# An Expectation-Realization Model for Metaphor Detection

**Oseremen O. Uduehi**
School of EECS
Ohio University
Athens, OH 45701
ou380517@ohio.edu

**Razvan C. Bunescu**
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC 28223
razvan.bunescu@uncc.edu

## Abstract

We propose a new model for metaphor detection in which an expectation component estimates representations of expected word meanings in a given context, whereas a realization component computes representations of target word meanings in context. We also introduce a systematic evaluation methodology that estimates generalization performance in three settings: within distribution, a new strong out of distribution setting, and a novel out-of-pretraining setting. Across all settings, the expectation-realization model obtains results that are competitive with or better than previous metaphor detection models.

## 1 Introduction and Motivation

Metaphors enhance the communicative aspects of language by connecting concepts from new domains, often abstract, with more familiar ones, usually concrete (Lakoff and Johnson, 1980). Metaphorical expressions have many uses, from helping frame an issue in order to emphasize some aspects of reality (Boeynaems et al., 2017), to creating a strong emotional effect (Blanchette and Dunbar, 2001; Citron and Goldberg, 2014). The ubiquity of metaphors means their computational treatment (Veale et al., 2016) has received significant attention in the NLP community, as surveyed by Shutova (2015) and more recently Tong et al. (2021). Owing to its important communicative function, metaphorical expression detection has been approached over the years using a wide variety of NLP techniques, ranging from models employing hand-engineered features (Shutova et al., 2010; Bulat et al., 2017), to RNNs (Gao et al., 2018; Mao et al., 2019), to more recently pre-trained language models (Choi et al., 2021; Ghosh et al., 2022; Li et al., 2023), to mention just a few.

Recent state of the art models for metaphor detection rely on the Metaphor Identification Procedure (MIP) (Group, 2007), according to which metaphors happen whenever the contextual meaning of a word is different from its basic, literal meaning. Implementations of MIP vary mainly in how they estimate representations of the basic meaning of a word: MelBert (Choi et al., 2021) uses simply the BERT embedding of the word without any context, whereas BasicBERT and BasicMIP (Li et al., 2023) use an average of all literal uses of the word as marked in the training data.

In this paper we propose a new theory of metaphor identification, the *Expectation-Realization* model, that is motivated by the observation that the metaphorical use of a word, i.e. its *realization* in context, leads to surprise due to a violation of a literal word *expectation* engendered by the same context. Surprise offers a general mechanism through which stories and music trigger emotion (Meyer, 1961), and correlates with creative uses of language, such as humor and metaphor (Bunescu and Uduehi, 2022). Correspondingly, we propose an architecture that is structured around two modules: one module aims to estimate the literal meaning expectation through the use of a context where the target word is masked, whereas the other module aims to estimate the realized meaning of the target word as used in context. The new model is competitive with previous SoA in terms of within distribution (WiD) generalization. We further propose two new evaluation scenarios: a *strong out-of-distribution* (OoD) setting that ensures target lexemes do not appear during training, and a novel *out-of-pretraining* (OoP) setting that aims to ensure that the metaphorical phrase was not seen during pretraining. The large gap between OoP and WiD results elucidates why pretrained LMs struggle with metaphor identification.

## 2 The Expectation-Realization Model

The architecture of the Expectation-Realization (ER) model for metaphor detection is shown in

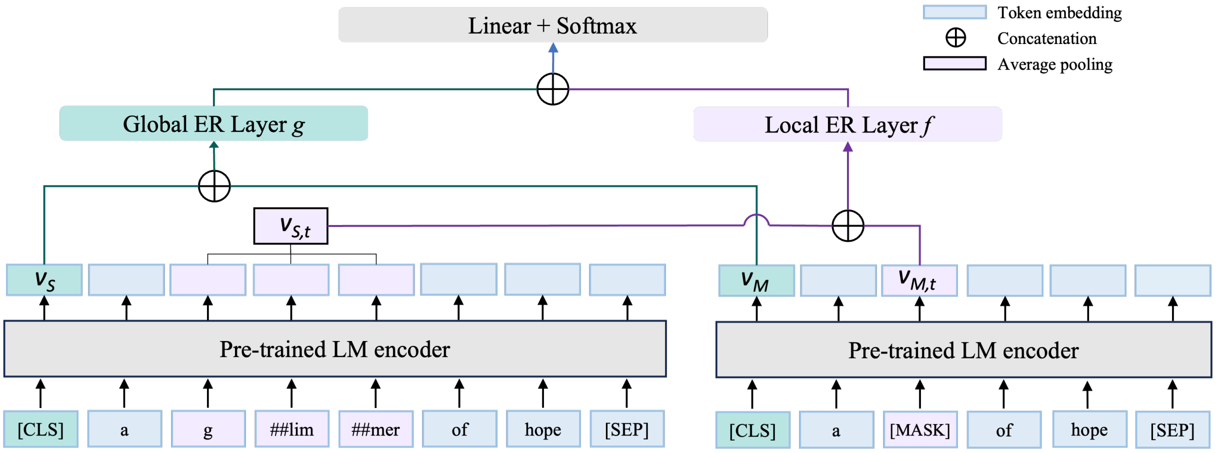

Figure 1: ER architecture: left branch for Realization embeddings, right branch for Expectation embeddings. The high expectation for the literal meaning "a [bit] of hope" is confounded by the word "glimmer", causing surprise.

Figure 1. To compute the realized (R) meaning $v_{S,t}$ of the target word in context, a copy of the Transformer encoder of a pre-trained language model (shown on the left) processes the input text $S$ where the target word at position $t$ is marked with a special token. To compute the expectation (E) of the literal meaning $v_{M,t}$ induced by the context, the same pre-trained language model (shown on the right) process the same input text $M$ where the target word is masked. Additionally, global expectation $v_M$ and realization $v_S$ representations are also computed at the sentence level using the embeddings for the special [CLS] token. The concatenation of the local target word ER embeddings and the sentence-level ER embeddings are passed through non-linear layers $f$ and $g$, respectively, to capture interactions between expectation and realization embeddings at word-level as $h_{local} = f[v_{M,t}; v_{S,t}]$, and at sentence level as $h_{global} = g[v_M; v_S]$. To enable a fair comparison with previous models, we instantiate the pre-trained Transformer encoder using RoBERTa base (Liu et al., 2019). The concatenated local and global ER representations are then used as input features to a logistic regression model that estimates the probability $\hat{y}$ that the target word is used metaphorically.

$$\hat{y} = \sigma(w^\intercal[h_{local}; h_{global}] + b)$$

The ER model parameters together with the pre-trained LM parameters are trained and fine-tuned, respectively, in order to minimize a loss function $L^i = L^i_{CE} - L^i_{Sim}$ that contains a *cross-entropy loss* $L^i_{CE}$ and a *similarity loss* $L^i_{Sim}$ computed as:

$$\begin{aligned} L^i_{CE} &= y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \\ L^i_{Sim} &= \alpha_1 \cos(u_{M,t}, v_{M,t}) + \alpha_2 \cos(u_M, v_M) \end{aligned}$$

where $y_i$ and $\hat{y}_i$ are the ground truth and predicted labels, respectively, for training sample $i$. The embeddings $u$ are obtained from the original pre-trained LM with fixed parameters, whereas the embeddigns $v$ are obtained from the fine-tuned LM. Importantly, the similarity loss encourages the fined-tuned LM to learn expectation embeddings $v$ that do not deviate much from the original embeddings produced by the pre-trained LM. The hyper-parameters $\alpha_1$ and $\alpha_2$ trade-off the global and local components of the similarity term within the overall loss. Given that most words in the vocabulary are used with their literal meaning most of the time, the similarity loss has the effect of anchoring the fine-tuned LM such that its expectation embeddings $v$ reflect a literal meaning of words.

## 3 Experimental Evaluation

We run evaluations on three English metaphor datasets: the VUA-18 Amsterdam Metaphor Corpus (Chen et al., 2020), TroFi (Birke and Sarkar, 2006) and LCC (Mohler et al., 2016). Table 1 summarizes the statistics of the datasets used in our evaluations. The VUA-18 dataset is split into training, validation and test datasets denoted by VUA-18$_{tr}$, VUA-18$_{dev}$ and VUA-18$_{te}$ respectively. The examples in the VUA-18 dataset are sentences where selected words of the sentence are annotated as metaphorical or not. The LCC Metaphor dataset is a large, multilingual dataset of metaphor annotations created by a team of researchers at the Language Computer Corporation (LCC). Each target word is annotated with a metaphoricity rating on a four-point scale [0, 3]. In our experiments we use a subset of the English dataset where examples with

| Dataset | #words | %M | #Sent | Len |
|---|---|---|---|---|
| VUA-18$_{tr}$ | 116,622 | 11.2 | 6,323 | 18.4 |
| VUA-18$_{dev}$ | 38,628 | 11.6 | 1,550 | 24.9 |
| VUA-18$_{te}$ | 50,175 | 12.4 | 2,694 | 18.6 |
| LCC | 5,646 | 28.9 | 5,390 | 28.9 |
| TroFi | 3,737 | 43.5 | 3,737 | 28.3 |

Table 1: Detailed statistics of datasets. #words is the number of target words to be classified, %M is the percentage of metaphorical words, #Sent is the number of sentences, and Len is the average sentence length.

metaphoricity score of 3 are considered as positive and examples with metaphoricity score of 0 as negatives. The TroFi dataset consists of a collection of literal and nonliteral usage of 50 verbs which occur in 3,737 sentences selected from the WSJ corpus.

For the evaluations on VUA-18 dataset, we use the same hyperparameter settings from (Choi et al., 2021) for training all models. For the LCC and TroFi experiments, the development dataset was used for determining the best hyperparameter settings. We use the same hyperparameter settings for all the models. The batch size and max sequence length were set at 32 and 150, respectively. We train for 12 epochs without dropout, and linearly increase the learning rate from 0 to 5e-5 in the first two epochs, after which we decreased it linearly to 0 during the remaining 10 epochs. The tuned similarity weights $\alpha_1$ and $\alpha_2$ were 1.0. for the within-distribution experiments and 0.0 for out-of-distribution experiments. Results are averaged over 5 runs with different random seeds. The detailed ranges used for hyperparameters tuning are presented in Appendix A.

Given that VUA-18 is the only dataset on which all 3 metaphor-detection baselines were previously evaluated, we use it to compare their performance against ER. As shown in Table 2, the ER model outperforms both MDGI-Joint-S (Wan et al., 2021) and MelBERT (Choi et al., 2021), and is competitive with the more complex BasicBERT (Li et al., 2023) that requires annotation of literal tokens.

### 3.1 Three Generalization Scenarios

The generalization performance of each of the 3 models is evaluated in three settings: *within distribution (WiD)*, strong *out of distribution (OoD)*, and *out-of-pretraining (OoP)* metaphor generalization. For the WiD generalization, we randomly split the dataset into 10 folds and run 10-fold evaluation, where 9 folds are used for training and development, and 1 fold is used for testing, with the procedure repeated 10 times so that each folds gets to be used as a test fold. For strong OoD generalization, the 10 folds are created such that the lemmas of target words are disjoint across the folds. For the OoP generalization setting, we identify a subset of 237 positive examples within the LCC dataset that are novel or unconventional metaphors. The criteria for creating this subset were example with the highest metaphoricity score of 3.0 that were also rare according to a Google search, i.e. returning fewer that 25 search results. To complete the novel version of the dataset, negative examples are randomly sampled from the LCC dataset such that the ratio of positive to negatives for this novel dataset is similar to that of the original LCC dataset. Note that the OoP examples, which are *novel to the pretrained LM*, are different from the crowdsourced novel metaphors from (Do Dinh et al., 2018), which are *novel to the average human* annotator. For the OoP evaluation we only compute the test performance on the OoP subset of examples using the models already trained on data from the within-distribution setting, ensuring that no OoP test example has been used during training.

Due to the imbalanced distribution of positive and negative examples in the datasets, we report only precision, recall and F1-score metrics. For 10-fold evaluation we report their micro-averages.

### 3.2 Generalization Results

Tables 3 and 4 show the results of comparison of the ER Model against MelBERT and R-SPV on the LCC and TroFi datasets. The R-SPV model implements only the realization component of the ER model, using as input the sentence with the target word marked, as shown on the left of Figure 1. Note that even though this is equivalent with the SPV component of the MelBERT model, it is found to perform as well as MelBERT. Additionally, for the LCC and TroFi datasets in the WiD setting we also report the performance of a logistic regression model that trained on binary responses from GPT-4 on 13 questions that are aimed at identifying metaphors and also distinguishing metaphors from other types of figurative language (Appendix B).

For the the within distribution (WiD) setting of VUA-18, LCC and TroFi, the ER model statistically significantly outperforms R-SPV and MelBERT, as determined through a one-tailed, paired

| Dataset | Model | Prec | Rec | F1 |
|---|---|---|---|---|
| VUA-18 (WiD) | MDGI-Joint-S | 81.3 | 73.2 | 77.0 |
| | MelBERT | 80.1 | 76.9 | 78.5 |
| | BasicBERT | 79.5 | 78.5 | 79.0 |
| | ER | 80.2 | 77.5 | 78.8 |

Table 2: Performance comparison of ER model with baselines on the VUA-18 dataset.

| Dataset | Model | Prec | Rec | F1 |
|---|---|---|---|---|
| LCC (WiD) | R-SPV | 86.2 | 83.9 | 85.0 |
| | MelBERT | 86.1 | 83.8 | 84.9 |
| | GPT-4 | 82.1 | 77.5 | 79.7 |
| | ER | **86.9** | **84.3** | **85.5**$^{*\dagger}$ |
| | ER-Ens | 87.7 | 85.3 | 86.5 |
| LCC (OoD) | R-SPV | 83.6 | 79.8 | 81.6 |
| | MelBERT | 83.4 | 79.8 | 81.5 |
| | ER | **84.0** | **80.6** | **82.2**$^{*\dagger}$ |
| | ER-Ens | 85.9 | 81.9 | 83.9 |
| LCC (OoP) | R-SPV | 88.0 | 94.3 | 91.1 |
| | MelBERT | 87.6 | 94.5 | 90.9 |
| | ER | **88.8** | **95.1** | **91.8**$^{*\dagger}$ |
| | ER-Ens | 89.3 | 95.7 | 92.4 |

Table 3: Performance comparison of ER model with baselines on LCC dataset. * and †indicate significantly better F1 than R-SPV and MelBERT, respectively.

t-test of significance at $p < 0.05$ level. The VUA-18 results are notably lower than the LLC results for all methods. Error analysis revealed that almost any non-literal use of a word is annotated as a positive example in VUA, including idioms. Therefore, the patterns are more complicated. Idioms, in particular, lack any clear pattern, hence they require memorization, which may explain the much lower VUA performance. The logistic regression model on top of features from GPT-4 had the lowest WiD F1 on LCC and TroFI, indicating that, despite its language understanding capabilities, it still struggles to accurately identify metaphors, a result that can also be understood in light of insights drawn from the OoP scenario below. The GPT-4 results were obtained using binary answers to questions in a zero-shot setting; it is expected that in-context learning with few-shot examples or fine-tuning of GPT models, while more computationally demanding than using BERT-like models, will lead to better results. We leave such experiments for future work.

| Dataset | Model | Prec | Rec | F1 |
|---|---|---|---|---|
| TroFi (WiD) | R-SPV | 70.2 | 71.8 | 71.0 |
| | MelBERT | 69.5 | 73.3 | 71.3 |
| | GPT-4 | 63.5 | 60.9 | 62.1 |
| | ER | **70.2** | **73.7** | **71.9**$^{*\dagger}$ |
| | ER-Ens | 72.2 | 73.5 | 72.8 |
| TroFi (OoD) | R-SPV | 57.4 | 69.6 | 62.8 |
| | MelBERT | 57.1 | 69.8 | 62.7 |
| | ER | 57.0 | 70.5 | 63.0 |
| | ER-Ens | 58.1 | 71.8 | 64.2 |

Table 4: Performance comparison of ER model with baselines on TroFi dataset. * and †indicate significantly better F1 than R-SPV and MelBERT, respectively.

For the strong out-of-distribution (OoD) evaluation on the LCC and TroFi datasets, the ER model on average performs better than both R-SPV and MelBERT, with the comparison on LCC being statistically significant. The results from the OoD settings show a significant drop compared to the within distribution setup with the result being less worse for LCC than TroFi because of the more diverse nature of the target words in the LCC dataset. This drop in performance in the OoD scenario suggests that the models rely on some form of memorization, which is detrimental to identifying metaphors that use unseen words. The nature of the TroFi dataset makes the OoD generalization even worse, as the dataset contains only 50 words and thus the model has limited diversity in terms of target metaphorical words.

In the out-of-pretraining (OoP) evaluation setting conducted for the LCC dataset, the ER model again outperforms both baselines, obtaining a 9.8% relative error reduction over MelBERT. Note that the OoP results are much higher than the WiD results for all methods, which seems to indicate that the difficulty of metaphor detection comes from the large number of conventional metaphors that appear often in the pretraining data; that in turn makes it hard for pretrained models such as BERT or GPT to create embeddings that can discriminate conventional metaphors from literal language.

Lastly, ensembles ER-Ens of 5 ER models further improve metaphor detection in all settings.

## 4 Conclusion and Future Work

We introduced a new model for metaphor detection rooted in the hypothesis that non-literal uses of words trigger surprise, or violation of expectations given by the context. We further proposed two new evaluation scenarios: strong out-of-distribution and out-of-pretraining. Extensive experiments show that the simple ER model is competitive with, and often outperforms, state-of-the-art models.

In this work, expectations of literal meaning were computed based on context words. In future work, we plan to also compute expectations of literal meanings of words by leveraging large amounts of text where words are known to be used literally, such as descriptions of physical, concrete concepts in Wikipedia. Furthermore, we plan to generalize the ER approach from word-level metaphors to phrase-level constructions, such as idioms, which too violate expectations of literal language use.

## Acknowledgements

We would like to thank the anonymous reviewers for their suggestions and constructive feedback. This research was partly supported by the United States Air Force (USAF) under Contract No. FA8750-21-C-0075. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the USAF.

## References

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336, Trento, Italy. Association for Computational Linguistics.

Isabelle Blanchette and Kevin Dunbar. 2001. Analogy use in naturalistic settings: The influence of audience, emotion, and goals. *Memory & Cognition*, 29(5):730–735.

Amber Boeynaems, Christian Burgers, Elly Konijn, and Gerard Steen. 2017. The impact of conventional and novel metaphors in news on issue viewpoint. *International Journal of Communication*, 11(0).

Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. Modelling metaphor with attribute-based semantics. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 523–528, Valencia, Spain. Association for Computational Linguistics.

Razvan C. Bunescu and Oseremen O. Uduehi. 2022. Distribution-based measures of surprise for creative language: Experiments with humor and metaphor. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 68–78, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Xianyang Chen, Chee Wee (Ben) Leong, Michael Flor, and Beata Beigman Klebanov. 2020. Go figure! multi-task transformer-based architecture for metaphor detection using idioms: ETS team in 2020 metaphor shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 235–243, Online. Association for Computational Linguistics.

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. MelBERT: Metaphor Detection via Contextualized Late Interaction using Metaphorical Identification Theories. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1763–1773, Online. Association for Computational Linguistics.

Francesca M. M. Citron and Adele E. Goldberg. 2014. Metaphorical sentences are more emotionally engaging than their literal counterparts. *Journal of Cognitive Neuroscience*, 26(11):2585–2595.

Erik-Lân Do Dinh, Hannah Wieland, and Iryna Gurevych. 2018. Weeding out Conventionalized Metaphors: A Corpus of Novel Metaphor Annotations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1424, Brussels, Belgium. Association for Computational Linguistics.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 607–613, Brussels, Belgium. Association for Computational Linguistics.

Debanjan Ghosh, Beata Beigman Klebanov, Smaranda Muresan, Anna Feldman, Soujanya Poria, and Tuhin Chakrabarty, editors. 2022. *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid).

Pragglejaz Group. 2007. Mip: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.

George Lakoff and Mark Johnson. 1980. *Metaphors we Live by*. University of Chicago Press, Chicago.

Yucheng Li, Shun Wang, Chenghua Lin, and Frank Guerin. 2023. Metaphor Detection via Explicit Basic Meanings Modelling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 91–100, Toronto, Canada. Association for Computational Linguistics.

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.

Rui Mao, Chenghua Lin, and Frank Guerin. 2019. End-to-end sequential metaphor identification inspired by linguistic theories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3888–3898, Florence, Italy. Association for Computational Linguistics.

Leonard Meyer. 1961. *Emotion and Meaning in Music*. University of Chicago.

Michael Mohler, Mary Brunson, Bryan Rink, and Marc Tomlinson. 2016. Introducing the LCC metaphor datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4221–4227, Portorož, Slovenia. European Language Resources Association (ELRA).

Ekaterina Shutova. 2015. Design and Evaluation of Metaphor Processing Systems. *Computational Linguistics*, 41(4):579–623. _eprint: https://direct.mit.edu/coli/article-pdf/41/4/579/1807226/coli_a_00233.pdf.

Ekaterina Shutova, Lin Sun, and Anna Korhonen. 2010. Metaphor identification using verb and noun clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1002–1010, Beijing, China. Coling 2010 Organizing Committee.

Xiaoyu Tong, Ekaterina Shutova, and Martha Lewis. 2021. Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4673–4686, Online. Association for Computational Linguistics.

Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov. 2016. Metaphor: A Computational Perspective. *Synthesis Lectures on Human Language Technologies*, 9(1):1–160. Publisher: Morgan & Claypool Publishers.

Hai Wan, Jinxia Lin, Jianfeng Du, Dawei Shen, and Manrong Zhang. 2021. Enhancing Metaphor Detection by Gloss-based Interpretations. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1971–1981, Online. Association for Computational Linguistics.

## A Hyperparameter Tuning

Details for the hyperparameter tuning for the models and dataset are presented in Table 5.

| Hyperparameter | Tuning values |
|---|---|
| learning rate | [1e-5, 2e-5, 3e-5, 4e-5, 5e-5] |
| dropout ratio | [0.0, 0.1, 0.2, 0.25, 0.4, 0.5] |
| similarity weight $\alpha$ | [0, 0.5, 1, 2, 4] |
| hidden dims | [[768], [768,768], [768,768,1]] |
| hidden activation | [None, relu] |
| optimizer | [Adam] |
| train batch size | [32] |

Table 5: Hyperparameters tuning range used in experiments. For the similarity weight, $\alpha = \alpha_1 = \alpha_2$.

## B GPT-4 prompt template

The sample prompt we used to query GPT-4 is shown below:

You are a professional linguist. For the text below, answer precisely the following questions. Only print out a Python list containing your answers.

text: The sun *walked* between the clouds.

1. What word is emphasized?
2. Is the emphasized word "walked" used literally in the text? Yes or No?
3. Is the emphasized word "walked" used figuratively in the text? Yes or No?
4. Is the emphasized word "walked" used metaphorically in this text? Yes or No?
5. Is the emphasized word "walked" used with its literal meaning in the text? Yes or No?
6. Is the emphasized word "walked" used with its most common literal meaning in this text? Yes or No?
7. Is the emphasized word "walked" used with a concrete meaning in the text? Yes or No?
8. Is the emphasized word "walked" used with a physical meaning in the text? Yes or No?
9. Is the emphasized word "walked" used with its conventional meaning in the text? Yes or No?
10. Is the emphasized word "walked" used with its most common meaning in this text? Yes or No?
11. Is the emphasized word "walked" used with its original (oldest) meaning in this text? Yes or No?
12. Is the emphasized word "walked" part of a metaphorical expression in the text? Yes or No?
13. Is the emphasized word "walked" part of an idiomatic expression in the text? Yes or No?
14. Is the emphasized word "walked" part of a multiword expression in the text? Yes or No?