Context-aware Adversarial Attack on Named Entity Recognition

Shuguang Chen
University of Houston
schen52@uh.edu

Leonardo Neves
Snap Inc.
lneves@snap.com

Thamar Solorio
University of Houston
Mohamed bin Zayed University of Artificial Intelligence
tsolorio@uh.edu

Abstract

In recent years, large pre-trained language models (PLMs) have achieved remarkable performance on many natural language processing benchmarks. Despite their success, prior studies have shown that PLMs are vulnerable to attacks from adversarial examples. In this work, we focus on the named entity recognition task and study context-aware adversarial attack methods to examine the model's robustness. Specifically, we propose perturbing the most informative words for recognizing entities to create adversarial examples and investigate different candidate replacement methods to generate natural and plausible adversarial examples. Experiments and analyses show that our methods are more effective in deceiving the model into making wrong predictions than strong baselines.

1 Introduction

Existing methods for adversarial attacks mainly focus on text classification (Liang et al., 2018; Garg and Ramakrishnan, 2020), machine translation (Belinkov and Bisk, 2018; Cheng et al., 2019), reading comprehension (Jia and Liang, 2017; Wallace et al., 2019), etc. A slight perturbation to the input can deceive the model into making wrong predictions or leaking important information. Such adversarial attacks are widely used to identify potential vulnerabilities and audit the model robustness. However, in the context of named entity recognition (NER), these adversarial attack methods are inadequate since they are not customized for the labeling schemes in NER (Lin et al., 2021). This is especially problematic as the generated adversarial examples can be mislabeled.

Prior studies have proposed various context-aware attacks (i.e., perturb non-entity words) and entity attack (i.e., perturb only entity words) methods to address this issue. Despite their success, most existing methods randomly select words to perturb without taking the linguistic structure into consideration, limiting their effectiveness to consistently generate natural and coherent adversarial examples. Some words in a sentence are more informative than others in guiding the model to recognize named entities. For instance, in Figure 1, the word "rackets" can provide more information than the word "tournament" to infer the entity type of "Wilson". Perturbing such words can be effective in leading to more incorrect model predictions.

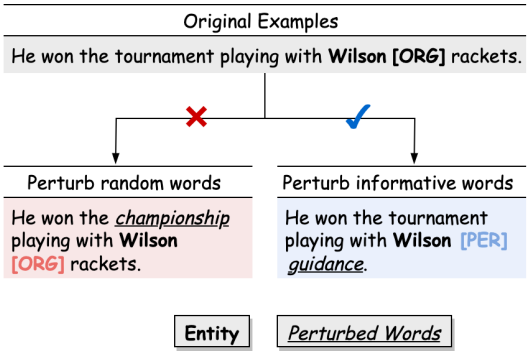

Figure 1: Comparison between adversarial attack with and without perturbing informative words.

In this work, we explore the correlation between model vulnerability and informative words. We aim to conduct adversarial attacks by perturbing the informative words to expose the potential vulnerabilities of NER systems. To this end, we investigate different candidate selection methods to determine which words should be perturbed, including part-of-speech (POS) tagging, dependency parsing, chunking, and gradient attribution. To demonstrate the effectiveness of our proposed methods, we adapt two commonly-used candidate replacement approaches to replace the selected candidate words: synonym replacement (i.e., replace with a synonym) and masked language model replacement (i.e., replace with a candidate generated from a masked language model). We conduct experiments on three corpora and systematically evaluate our proposed methods

Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024), pages 11–16

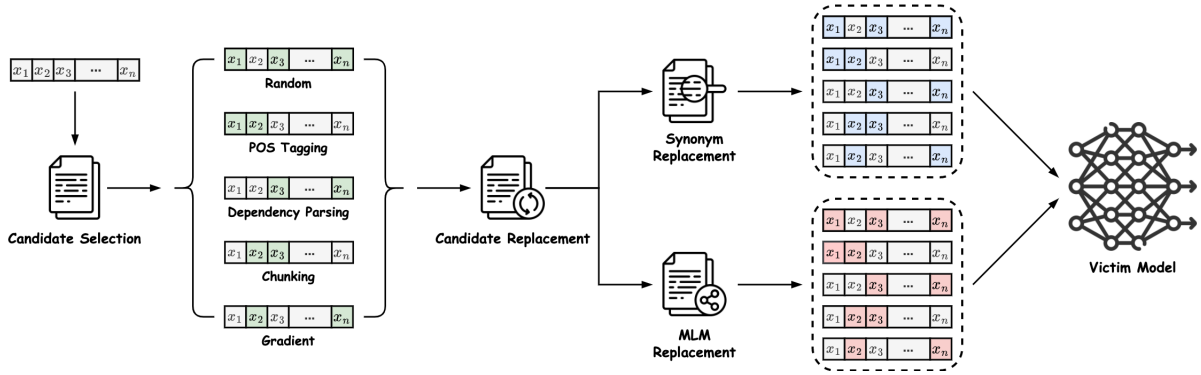

Figure 2: The pipeline of the proposed context-aware adversarial attack, including candidate selection to determine which words to perturb and candidate replacement for replacing candidate words.

with different metrics. Experimental results and analyses show that our proposed methods can effectively corrupt NER models.

In summary, our contributions are as follows:

1. We investigate different methods to perturb the most informative words for generating adversarial examples to attack NER systems.
2. Experiments and analyses show that the proposed methods are more effective than strong baselines in attacking models, posing a new challenge to existing NER systems.

2 Related Work

Adversarial attacks have been receiving increasing attention in the field of NER. Prior work in this research direction can be generally classified into two categories: i) context-aware attacks and ii) entity attacks. In the context-aware attacks, only the non-entity context words are modified. To achieve this, Lin et al. (2021) proposed to perturb the original context by sampling adversarial tokens via a masked-language model. Simoncini and Spanakis (2021) presented multiple modification methods to substitute, insert, swap, or delete characters and words. Wang et al. (2021) studied to create adversarial samples by concatenating different sentences into a single data point. For entity attacks, the entity words are modified while the non-entity context words are kept unchanged. In particular, Lin et al. (2021) exploited an external dictionary from Wikidata to find replacements for entity instances. Simoncini and Spanakis (2021) studied the use of the SCPNs (Iyyer et al., 2018) to generate candidate paraphrases as adversarial samples. Reich et al. (2022) proposed leveraging expert-guided heuristics to modify the entity tokens and their surrounding contexts, thereby altering their entity types as adversarial attacks. Wang et al. (2021) performed adversarial attacks by swapping words or manipulating characters.

3 Context-aware Adversarial Attack

In this work, we propose different methods to generate adversarial samples for the purpose of auditing the model robustness of NER systems. In the following sections, we describe the two main stages involved in this process: 1) candidate selection, which aims to determine which candidate words should be replaced; and 2) candidate replacement, which aims to find the best way to replace candidate words. The pipeline of adversarial data generation is shown in Figure 2.

3.1 Candidate Selection

To effectively attack the model, we consider perturbing the most informative words for recognizing entities. We investigate the following automated methods to select such words as candidates:

- **Random (RDM)**: select non-entity words at random from the sentence as candidate words.
- **POS tagging (PST)**: select semantic-rich non-entity words as candidate words based on their POS tags. Here, following Lin et al. (2021), we consider selecting adjectives, nouns, adverbs, and verbs.
- **Dependency parsing (DEP)**: select the non-entity words related to entity instances, including ascendants and descendants, as candidate words based on dependency parsing.
- **Chunking (CHK)**: select the non-entity words in the noun chunks that are close to entity instances as candidate words to preserve both semantic and syntactic coherence.
- **Gradient (GDT)**: select the non-entity words

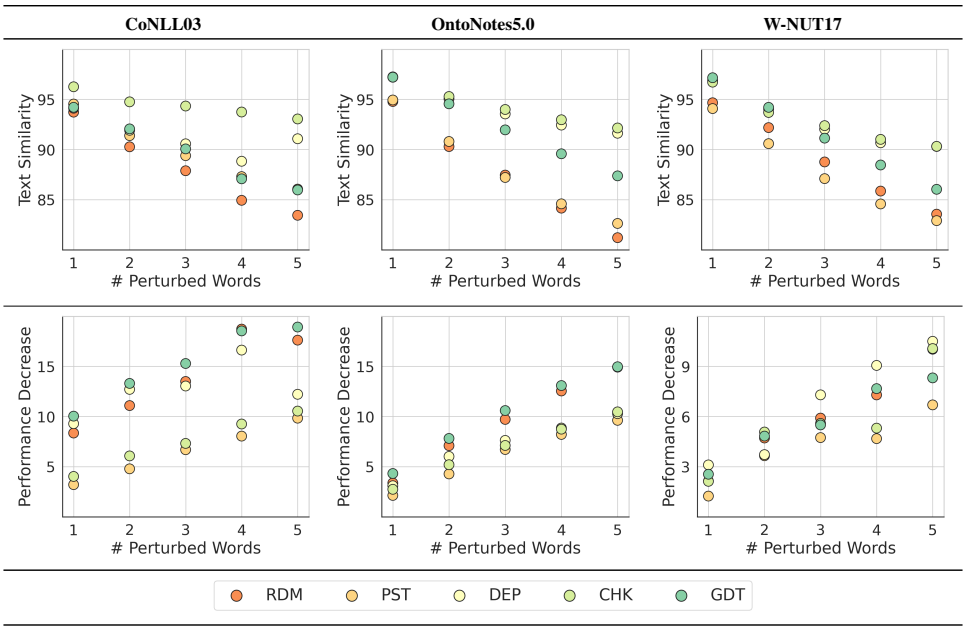

Table 1: Comparison between different candidate selection methods using synonym replacement. RDM, PST, DEP, CHK, GDT are short for random, POS tagging, dependency parsing, chunking, and gradient candidate selection, respectively. The x-axis denotes the number of perturbed words while the y-axis denotes the difference in F1 scores.

according to the integral of gradients. We use Integrated Hessians (Janizek et al., 2021) to determine the importance of non-entity words based on their feature interactions with entity instances, and select the words with higher importance scores to perturb.

To obtain linguistic features, including part-of-speech tags, dependency parsing, and chunking, for our proposed method, we use the statistical model from spaCy [1] to process text. Then we select the candidate words to perturb based on this information. For GDT, we use the gradient of the pre-trained $\text{BERT}_{\text{base}}$ model (Devlin et al., 2019) to determine the importance of each word.

3.2 Candidate Replacement

Perturbations in text at the character-level can be easily detected and defended by spell check and correction (Pruthi et al., 2019; Li et al., 2020). Therefore, we exclusively focus on the word-level perturbations in this work. Simply replacing a word with another one at random can lead to noisy data. For instance, in Figure 1, the label for “Wilson" is changed from *ORG* to *PER* by replacing “rackets" with “guidance", which has a conflict with its original gold label. Therefore, to keep original labels valid, we investigate the following two approaches to replace candidate words:

- **Synonym Replacement**: Using synonyms to replace candidate words as adversarial samples can guarantee the preservation of text semantics and make it hard to be perceived by human investigation. We use the WordNet (Miller, 1998) dictionary to find synonyms for candidate words, and then randomly select one of them as a replacement.
- **Masked Language Model Replacement**: The masked language model (MLM) attempts to predict the masked words in the given input sequence. In our work, we first create masks for candidate words, and then use a masked language model $\text{RoBERTa}_{\text{base}}$ (Liu et al., 2019) to generate a replacement based on the context. This approach is capable of preserving both semantics and syntax in the generated adversarial samples.

4 Experiments

In this section, we present the experimental setup and results. We systematically conduct experiments to evaluate our proposed methods on three corpora with different metrics and provide analyses to better understand their effectiveness.

4.1 Experiment Setup

Datasets We evaluate the proposed methods on three commonly-used corpora for NER tasks, in-

[1] https://github.com/explosion/spaCy

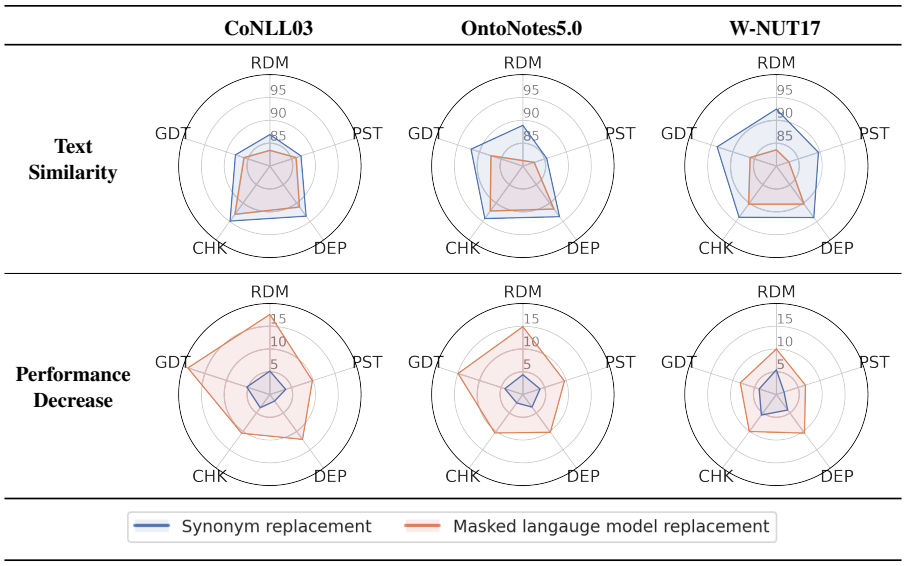

Table 2: Comparison between different candidate replacement methods when perturbing five words in each sentence. RDM, PST, DEP, CHK, GDT are short for random, POS tagging, dependency parsing, chunking, and gradient candidate selection, respectively.

cluding CoNLL03 (Tjong Kim Sang and De Meulder, 2003), OntoNotes5.0 (Pradhan et al., 2013), and W-NUT17 (Derczynski et al., 2017). The data statistics are summarized in Appendix A.

Victim Model The victim model consists of the $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) as the base model and a linear layer as the classifier to assign NER tags. The details of hyper-parameters and fine-tuning are described in Appendix B.

Evaluation Metrics To examine the effectiveness of our proposed methods, we consider the following metrics for evaluation:

- **Textual Similarity (Sim.)**: cosine similarity between adversarial examples and the corresponding original examples using the Universal Sentence Encoder (Giorgi et al., 2021). A higher textual similarity score indicates that more semantics are preserved.
- **Performance Decrease (ΔPerf.)**: the difference in F1 scores between adversarial examples and their corresponding original examples. A higher performance decrease indicates that the model makes more mistakes.

4.2 Main Results

We compare candidate selection and replacement methods by perturbing the same number of words in the sentences. Below we present experimental results and summarize our findings:

Candidate Selection V.S. Metrics From the results in Table 1, we observe that the model performance decreases rapidly under adversarial attacks. When perturbing five words in the sentence, the F1 scores decrease by 10% ~20%. Among these attack methods, GDT and RDM are more effective at deceiving the model into making wrong predictions. When performing attacks with RDM, however, the text similarity is sacrificed in exchange for a greater performance decrease, which can potentially make adversarial examples easier to detect. Additionally, it is worth noting that DEP is also effective at a slight perturbation, although it can only result in a smaller performance decrease as we increase the number of perturbed words. In terms of textual similarity and performance decrease, PST is the least effective method in most cases.

Candidate Replacement V.S. Metrics The comparison between different candidate replacement methods is shown in Table 2. In general, compared to masked language model replacement, synonym replacement can achieve a higher textual similarity, indicating that more semantics are preserved in adversarial examples. However, its performance decrease is quite limited. At a slightly lower textual similarity, masked language model replacement leads to a much larger performance decrease. Besides, both replacement methods are relatively less effective on the W-NUT17 corpus. Compared to the text from CoNLL03 and OntoNotes5.0 which

is long and formal, the text from W-NUT17 is short and noisy as it contains many misspellings and grammar errors. For this reason, the model cannot rely too heavily on context when making predictions, limiting the effectiveness of adversarial attacks on this corpus.

5 Conclusion

In this work, we study adversarial attacks to examine the model robustness using adversarial examples. We focus on the NER task and propose context-aware adversarial attack methods to perturb the most informative words for recognizing entities. Moreover, we investigate different candidate replacement methods for generating adversarial examples. We undertake experiments on three corpora and show that the proposed methods are more effective in attacking models than strong baselines.

Limitations

The proposed methods require linguistic knowledge (e.g., part-of-speech tags and dependency parsing) to processing the text. Most existing tools can automate this process for English. However, these tools may need to be extended to support other languages, especially for minority languages. Additionally, the proposed methods maybe not applicable with low computational resources or in real-time scenarios.

Acknowledgements

This work received partial support from the National Science Foundation under award number 1910192.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

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.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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 879–895, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Joseph D. Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *J. Mach. Learn. Res.*, 22:104:1–104:54.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xiangci Li, Hairong Liu, and Liang Huang. 2020. Context-aware stand-alone neural spelling correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 407–414, Online. Association for Computational Linguistics.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4208–4215. ijcai.org.

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3728–3737, Online and Punta Cana, Dominican Republic. 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. *CoRR*, abs/1907.11692.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Aaron Reich, Jiaao Chen, Aastha Agrawal, Yanzhe Zhang, and Diyi Yang. 2022. Leveraging expert guided adversarial augmentation for improving generalization in named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1947–1955, Dublin, Ireland. Association for Computational Linguistics.

Walter Simoncini and Gerasimos Spanakis. 2021. SeqAttack: On adversarial attacks for named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 308–318, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. 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 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

A Data Statistics

Table 3 shows data statistics of the NER datasets we used in our experiments:

Split	CoNLL03	OntoNotes5.0	W-NUT17
Train	14,041	115,812	3,394
Validation	3,250	15,680	1,009
Test	3,453	12,217	1,287
Total	20,744	143,709	5,690

Table 3: Data Statistics of CoNLL03, OntoNotes5.0 and W-NUT17 corpus.

B Hyper-parameters and Fine-tuning

For the victim model, we use the $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) without changing any hyper-parameters. The learning rate is set to 5e-5 and the training batch size is set to 8. We train the model using the Adam optimizer (Kingma and Ba, 2015) with a weight decay 0.01 for 10 epochs on CoNLL03 and OntoNotes5.0 data and 20 epochs on W-NUT17 data. For the hardware, we use 8 NVIDIA V100 GPUs with a memory of 24GB.