Word-level Prefix/Suffix Sense Detection: A Case Study on Negation Sense with Few-shot Learning

Yameng Li[1,2] **Zicheng Li**[2] **Ying Chen**[1*] **Shoushan Li**[2]
[1]College of Information and Electrical Engineering, China Agricultural University, China
[2]Natural Language Processing Lab, Soochow University, China
{ymli233, 20205227019}@stu.suda.edu.cn
chenying@cau.edu.cn
lishoushan@suda.edu.cn

Abstract

Morphological analysis is an important research issue in the field of natural language processing. In this study, we propose a context-free morphological analysis task, namely word-level prefix/suffix sense detection, which deals with the ambiguity of sense expressed by prefix/suffix. To research this novel task, we first annotate a corpus with prefixes/suffixes expressing negation (e.g., *il-*, *un-*, *-less*) and then propose a novel few-shot learning approach that applies an input-augmentation prompt to a token-replaced detection pre-training model. Empirical studies demonstrate the effectiveness of the proposed approach to word-level prefix/suffix negation sense detection.[1]

1 Introduction

Morphological analysis mainly refers to processing a word into a lemma (root) and a well-defined morphological tag (Anglin et al., 1993; Haspelmath and Sims, 2013; Morita et al., 2015; Nicolai and Kondrak, 2017; Deacon et al., 2017; Ganesh et al., 2019). For instance, through morphological analysis, the word "*unhappy*" will be divided into a lemma "*happy*" and a negation sense prefix tag "*un-*". Morphological analysis has played an important role in natural language processing (NLP) and it has been applied to many downstream tasks such as spelling checking (Aduriz et al., 1993; Oflazer, 1995; Sénéchal and Kearnan, 2007; Levesque et al., 2021) and machine translation (Lee, 2004; Habash, 2007; Toutanova et al., 2008; Belinkov et al., 2017).

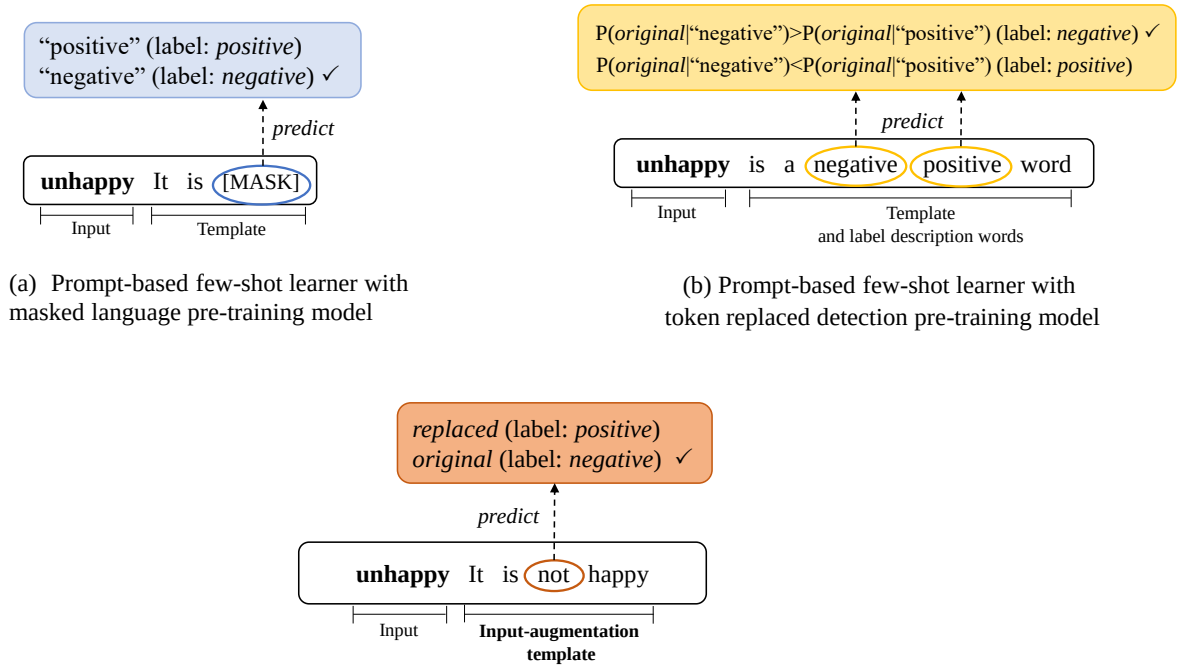
One major challenge in morphological analysis is that prefixes/suffixes are sometimes ambiguous. For instance, in English, the prefix "*un-*" often means a meaning "*not*", i.e., a negation sense. However, not all words with the prefix "*un-*" have a negation sense, such as "*unanimous*" and "*unpick*". Besides, the substring "*un-*" sometimes does not appear as a prefix in some words, such as "*universe*" and "*unique*". In this study, we directly address the above challenge by proposing a novel morphological analysis task, namely word-level prefix/suffix negation sense detection, which aims to detect whether a substring in a word is a prefix/suffix and meanwhile takes a specific pre-defined morphological sense. As a preliminary study, we focus on negative prefixes/suffixes. In many languages, one way to make a negative expression is to add a negative prefix/suffix to a word. For instance, in English, *il-*, *im-*, *un-*, and *-less* are some popular negative prefixes/suffixes.

One straightforward approach to prefix/suffix negation sense detection is to build a dictionary that covers all words with the prefixes/suffixes expressing such a sense. However, this is unrealistic because there are always many newly-emerging words due to non-standard language usage or incorrect spelling in some informal texts like Twitter. Therefore, we address the task of word-level prefix/suffix negation sense detection in a computational way.

Specifically, to further reduce the annotation cost, we propose a few-shot learning approach by employing the token-replaced detection model as our basic prompt-learning model due to its excellent performance in few-shot learning (Li et al., 2022). Furthermore, we propose a novel prompt, namely input-augmentation prompt, which relies only on the input word. As illustrated in Fig.1(c), for the input word is "*unhappy*", the prompt, "*unhappy It is not happy*", is used to predict whether the word "*not*" is *original* or *replaced* so as to determine whether the input word is a negation word or not, where the substring "*happy*" is generated by removing the potential prefix (i.e., *un-*) from the input word. The de-

*Corresponding author
[1]https://github.com/mengmeng233/Word-level-Prefix-Suffix-Sense-Detection

Findings of the Association for Computational Linguistics: ACL 2023, pages 7651–7658



(a) Prompt-based few-shot learner with masked language pre-training model

(b) Prompt-based few-shot learner with token replaced detection pre-training model

(c) Input-augmentation prompt-based few-shot learner with token replaced detection pre-training model (our approach)

Figure 1: Different few-shot learners for prefix/suffix sense detection on negation.

sign of our input-augmentation prompt can avoid one major shortcoming of existing few-shot learning approaches, i.e., the selection of labels (e.g., two labels, "*positive*" and "*negative*" in Fig.1a) or the selection of label description words (e.g., "*negative positive*" in Fig.1b) has a big impact on learner performance (Jiang et al., 2020; Gao et al., 2020; Li et al., 2022). Moreover, our empirical studies also demonstrate that our approach achieves much better performances than the existing few-shot learning approaches.

2 Related work

Morphological analysis aims to learn about the morphological structure of a given word form, and in general, there are four specific tasks: *morphological tagging* (i.e., assigning some pre-defined morphological tags to a word in a sentence) (Müller et al., 2013; Labeau et al., 2015; Cotterell and Heigold, 2017; Conforti et al., 2018; Malaviya et al., 2019), *lemmatization* (i.e., converting a word in a sentence into the normalized form) (Plisson et al., 2004; Chrupała, 2006; Jongejan and Dalianis, 2009; Straková et al., 2014; Bergmanis and Goldwater, 2018), *morphological segmentation* (i.e., judging whether the substring in a word could be segmented as a prefix/suffix) (Ruokolainen et al., 2013, 2016; Goldsmith et al., 2017; Cotterell et al., 2019), and *morphological disambiguation* (i.e., assigning a correct morphological segmentation to a word by leveraging the context) (Hakkani-Tür et al., 2002; Yildiz et al., 2016; Cotterell et al., 2018; Wiedemann et al., 2019).

Compared to the above tasks, our work has at least three different aspects. First, our task is a combination of *morphological tagging* and *morphological segmentation*. Second, our task is word-level, i.e., the input contains only a single word without context, which leads to the inapplicability of previous approaches based on contextual information. Third, we propose a novel few-shot learning approach to our task. To the best of our knowledge, this is the first attempt of studying few-shot learning in morphological analysis.

3 Corpus Generation

We use six prefixes, i.e., *un-*, *im-*, *in-*, *il-*, *ir-* and *dis-* as negation prefixes and two suffixes, i.e., *-less* and *-free* as negation suffixes to collect words from two resources, i.e., the ninth edition of *Oxford Advanced Learner's Dictionary* (AS et al., 2005) and 1.6 million English Tweeter data collected by Go et al. (2015). In summary, we obtain 2,717 and 6,671 words with negation prefixes/suffixes from the Oxford dictionary and

tweeter data, respectively. Then, we randomly select 3,000 words and annotates such words as our corpus. Specifically, we assign two annotators to annotate each word into two categories, i.e., $positive$ and $negative$. The $Kappa$ consistency check value of the human annotation is 0.87. Moreover, for words with different sense annotations, we assign another annotator to make a final decision. Table 1 shows the data statistics of the corpus.

	$Neg.$	$Pos.$		$Neg.$	$Pos.$
un-	482	186	il-	20	59
in-	372	858	dis-	172	256
im-	100	194	$-less$	163	24
ir-	48	53	$-free$	9	4
ALL:		$Neg.$ 1634		$Pos.$ 1366	

Table 1: Statistics of the annotated corpus.

4 Methodology

Problem statement: The prefix/suffix negation sense detection task can be formulated as follows. Let $D_l = \{w, y\}$ be labeled data, where w is the input word and y is a label in $\{positive, negative\}$. Our approach aims to provide a few-shot learner for such a detection task.

Approach overview: As shown in Figure 1(c), a prompt-based learner, which is based on a pre-trained token-replaced detection model and an input-augmentation prompt, is built for the prefix/suffix negation sense detection task. The goal of a pre-trained token-replaced detection model (e.g., ELECTRA) is to predict whether a token in the input string is $replaced$ or not.

Approach specification: First, an input-augmentation prompt x_{prompt} is constructed for an input word w, as follows.

$$x_{prompt} = w\ It\ is\ not\ \overline{w}, \tag{1}$$

where "$It\ is\ not$" is a template, and $\overline{w}$ is a substring of the input word w without the prefix/suffix, such as $\overline{w}$ = "$happy$" for w = "$unhappy$".

Second, prompt $x = [w_1, w_2, ..., w_n]$ is fed into the encoder in the discriminator of the pre-trained token-replaced detection model to obtain an output sequence $y = [y_1, y_2, ..., y_n]$, where w_i is the ith word in the prompt, and y_i is the prediction label (either $original$ or $replaced$) for word w_i, indicating whether the word is $original$ or $replaced$.

Finally, we map the label set of the pre-trained token-replaced detection model to the label set of our task, with the following formulas.

$$P(\text{“}negative\text{”}|x_{prompt}) = P(y_{\text{“}not\text{”}} = original) \tag{2}$$

and

$$P(\text{“}positive\text{”}|x_{prompt}) = P(y_{\text{“}not\text{”}} = replaced), \tag{3}$$

where $y_{\text{“}not\text{”}}$ denotes the label corresponding to the word "not" in the input-augmentation prompt as shown in formula (1). For instance, suppose that the input word is "$unhappy$", we first obtain the input-augmentation prompt "$unhappy\ It\ is\ not\ happy$" and then use the pre-trained token-replaced model to predict whether the word "not" in the prompt is $original$ or $replaced$. If the prediction result is $original$, we conclude that the input word "$unhappy$" is a negative word.

In the training phase of our few-shot learning setting, only a few prompt samples, together with their labels are used to update the parameters in the discriminator of the pre-trained token-replaced detection model. It is important to note that our approach reuses the pre-trained parameters in the pre-trained token-replaced detection model and does not use any other new parameters.

5 Experiments

Data setting: 2,000 samples are randomly selected from the human-annotated corpus. First, 400 samples are selected as test data, including 200 for each class. Then, we follow the evaluation protocol of Li et al. (2022) by running 5 experiments with 5 different training and development splits. In each split, 16 training samples (i.e., 8 samples in each class) and 16 development samples (i.e., 8 samples in each class) are selected in few-shot learning. In fully-supervised learning, 1,400 training samples and 200 development samples are used.

Evaluation Metrics: Standard *Macro-F1*, *Accuracy*, F1 for negative samples (1-F1), and F1 for positive samples (0-F1) are used to evaluate the performance.

Model Settings: We employ ELECTRA as the pre-trained token-replaced detection model. The weight_decay is 2e-3, the maximum length is set to 64, and the remaining hyper-parameters are obtained by searching.

Approach	Basic model	0-F1	1-F1	*Macro-F1*	*Acc.*
Finetuning-RoBERTa	RoBERTa-large	73.3(4.3)	75.2(3.6)	74.3(3.9)	74.3(3.9)
Finetuning-ELECTRA	ELECTRA-large	70.8(4.0)	75.1(4.5)	73.0(3.4)	73.4(3.3)
Prompt-RoBERTa	RoBERTa-large	75.6(1.8)	79.0(1.0)	77.3(1.1)	77.5(1.1)
Prompt-ELECTRA	ELECTRA-large	78.2(1.6)	79.6(3.1)	78.8(2.3)	78.8(2.3)
Warp	RoBERTa-large	69.8(3.4)	73.2(5.7)	71.5(4.3)	71.8(4.4)
DART	RoBERTa-large	70.6(2.1)	71.2(7.6)	70.9(4.8)	71.2(4.9)
P-tuning-v2	RoBERTa-large	70.7(1.3)	75.2(2.8)	73.0(1.5)	73.2(1.8)
Our Approach	**ELECTRA-large**	**87.4(2.9)**	**87.4(3.6)**	**87.4(3.2)**	**87.4(3.2)**
Fully-supervised Learning	ELECTRA-large	87.1	87.9	87.5	87.5

Table 2: The performances of different methods for prefix/suffix negation sense detection (k=16).

We implement the following approaches for comparison:

(1) **Finetuning-RoBERTa** (Liu et al., 2019): Based on the fine-tuning approach and RoBERTa-large model, the prediction label is obtained by mapping the "[CLS]" token to label space.

(2) **Finetuning-ELECTRA** (Clark et al., 2020): It is similar to finetuning-RoBERTa except that the ELECTRA-large model is used.

(3) **Prompt-RoBERTa** (Gao et al., 2020): It is a discrete prompt learning approach based on RoBERTa-large, as shown in Figure 1(a), where the prompt is " w *it is* $[mask]$ ", and the prediction label is obtained by the filling of "$[mask]$" (either "$negative$" or "$positive$").

(4) **Prompt-ELECTRA** (Li et al., 2022): It is a discrete prompt learning approach based on ELECTRA-large, as shown in Figure 1(b), where the prompt is " w *is a negative positive word* ".

(5) **Warp** (Hambardzumyan et al., 2021): It is a continuous prompt learning approach, in which the best prompt template is obtained by searching in the (continuous) embedding space. Moreover, the template is learned using adversarial refactoring.

(6) **DART** (Zhang et al., 2021): It is a continuous prompt learning approach, in which the search for the best prompt template is based on backpropagation.

(7) **P-tuning-v2** (Liu et al., 2021): It is a continuous prompt learning approach, in which the search for the best prompt is based on a prefixed-tuned multi-layer prompt.

(8) **Fully-supervised Learning:** 1,400 training and 200 development samples are used to re-train the ELECTRA-large model.

Table 2 shows the performances of different approaches, from which we can see that : (1) Our approach significantly outperforms the fully-supervised learning and fine-tuning approaches, which proves the effectiveness of our few-shot learner. (2) Our approach performs much better than other prompt-based learners, e.g., obtaining 8.6% increase on *Macro-F1* when compared with Prompt-ELECTRA. The improvement confirms the effectiveness of our input-augmentation prompt. (3) Our approach, using only 16 training and 16 development samples, almost performs equivalent to the fully-supervised learning approach with 1,400 training and 200 development samples.

An error analysis is made for our approach, which shows two main error causes: (1) the input word w or its substring $\overline{w}$ has multiple meanings, such as "$hapless$" vs. "hap", and "$disembarkation$" vs. "$embarkation$". (2) the meaning of w and $\overline{w}$ is irrelevant, such as "$dispossession$" vs. "$possession$", and "$ingot$" vs. "got". This indicates that more efforts are needed for our prefix/suffix negation sense detection.

6 Conclusion

In this study, we propose a novel word-level morphological analysis task, namely prefix/suffix sense detection, and make a case study on negation sense. We provide an annotated corpus for the prefix/suffix negation sense detection, and then propose a novel few-shot learning approach, which uses an input-augmentation prompt and a pre-trained token-replaced detection model to effectively make the negation sense detection. Empirical studies show that our approach performs much better than other approaches in the few-shot scenario, such as using only 16 training samples.

Limitations

The limiation of this work is that we only consider one type of prefixes/suffixes, i.e., negative prefixes/suffixes. In our future work, we would like to work on other types of prefix/suffix sense detection tasks, such as prefix/suffix sense detection on occupation. For instance, in English, there are many suffixes such as *-or*, *-er*, and *-ee*, which mean a person with a certain occupation.

Acknowledgement

We thank the reviewers for their insightful comments and suggestions. This work was supported by the NSFC grant (No.62076176), and the General Research Fund (GRF) project sponsored by the Research Grants Council Hong Kong (Project No.15611021).

References

Itziar Aduriz, Eneko Agirre, Iñaki Alegria, Xabier Arregi, Jose Maria Arriola, Xabier Artola, Arantza Díaz de Ilarraza, Nerea Ezeiza, Montse Maritxalar, Kepa Sarasola, et al. 1993. A morphological analysis based method for spelling correction. In *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, pages 463–463.

Jeremy M Anglin, George A Miller, and Pamela C Wakefield. 1993. Vocabulary development: A morphological analysis. *Monographs of the society for research in child development*, pages i–186.

Wehmeier Sally Hornby AS, Ashby Michael, Albert Sydney Hornby, Sally Wehmeier, and Michael Ashby. 2005. *Oxford Advanced Learner's English-Chinese Dictionary: AS Hornby, Sally Wehmeier, Michael Ashby*. oxford university press.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? *arXiv preprint arXiv:1704.03471*.

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1391–1400.

Grzegorz Chrupała. 2006. Simple data-driven context-sensitive lemmatization.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Costanza Conforti, Matthias Huck, and Alexander Fraser. 2018. Neural morphological tagging of lemma sequences for machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 39–53.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual, character-level neural morphological tagging. *arXiv preprint arXiv:1708.09157*.

Ryan Cotterell, Christo Kirov, Sabrina J Mielke, and Jason Eisner. 2018. Unsupervised disambiguation of syncretism in inflected lexicons. *arXiv preprint arXiv:1806.03740*.

Ryan Cotterell, Arun Kumar, and Hinrich Schütze. 2019. Morphological segmentation inside-out. *arXiv preprint arXiv:1911.04916*.

S Hélène Deacon, Xiuli Tong, and Kathryn Francis. 2017. The relationship of morphological analysis and morphological decoding to reading comprehension. *Journal of Research in Reading*, 40(1):1–16.

LS Ganesh, Rahul R Marathe, et al. 2019. Dynamic capabilities: A morphological analysis framework and agenda for future research. *European Business Review*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

A Go, R Bhayani, and L Huang. 2015. Sentiment140-twitter sentiment analysis tool. *Sentiment140,[Online]. Available: http://help. sentiment140. com/home.[Accessed 30 March 2018]*.

John A Goldsmith, Jackson L Lee, and Aris Xanthos. 2017. Computational learning of morphology. *Annual Review of Linguistics*, 3:85–106.

Nizar Habash. 2007. Arabic morphological representations for machine translation. In *Arabic computational morphology*, pages 263–285. Springer.

Dilek Z Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*.

Martin Haspelmath and Andrea Sims. 2013. *Understanding morphology*. Routledge.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Bart Jongejan and Hercules Dalianis. 2009. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 145–153.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen. 2015. Non-lexical neural architecture for fine-grained pos tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 232–237.

Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. Technical report, IBM THOMAS J WATSON RESEARCH CENTER YORKTOWN HEIGHTS NY.

Kyle C Levesque, Helen L Breadmore, and S Hélène Deacon. 2021. How morphology impacts reading and spelling: Advancing the role of morphology in models of literacy development. *Journal of Research in Reading*, 44(1):10–26.

Zicheng Li, Shoushan Li, and Guodong Zhou. 2022. Pre-trained token-replaced detection model as few-shot learner. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3274–3284.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

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*.

Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. A simple joint model for improved contextual neural lemmatization. *arXiv preprint arXiv:1904.02306*.

Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. 2015. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297, Lisbon, Portugal. Association for Computational Linguistics.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332.

Garrett Nicolai and Grzegorz Kondrak. 2017. Morphological analysis without expert annotation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 211–216.

Kemal Oflazer. 1995. Error-tolerant finite state recognition with applications to morphological analysis and spelling correction. *arXiv preprint cmp-lg/9504031*.

Joël Plisson, Nada Lavrac, Dunja Mladenic, et al. 2004. A rule based approach to word lemmatization. In *Proceedings of IS*, volume 3, pages 83–86.

Teemu Ruokolainen, Oskar Kohonen, Kairit Sirts, Stig-Arne Grönroos, Mikko Kurimo, and Sami Virpioja. 2016. A comparative study of minimally supervised morphological segmentation. *Computational Linguistics*, 42(1):91–120.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37.

Monique Sénéchal and Kyle Kearnan. 2007. The role of morphology in reading and spelling.

Jana Straková, Milan Straka, and Jan Hajic. 2014. Open-source tools for morphology, lemmatization, pos tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*.

Eray Yildiz, Caglar Tirkaz, H Sahin, Mustafa Eren, and Omer Sonmez. 2016. A morphology-aware network for morphological disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations.

☒ A2. Did you discuss any potential risks of your work?
At present, we haven't found any potential risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

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

B ☑ Did you use or create scientific artifacts?

3, 5.

☑ B1. Did you cite the creators of artifacts you used?
3, 5.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
3

☑ 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)?
3

☐ 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?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ 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.
3, 5.

C ☑ Did you run computational experiments?

5

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We report the model sizes used in all our work, which run on the same GPU.

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?
5

☑ 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?
5

☑ 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.)?
5

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.?
Not applicable. 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)?
Not applicable. 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?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.