentence proxies the judged meaning's sense. However, an ambiguous word may still retain ambiguity even in the clearest of contexts. Thus, we posit that a weighted pooling of the embeddings in judged meaning delivers a more accurate representation of the target meaning.

Additionally, while the previous approach uses embedding similarity to map directly to human plausibility judgment, we ask here if there exists a projection from embedding similarity to a plausibility score. Thus, this approach consists of four modules: sentence transformer, context transformer, similarity, and classifier.

5.2.1 Sentence Transformer Module

We calculate and pool the contextual embeddings of the judged meaning with a pretrained SBERT-like model in this module. For an embedding dimension of E, the resultant vector is shape $1 \times E$.

5.2.2 Context Transformer Module

We gather contextual embeddings for the full context, making this module rather similar to the naive similarity approach. However, unlike the naive similarity approach, the model used here needs to match the SBERT-like model used in the previous module, but without pooling, ensuring the embedding sizes are the same. The resultant matrix shape is $S \times E$, where S is the sequence length and E is the embedding dimension.

5.2.3 Similarity Module

We batch multiply the context transformer module result and the sentence transformer module result to calculate the similarity, preserving angle and magnitude, as opposed to only angle in cosine similarity (Vaswani et al., 2017). The resultant vector is shape $S \times 1$.

5.2.4 Classifier Module

The $S \times 1$ vector is fed into a general classifier with one hidden layer size 128 and a ReLU activation and dropout layer after the hidden layer. The classifier outputs a distribution over five classes, representing the probability of each plausibility score.

5.3 Definition Cross-Attention (DXA)

The previous approach relies on SBERT pooling to capture the meanings in judged meaning, limiting the contentfulness of the signal. Zaratiana et al. 2024 addresses this disconnect between the pretrained model and trainable signals by mapping contextual embeddings from a BERT-like model to a space where similarity is meaningful. Instead of searching for entity spans, a word sense disambiguation task can be viewed as searching for important parts of a definition within a context, utilizing the definition as a expanded proxy of the target sense.

We implement this using cross-attention, where keys are from the context, and queries and values are from the words within the definitions. This approach consists of three modules: encoder, cross-attention, and classifier module. The cross-attention module replaces the similarity module from CCE, and the classifier module is identical to CCE's.

5.3.1 Encoder Module

A BERT-like module is used to gather the contextual embeddings of an input with the following construction, borrowed from Zaratiana et al. 2024:

`[JUDGED_MEANTING]+[SEP]+[FULL_CONTEXT]`

Judged meaning and full context are both padded separately to a length of 255 and 256 respectively, yielding a sequence length of 512 regardless of judged meaning or full context length.

5.3.2 Cross-Attention Module

Following the approach in Zaratiana et al. 2024, the encoded input is split at the $[SEP]$ token, resulting in two sequences: x_{target} and x_{source}. Cross-attention is then calculated, following PyTorch's `scaled_dot_product_attention()` implementation:

$$q = W_q @ x_{target}$$
$$k = W_k @ x_{source}$$
$$v = W_v @ x_{target}$$
$$H = softmax(\frac{q @ k^T}{\sqrt{d}}) @ v$$

where d is the size of k, and H the output of the attention. The logits are compiled by concatenating the max and mean across the keys (Howard and Ruder, 2018):

$$h = [max(H), mean(H)]$$

The resultant vector is shape $1 \times 2 * L_{source}$

5.4 Synonym Entity Recognition (SER)

Zaratiana et al. 2024's addition of the trainable [ENT] tag to facilitate the NER task in a BERT-like system functions as a sort of latent carrier for the entity type, allowing for a sigmoid to be calculated between the relevant spans and for entities to be properly captured & scored. We explore, in SER, the potential of quasi-clustering to measure similarity of context across multiple word senses. The intuition of this is that the plausibility of a given sense would correlate with the similarity between a context and a set of senses as a whole. Plausibility score will then, in theory, correlate with the variance of similarity scores across the sense represented by the synonyms.

The general structure of this model's architecture is largely similar to DXA. The following elucidates changes between DXA and SER.

5.4.1 Encoder Module

The encoder module primarily differs through the inclusion of a synonym generation step. Pulled from WordNet, the set of senses are concatenated into a vector consisting of alternating words and $[SYN]$ tags. An additional trainable token $[MISC]$ is added to the end of this sequence to capture other senses not covered by the selected synonyms, resulting the from:

```
[[SYN], w_1, ... [SYN], w_n, [MISC]]
```

These values were tokenized and concatenated to the relevant data from the training instance and then padded to length of 512. The final form of the contextual embeddings of an input are as follows:

```
[SYNONYMS]+[SEP]+[FULL_CONTEXT]+[JUDGED_MEANING]
```

5.4.2 Cross-Attention Module

We select only $[SYN]$ and $[MISC]$ embeddings in the encoded target sequence. The judged meaning portion of the source sequence is mean pooled and joined at the end of the encoded full context. Thus the two sequences are represented as such:

$$x_{target} \rightarrow [[SYN], [SYN], ...[MISC]]$$
$$x_{source} \rightarrow [FULL\_CONTEXT] + mean([JUDGED\_MEANING])$$

The cross-attention mechanism in this module is identical to its counterpart in DXA. The synonym entity index with the max similarity with the judged meaning is used to select the attention weight row used in classification:

$$h = [H[argmax(H[:, -1]), : L_{FULL\_CONTEXT}]]$$

where H is the result from attention and $L_{FULL\_CONTEXT}$ is the length of the full context. This setup facilitates the computation of a cross-attention between the positions representing each synonym and the full context of the ambiguous sentence.

6 Experimental Setup

6.1 General Setup

The attention and classifier weights were initialized with Xavier. We trained all of the models for 100 epochs (or until convergence) with AdamW ($\beta_1 = 0.7$, $\beta_2 = 0.99$) and used KL Divergence for training loss to match the predicted class distribution to the provided distribution. We used a learning rate of $1e^{-5}$ for the encoder layers and $3e^{-5}$ for others. A weight decay of 0.1 on the encoder layers and 0.01 on others was also used. These hyperparameters were chosen to reflect Howard and Ruder 2018 and (Zaratiana et al., 2024)'s paradigm. Evaluation uses the same hyperparameters, expect for dropout.

6.2 DXA

The training was split into 2 parts, as opposed to the 3 parts in Howard and Ruder 2018: (1) pretraining of the encoder and attention mechanism, and (2) finetuning of the classifier. Preliminary testing indicated that straight classifier finetuning without pretraining the attention mechanism prevented the capture of any generalizable information, resulting in overfitting

Similar to BERT, masked language modeling (MLM) was used to pretrain. For each instance, a random token was chosen in the definition was masked at index i. Then a possible token is predicted from the ith row of cross-attention module, representing the values of the masked token. 0.4 dropout was applied to the attention mechanism and 0.3 dropout was applied to the classifier.

In the classifier finetuning phase, we used gradual thawing to ensure training started from the least generalized components first (Howard and Ruder, 2018).

6.3 SER

This approach utilized a similar methodology as DXA, but without pretraining the cross-attention. We figured the problem with the attention weights here cannot be addressed by a generalizing task, like MLM, but this is not within the scope of this

paper. We use the same hyperparameters as in the DXA gradual thawing here.

7 Results

While Spearman correlation is not significant, our models overall outperform the naive similarity approach by up to 6% in terms of accuracy, as seen in Table 1. The models are ranked from CCE, DXA, to SER, from best to worst. The CCE models generally seem to perform the best, regardless of pretrained model choice. Out of those models, CCE-MPNet was the highest performer with 57% accuracy and a Spearman's correlation of 0.20.

Model	Accuracy	Spearman	Average
Baseline			
GPT-4o*			0.756
Llama-3.1 8B*			0.563
Naive Similarity	0.51	0.21	0.36
CCE			
DistilRoBERTa	0.52	-0.01	0.26
MiniLM	0.57	0.04	0.31
MPNet	**0.57**	**0.20**	**0.385**
RoBERTa	0.57	0.10	0.335
DXA			
BERT	0.53	0.07	0.30
DeBERTa	0.53	-0.02	0.26
DistilBERT	0.52	0.04	0.28
SER			
BERT	0.50	-0.33	0.09
DeBERTa	0.52	0.03	0.28
DistilBERT	0.52	0.15	0.34

Table 1: Task Performance Results.
* Baselines provided by the task organizers.

8 Discussion

8.1 Similarity Module Attention Weights

By tagging the tokens by part of speech (POS) and looking at the scores produced by each model's similarity module,[1] we can take a closer look at how the models process token relationships. Additionally, by grouping up instances into positive, neutral, and negative samples to remove inverse relationships between the groups, we expect to have low dispersion.

The models' similarity modules highlight some shortcomings of the models. As expected, we do see lower scores towards negative samples and higher scores towards positive samples, but, surprisingly, this does not necessarily translate to better performance. For example, in Figures 1 and 2, we see that DXA-BERT had similarly low scores in both positive and negative samples. In comparison, DXA-DistilBERT has a significantly more contentful attention weights for positive samples, such as its proper noun (NNP) and noun (NN) rows. Despite this, DXA-BERT performs better overall. This indicates that the attention mechanism is not capturing the desired query-key relations. We suspect high scores in negative samples indicate knowledge, such as a lack of similarity, that will help with determining a non-example.

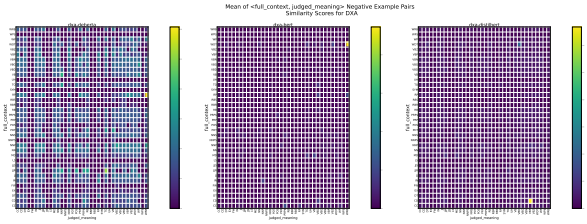

Figure 1: Mean of [full_context, judged_meaning] **Negative** Sample Pairs Similarity Scores for DXA

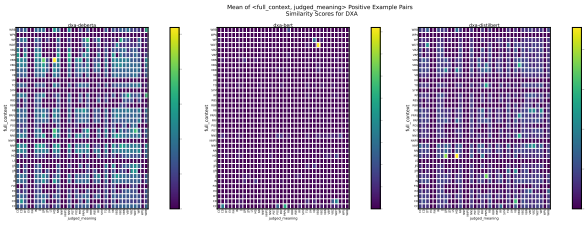

Figure 2: Mean of [full_context, judged_meaning] **Positive** Sample Pairs Similarity Scores for DXA

On the other hand, the dispersion, measured by the coefficient of variance (CV), illustrates the models' consistency for each sampling group. We found that an inverse relationship between CV and a model's Spearman. In Figure 3, we see that CCE-DistilRoBERTa has by far the most CV and has a negative Spearman. This pattern is observed in nearly all of the sampling groups for all the approaches, suggesting a method of reducing variableness of this mechanic would also increase Spearman across all approaches.

Surprisingly, there was not a strong pattern between the feature pairs we compared. POS, such as cardinal numbers and wh-words, were attended to regardless of model performance. This suggests that methods of reducing this noise could improve the classification signal. The better overall performance of the CCE models, despite only having

[1] Other attention weights can be found on the GitHub repo.

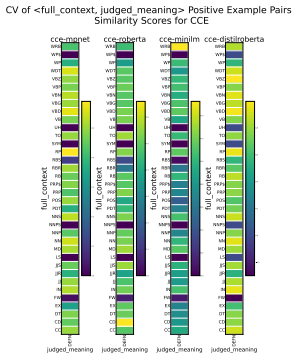

Figure 3: CV of [full_context, judged_meaning] Positive Example Pairs Similarity Scores for CCE

1D similarity calculation, indicates that pooling with value and the choice of classification signal in DXA and SER dilutes the informativeness of the attention weights, over inflating the noise.

8.2 Training Curriculum

Notably, the models perform similarly. The consistency within architecture suggests that the differences among pretrained models do not significantly affect the results. This indicates that model performance on this novel task is less dependent on the pretraining regimen and more on the finetuning. This is surprising. Previous work suggests promising knowledge transfer capabilities for many tasks, implying a resistance to forgetting.

Moreover, the lacking performance from DXA and SER compared to CCE highlights the difference of the models' initial naivete. In other words, DXA and SER required training of an attention layer from scratch. As mentioned earlier, gross overfitting (e.g. train: 0.98 and val: 0.4 accuracy) in initial training experiments on DXA indicated the training paradigm sensitivity these models have. On the other hand, we reason the more informative similarity calculation in DXA and SER causes the lower prediction variance between base model. This suggests that with a better training curriculum DXA and SER could out perform CCE.

The clearest next steps are a further exploration of the training curriculum and a more robust approach to constructing the "meaningful" embedding space. Although MLM has been fruitful for previous BERT-like models, the success of GLiNER suggests explicit training of the keys, queries, and values could be useful instead of relying on a classification head to translate a meaningful signal. A better training curriculum could also consolidate DXA and SER as SER requires the

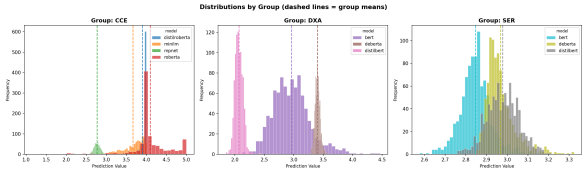

Figure 4: Distribution of model predictions on test set

understanding of judged meaning in addition to the synonym senses. Implementing pretraining in SER, in order to capture other forms of information relevant to this specific model's goals (e.g. positional information for the carrier embeddings) would allow us to more closely associate the synonyms with the target homonym and directly compare the way that these terms compose with the context.

In DXA and SER, the target tokens attended over every source token equally. A training paradigm that introduces informed sparsity can reduce the confusion that stop words may introduce.

8.3 Data Informativeness

While our results tend to surpass the naive similarity approach, the Spearman's correlation shows a weak monotonic relationship. One possible explanation is in how the accuracy is measured. As shown in Figure 4, even though within architectures the performance stays consistent, the distributions are vastly different. It indicates that the acceptable range for the cases can be fairly large. One might expect that a stricter measure of accuracy would tie closer to the Spearman's correlation.

Additionally, while WordNet is a quite useful resource for dynamic synonym generation, we cannot guarantee that a given homonym's sense is contained within the word's synsets. We tried to mitigate this by appending [MISC] as a catch-all for any senses not contained in the synsets. This also allows for calculations on homonyms which have *no* synsets, but this final tag is limited in its contentfulness, and ultimately may be superfluous in its current construction.

Inconsistent feature factors, such as variable judged meaning length and WordNet synsets, provides difficulties for padding. Despite accounting for padding in the attention masking, it is unclear how it affects the classifier. Moreover, in SER, this results limits us to a concrete number of senses across all homonyms, limiting our ability to account for variable ambiguity in words.

References

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.

Janosch Gehring, Selina Meyer, and Michael Roth. 2026. SemEval-2026 task 5: Rating plausibility of word senses in ambiguous stories through narrative understanding. In *Proceedings of the 20th International Workshop on Semantic Evaluation*, San Diego, California. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. 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.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, page 24–26, New York, NY, USA. Association for Computing Machinery.

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

Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Ryu Suzuki, Kanako Komiya, Masayuki Asahara, Minoru Sasaki, and Hiroyuki Shinnou. 2018. All words word sense disambiguation using concept embeddings. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, Miyazaki, Japan. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.