Metaphor Detection via Explicit Basic Meanings Modelling

Yucheng Li[1]*, Shun Wang[2]*, Chenghua Lin[2]†, Frank Guerin[1]
[1] Department of Computer Science, University of Surrey, UK
[2] Department of Computer Science, University of Sheffield, UK
{yucheng.li, f.guerin}@surrey.ac.uk
{swang209, c.lin}@sheffield.ac.uk

Abstract

One noticeable trend in metaphor detection is the embrace of linguistic theories such as the metaphor identification procedure (MIP) for model architecture design. While MIP clearly defines that the metaphoricity of a lexical unit is determined based on the contrast between its *contextual meaning* and its *basic meaning*, existing work does not strictly follow this principle, typically using the *aggregated meaning* to approximate the basic meaning of target words. In this paper, we propose a novel metaphor detection method, which models the basic meaning of the word based on literal annotation from the training set, and then compares this with the contextual meaning in a target sentence to identify metaphors. Empirical results show that our method outperforms the state-of-the-art method significantly by 1.0% in F1 score. Moreover, our performance even reaches the theoretical upper bound on the VUA18 benchmark for targets with basic annotations, which demonstrates the importance of modelling basic meanings for metaphor detection.

1 Introduction

Metaphors are widely used in daily life for effective communication and vivid description. Due to their unusual and creative usage, further processes are required for machines to understand metaphors, which results in Computational Metaphor Processing (CMP), an active research direction in NLP (Rai and Chakraverty, 2020). Recent studies demonstrate that CMP can benefit a wide range of NLP tasks including creative language generation (Chakrabarty et al., 2020; Li et al., 2022b), sentiment analysis (Li et al., 2022a), and machine translation (Mao et al., 2018). Metaphor identification, aiming to detect words used metaphorically, is the very first stage in CMP. For example, target words '*attack*' or '*defend*' in the context sentence *"He attacks/defends her point."* do not literally involve *physical engagement*, so they are supposed to be identified in metaphor detection for further process (Steen et al., 2010).

Linguists, philosophers and psychologists propose various ways to define metaphors, including substitution view (Winner, 1997), comparison view (Gentner, 1983), class inclusion view (Davidson, 1978), and conceptual metaphor theory (Lakoff and Johnson, 2008). In contrast to these theories which are relatively complex in nature, Pragglejaz (2007) propose a simple and effective linguistic theory called Metaphor Identification Process (MIP) which can identify metaphors in unrestricted textual corpora. MIP gains increasing popularity as it detects metaphorical terms regardless of specific conceptual mapping or comparison among source and target domain, which makes the identification operational and straightforward.

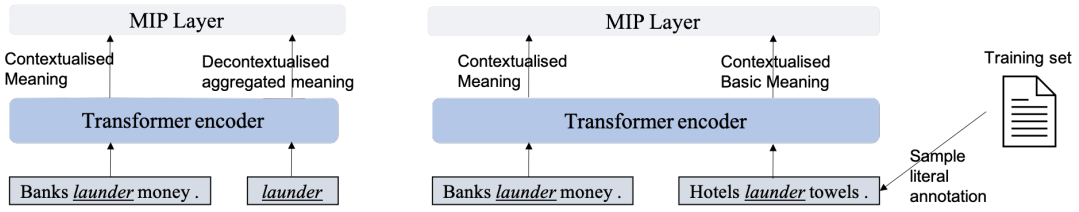
According to MIP, a word is tagged as a metaphor if its contextual meaning contrast with its "*more basic meaning*". The basic meaning here is defined as "*more concrete; related to bodily action; more precise (as opposed to vague); historically older*" guided by dictionaries[1]. For example, in the sentence "*This project is such a headache!*", the target *headache* here is metaphorical since its contextual meaning is "a thing or person that causes worry or trouble; a problem", which contrasts with the more basic meaning "a continuous pain in the head"[2].

Existing deep learning methods for metaphor identification usually depend on MIP in their model design (Mao et al., 2019; Choi et al., 2021; Song et al., 2021; Li et al., 2023; Wang et al., 2023). However, existing works usually ignore basic meaning modelling and instead use *aggregated meaning* to contrast with contextual meaning in MIP. We

* The two authors contributed equally to this work.
† Corresponding author

[1] MIP defines basic meanings based on Macmillan and Longman Dictionary
[2] ldoceonline.com/dictionary/headache

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 91–100



a. AMIP b. BasicMIP

Figure 1: Comparison of the AMIP implementation in (Mao et al., 2019; Choi et al., 2021) and our BasicMIP.

call the MIP in these implementations 'Aggregated MIP' (AMIP). For example, Mao et al. (2019) and Li et al. (2023) implement MIP by contrasting contextual meaning representation with GloVe embedding and Decontextualised[3] RoBERTa embedding, respectively. However, aggregated meaning representations, such as GloVe and decontextualised embeddings, are not the same as basic meanings in general. They usually represent a frequency-based weighted average of multiple word meanings. In cases where the basic meaning is the most frequent, then the aggregated meaning can be a reasonable approximation to basic meaning. However, it is very common that metaphorical meanings are more frequent so that using aggregated meaning violates the fundamental rule of MIP. For example '*back*' means 'the rear surface of the human body' as basic meaning, but its non-basic senses, e.g. '*going back*', '*back up*', '*back in 1960*', are more frequently used in corpora. This makes the aggregated representation of *back* diverge from its basic sense, so that metaphor cannot be identified via measuring contrast with contextual meaning.

A further pitfall of previous works is that the aggregated representations used are static rather than contextualised. For example, aggregated representation GloVe and Decontextualised RoBERTa embeddings used by Mao et al. (2019) and Li et al. (2023) are both static embedding, which are not compatible with the contextual meaning they compared to and has been shown to have worse representational quality (Bommasani et al., 2020).

In this paper, we propose a novel metaphor identification mechanism, BasicMIP, which implements MIP via direct basic meaning modelling of targets. BasicMIP explicitly leverages basic annotations from training set, where basic meaning of words are labeled as `literal` according to MIP theory. First, it samples `literal` instances for each target. Then, the basic meaning representation of target is obtained by summing up the target embeddings of sampled `literal` instances. Finally, the basic representations are contrasted with their contextual meaning representation in target sentences to identify metaphors. We also present our novel metaphor detection model, BasicBERT, which not only uses BasicMIP but also inherits the AMIP module and SPV (Selectional Preference Violation Wilks, 1975, 1978) theory from prior works.

Extensive experiments conducted on two metaphor benchmarks show that BasicBERT significantly outperforms current SOTAs. In the VUA20 benchmark, our model exceeds MelBERT by 1% in F1 score. In the VUA18 benchmark, our performance even reaches the theoretical upper bound for the targets with `literal` annotations in the training set. Our code and data can be found at `https://github.com/liyucheng09/BasicBERT`.

2 Method

BasicBERT model consists of three main components: BasicMIP, AMIP, and SPV. We include both AMIP and BasicMIP as some words do not have literal annotations in training set, so AMIP is an useful augmented component for these cases.

2.1 BasicMIP

BasicMIP, as shown in Figure 1, is based on MIP, in which a target word's contextualised meaning in the current context is compared with its more basic meaning. **First**, the contextual meaning representation is produced by feeding the current sentence to the RoBERTa network (Liu et al., 2019). Formally, given a sentence $S = (w_1, ..., w_t, ..., w_n)$, where w_t is the target word, we obtain representations as follows:

$$H = \text{RoBERTa}(\text{emb}_{\text{cls}}, ..., \text{emb}_t, ..., \text{emb}_n) \quad (1)$$

[3] which means feed the single word to pretrained language model and use the outputted vector as the representation.

Here CLS is a special token indicating the start of an input; emb_i is the input embedding for word w_i; and $H = (h_{\text{cls}}, ..., h_t, ..., h_n)$ represents the output hidden states. We denote the contextual meaning embedding of w_t as $v_{S,t} = h_t$.

Second, to contrast the contextual meaning with the basic meaning, our model learns the basic meaning representation of the target from the training annotations. According to MIP (Steen et al., 2010), we consider targets with `literal` label to represent their basic meaning. Therefore, we sample `literal` examples of the target w_t from the training set denoted as $S_b = (..., w_t, ...) \in \mathcal{U}$, where $\mathcal{U}$ is training set and S_b stands for the context sentence containing a basic usage of w_t. Our model obtains the basic meaning embedding of w_t by feeding S_b to a RoBERTa encoder similar to Equation 1 and get the t-th output hidden state h_t. The final *contextualised* basic representation of w_t is averaged among multiple literal instances, and is formulated as $v_{B,t}$, which is intrinsically different to the aggregated representation of frequent meaning used in prior works.

At last, we compute a hidden vector h_{BMIP} for-BasicMIP, by concatenating $v_{S,t}$ and $v_{B,t}$.

$$h_{\text{BMIP}} = f_0([v_{S,t}, v_{B,t}]) \tag{2}$$

where $f_0(\cdot)$ denotes a linear layer to learn semantic difference between $v_{S,t}$ and $v_{B,t}$.

2.2 AMIP and SPV

The AMIP implementation of MIP theory is inherited by our model, where contextual meaning and aggregated meaning of the target are compared. Here the contextual target meaning embedding of w_t is $v_{S,t}$, the same as in Equation 2. Then, we feed the single target word w_t to the RoBERTa network to derive the decontextualised vector representing the aggregated meanings of w_t (Choi et al., 2021): $v_{F,t} = \text{RoBERTa}(\text{emb}_t)$.

The SPV theory is also employed which measures the incongruity between the contextual meaning of the target and its context. Similarly, the contextual target meaning embedding is $v_{S,t}$, and the context sentence meaning is produced by the CLS embedding denoted as v_S, where $v_S = h_{cls}$.

Finally, we compute AMIP (h_{AMIP}) from the contextual and aggregated target embedding, and SPV (h_{SPV}) from the contextual target meaning embedding and the sentence embedding.

$$h_{\text{SPV}} = f_1([v_S, v_{S,t}]) \tag{3}$$

$$h_{\text{AMIP}} = f_2([v_{S,t}, v_{F,t}]) \tag{4}$$

where $f_1(\cdot)$ and $f_2(\cdot)$ denote a linear layer to learn the contrast between two features.

2.3 Prediction

Finally, we combine three hidden vectors h_{AMIP}, h_{SPV} and h_{BMIP} to compute a prediction score $\hat{y}$, and use binary cross entropy loss to train the overall framework for metaphor prediction.

$$\hat{y} = \sigma(W^\top [h_{\text{BMIP}}; h_{\text{AMIP}}; h_{\text{SPV}}] + b) \tag{5}$$

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \tag{6}$$

3 Experiments

Dataset. We conduct experiments on two public bench datasets: **VUA18** (Leong et al., 2018) and **VUA20** (Leong et al., 2020), which are the most popular metaphor detection benchmarks, released in the figurative language workshops of ACL in 2018 and 2020. VUA20 is an extended version of VUA18 which contains more annotations.

Baselines. **RNN_ELMo** (Gao et al., 2018) combined ELMo and BiLSTM as a backbone model. **RNN_MHCA** (Mao et al., 2019) introduced MIP and SPV into RNN_ELMo and capture the contextual feature within window size by multi-head attention. **RoBERTa_SEQ** (Leong et al., 2020) is a fine-tuned RoBERTa model in the sequence labeling setting for metaphor detection. **MelBERT** (Choi et al., 2021) realize MIP and SPV theories via a RoBERTa based model. **MrBERT** (Song et al., 2021) is the SOTA on verb metaphor detection based on BERT with verb relation encoded. **FrameBERT** (Li et al., 2023) uses frame classes from FrameNet in metaphor detection and achieves SOTA performance on both VUA18 and VUA20.

Implementation details. For target words which have no `literal` annotations in the training set, we return the decontextualised target representation as the basic meaning vector in the BasicMIP module. Therefore, the BasicMIP, in this situation, will degenerate to the AMIP implementation.

4 Results and Analysis

Overall results. Table 1 shows a comparison of the performance of our model against the baseline

Models	VUA18			VUA20		
	Prec	Rec	F1	Prec	Rec	F1
RNN_ELMo	71.6	73.6	72.6	-	-	-
RNN_MHCA	73.0	75.7	74.3	-	-	-
RoBERTa_SEQ	80.1	74.4	77.1	75.1	67.1	70.9
MrBERT	82.7	72.5	77.2	-	-	-
MelBERT	80.1	76.9	78.5	75.9	69.0	72.3
FrameBERT	82.7	75.3	78.8	79.1	67.7	73.0
BasicBERT	79.5	78.5	**79.0***	73.3	73.2	**73.3***
w/o BasicMIP	81.7	75.1	78.3	74.8	69.8	72.2

Table 1: Performance comparison on VUA datasets (best results in **bold**). NB: * denotes our model outperforms the competing model with $p < 0.05$ for a two-tailed t-test.

	Models	Annotation	#sample	#target	F1	Acc
VUA20	w/ BMIP	has literal	18060	4076	**74.7**	91.2
		no literal	4136	2539	68.2	86.9
	w/o BMIP	has literal	18060	4076	73.3	91.0
		no literal	4136	2539	68.2	87.6
VUA18	w/ BMIP	has literal	38825	3874	**81.1**	94.7
		no literal	5122	2915	67.3	87.4
	w/o BMIP	has literal	38825	3874	80.7	94.8
		no literal	5122	2915	66.5	88.0

Table 2: Breakdown results of BasicMIP. **has literal** indicates targets have `literal` annotations in the training set, and **no literal** indicates they have not.

models on VUA18 and VUA20. BasicBERT outperforms all baselines on both VUA18 and VUA20, including the SOTA model MelBERT by 0.5% and 1.0% in F1 score, respectively. A two-tailed t-test was conducted based on 10 paired results (with different random seeds) between BasicBERT and the strongest baseline MelBERT on both VUA18 ($p = 0.022$) and VUA20 ($p = 0.006$).

Ablation test. We also perform an ablation experiment to test the benefit of the basic modelling. As shown in Table 1, the performance of BasicBERT drops substantially when removing basic meaning modelling (w/o BasicMIP) by 0.7% on VUA18 and 1.1% on VUA20, respectively.

Target with and without basic annotation Some target words in the test set might not have `literal` annotations in the training set. To better understand the mechanism of basic meaning modelling, we test the performance of BasicBERT on targets *has* and *has not* basic meaning annotations in the training data. As shown in Table 2, there are 13% of samples in the VUA18 test set for which we cannot find a corresponding basic meaning annotation from training set. This number increases to 22% for VUA20. We find BasicBERT gains significant improvement on targets with `literal` annotations from VUA20 via basic meaning modelling by 1.4% in F1 score. For these targets with `literal` annotations in the VUA18 benchmark, BasicBERT gives 81.1% in F1 score, which reaches the theoretical upper bound since the Inter-annotator agreement (IAA) value of VUA18 is around 0.8 (Leong et al., 2018) (which means further improvement might lead to overfitting).

Modules	Metaphor	Literal
Contextual vs. Frequent	0.516	0.642
Contextual vs. Basic	**-0.082**	**0.809**

Table 3: Contrast of features within AMIP and BasicMIP. The experiment is conducted on VUA20.

Contrast measuring. To better compare our BasicMIP with AMIP, we conduct an experiment to directly measure the contrast between features within BasicMIP and AMIP, i.e., the contrast between the contextual and the basic meaning for BasicMIP, and the contrast between the contextual and the most frequent meaning for AMIP. Intuitively, we expect the contrast to be obvious for metaphor cases and to be slight for literal cases. Cosine distance is used to compute the contrast between two features. The contrast will fall into $(-1, 1)$, smaller numbers meaning more contrasting, larger numbers meaning less contrasting.

The results (see Table 3) show that the contrast of BasicMIP features is much more obvious for metaphorical samples, and there is less contrast for literal samples compared with AMIP. Moreover, AMIP only shows a minor gap of 0.13 contrast between metaphor and literal cases. However, a significant gap of 0.89 is captured by BasicMIP between metaphor and literal cases, which demonstrates that BasicMIP learns the difference between metaphorical and literal expressions well. In summary, the results show the effectiveness of basic meaning modelling in metaphor detection.

Case study. We perform an exploratory analysis on metaphors where BasicMIP succeeds to detect but fails without it. Prior methods might find very simple targets difficult to classify, such as *see, back, hot*. This is mainly because their metaphorical meanings are more frequent than their basic meanings, which leads the aggregated representations dominate by metaphorical semantics. For example, *see* means *look* basically. But, *I see why you are angry* and *this place has seen the war* are even more

frequent in language corpus. Therefore, the contrast with contextual meaning tends not to indicate metaphors anymore. On the contrary, basic meaning modelling learns their basic representation by focusing literal annotations directly, which enables BasicMIP to tackle them with high accuracy (see Appendix A for examples).

5 Conclusion

We proposed BasicBERT, a simple but effective approach for metaphor detection. The key feature of our method is the basic meaning modelling for metaphors from training annotations. Extensive experiments show that our model achieves best results on two benchmarks against SOTA baselines and also reaches the theoretical upper bound for instances with basic annotation. We believe our approach can be extended to other creative language with minor updates. In future, we will try apply our approach to identify other types of creative language, such as humour and sarcasm.

6 Limitations

This paper mainly focuses on modelling basic meaning to identify metaphors, typically learning basic meanings from literal annotations of the VUA dataset. However, our analysis reveals that the literal annotations of the VUA dataset are incomplete, which means that some words in VUA have no literal instances annotated. Although we propose using contextual word embeddings as a backup in this paper, another promising solution for this issue might be using external resources such as dictionaries. Leveraging dictionaries is commonly used to assist manual metaphor detection, so it could also help our BasicMIP mechanism to generalise. We leave this for future work.

References

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781.

Tuhin Chakrabarty, Smaranda Muresan, and Nanyun Peng. 2020. Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6455–6469.

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.

Donald Davidson. 1978. What metaphors mean. *Critical inquiry*, 5(1):31–47.

Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. 2018. Neural metaphor detection in context. *arXiv preprint arXiv:1808.09653*.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 vua and toefl metaphor detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 18–29.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 vua metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2022a. The secret of metaphor on expressing stronger emotion. In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 39–43, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yucheng Li, Chenghua Lin, and Frank Guerin. 2022b. Cm-gen: A neural framework for chinese metaphor generation with explicit context modelling. In *International Conference on Computational Linguistics*.

Yucheng Li, Shunyu Wang, Chenghua Lin, Frank Guerin, and Loïc Barrault. 2023. Framebert: Conceptual metaphor detection with frame embedding learning. In *Conference of the European Chapter of the 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. *arXiv preprint arXiv:1907.11692*.

Rui Mao, Chenghua Lin, and Frank Guerin. 2018. Word embedding and wordnet based metaphor identification and interpretation. In *Annual Meeting of the Association for Computational Linguistics*.

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.

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

Sunny Rai and Shampa Chakraverty. 2020. A survey on computational metaphor processing. *ACM Computing Surveys (CSUR)*, 53(2):1–37.

Wei Song, Shuhui Zhou, Ruiji Fu, Ting Liu, and Lizhen Liu. 2021. Verb metaphor detection via contextual relation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4240–4251.

Gerard Steen, Lettie Dorst, J. Herrmann, Anna Kaal, Tina Krennmayr, and Trijntje Pasma. 2010. *A method for linguistic metaphor identification: From MIP to MIPVU*.

Shunyu Wang, Yucheng Li, Chenghua Lin, Loïc Barrault, and Frank Guerin. 2023. Metaphor detection with effective context denoising. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.

Yorick Wilks. 1978. Making preferences more active. *Artificial intelligence*, 11(3):197–223.

Ellen Winner. 1997. *The point of words: Children's understanding of metaphor and irony*. Harvard University Press.

A Examples of targets *get* and *back*

Table 4 shows cases where previous methods fails but ours successes. Corresponding sentences with basic usage of target from training set are also included. We also show word senses illustration in Figure 2 and Figure 3. The figure is drawn via RoBERTa embedding and PCA techniques. We can see the most frequent meaning of *back* is *'former location'* and *'travel backward'* instead of the basic meaning *'human body'*. And the meanings of *get* are almost equally frequent.

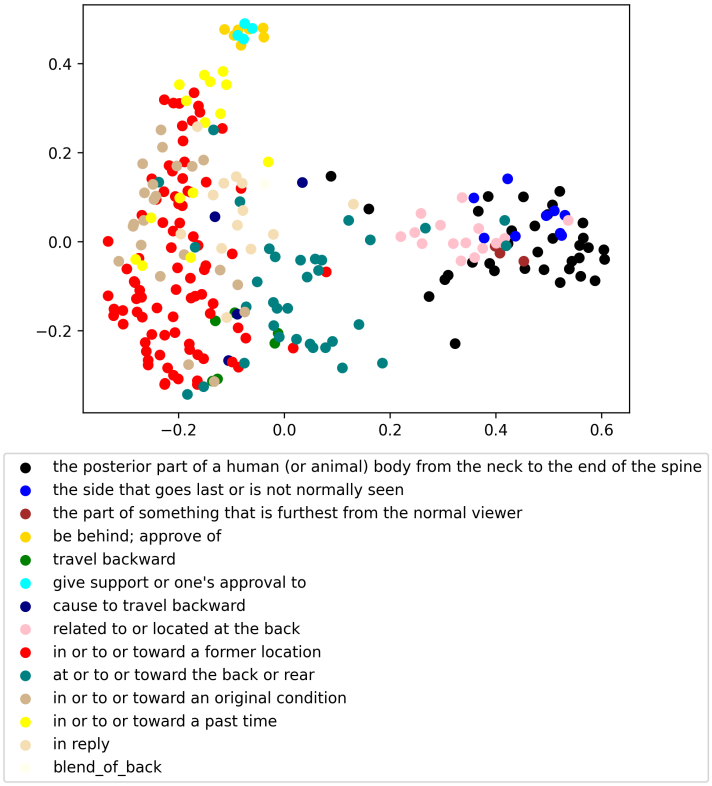


Figure 2: Senses of *back* from word sense disambiguation dataset Semcor.

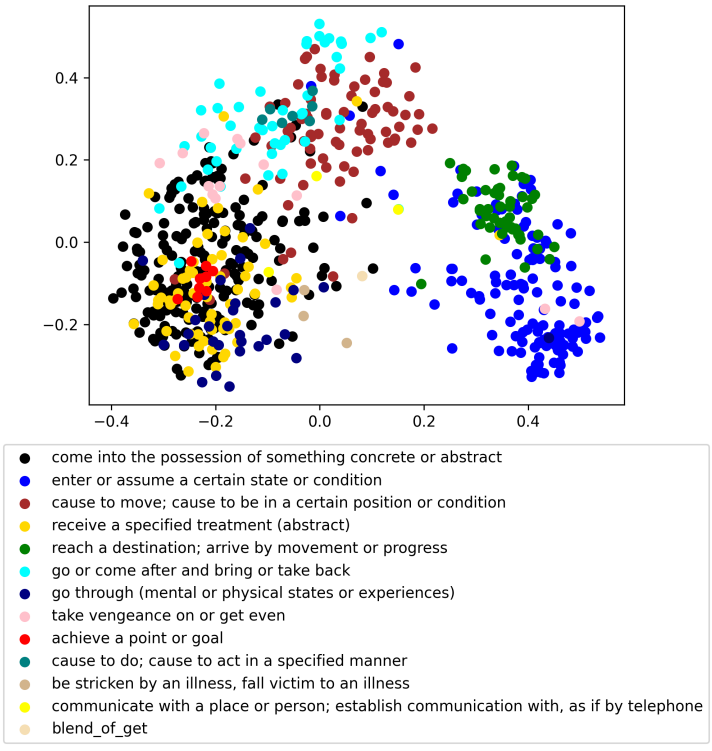


Figure 3: Senses of *get* from word sense disambiguation dataset Semcor.

Target	Cases	Basic Examples
get	we will , i 'm just saying we do wan na get into cocktail	where do you get your carrots from ?
	they 're watching neighbours come on , get up you lazy bugger !	and you 'll get a separate room
	oh we did n't get much further on there , what we started with this morning.	i 'm gon na get some cleaning , i 'll get some cleaning fluid this week .
back	why ca n't they take it through the back door and up the stair ?	within 10 minutes i had turned my back on the corduroy battalions of trees and was striding under a still.
	they are unlikely to find a place to do so which is not in somebody 's back yard .	on the edge of the lawn with his back to the cedar tree .

Table 4: Cases study of targets *get* and *back*

Hardware	TITAN RTX
Runtime/epoch	50 min
Parameters	252,839,426

Table 5: Experiment details

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D ☐ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.