SeqAttack: On Adversarial Attacks for Named Entity Recognition

Walter Simoncini and **Gerasimos Spanakis**
Maastricht University
{w.simoncini@student.,jerry.spanakis@}maastrichtuniversity.nl

Abstract

Named Entity Recognition is a fundamental task in information extraction and is an essential element for various Natural Language Processing pipelines. Adversarial attacks have been shown to greatly affect the performance of text classification systems but knowledge about their effectiveness against named entity recognition models is limited. This paper investigates the effectiveness and portability of adversarial attacks from text classification to named entity recognition and the ability of adversarial training to counteract these attacks. We find that character-level and word-level attacks are the most effective, but adversarial training can grant significant protection at little to no expense of standard performance. Alongside our results, we also release SeqAttack, a framework to conduct adversarial attacks against token classification models (used in this work for named entity recognition) and a companion web application to inspect and cherry pick adversarial examples.

1 Introduction

Named Entity Recognition (NER) is the task of recognizing named entities in a chunk of text. Named entities are words (one or more) belonging to a particular semantic category, such as location, person or organization. NER is used both as a standalone tool and as an essential component in several Natural Language Processing (NLP) pipelines, such as Information Retrieval (Petkova and Croft, 2007) and Machine Translation (Babych and Hartley, 2003). Traditionally, NER has been attempted with rule-based approaches, Hidden Markov Models and Conditional Random Fields (Li et al., 2020a). In recent years, deep learning has outperformed these methods (Li et al., 2017) (Liu et al., 2019a), especially with the introduction of general-purpose language models such as BERT (Devlin et al., 2019).

Neural networks are vulnerable to adversarial attacks, which can be defined as processes that craft incorrectly-predicted samples from correctly-predicted inputs by applying small perturbations, an example of which can be seen in Figure 1. This shows that deep learning models are fragile and might not be ready for deployment in a critical scenario. The most popular technique to overcome this issue is adversarial training, which uses adversarial attacks to craft additional training samples and retrains the model from scratch (Li et al., 2020b) (Li et al., 2021). Adversarial attacks and training were largely explored with regards to text classification, but current research on NER has only explored attacks based on adversarial typos (Araujo et al., 2020) and the effectiveness of more complex attacks (at the word and sentence levels) is unknown. Word-level attacks are particularly important because they generate adversarial examples highly likely to appear in the real world, providing valuable additional training data (an example can be seen in Figure 1). This paper aims to tackle this problem by investigating the following research questions:

- **RQ1**: How robust are named entity recognition models against adversarial attacks at the character, word and sentence level? In particular, this paper focuses on a BERT$_{\text{base}}$ cased model trained on CoNLL2003 (Tjong Kim Sang and De Meulder, 2003) in order to maintain consistency across the paper.

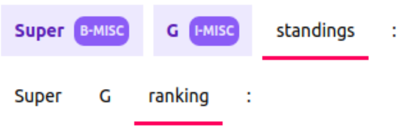

Figure 1: Word-level adversarial example for NER from CoNLL2003 (Tjong Kim Sang and De Meulder, 2003). Changing *standings* to *ranking* induces an incorrect classification of *Super G* as a non-entity.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 308–318
November 7–11, 2021. ©2021 Association for Computational Linguistics

- **RQ2**: How do word and character level adversarial training affect a named entity recognition model's robustness?

2 Related Work

2.1 Adversarial attacks

Several attack strategies are available to fool text classification models. In this paper, we follow the taxonomy by (Yuan et al., 2019), focusing on the properties in the list below, with the addition of granularity (Zhang et al., 2020):

- **Model knowledge:** if all the model information is known, attacks are defined as *white box*. *Black box* attacks instead have access only to the confidence scores. This paper focuses on *black box* attacks.
- **Specificity:** attacks which aim to change the model's prediction to a specific class are called *targeted*, whereas *untargeted attacks* consider any incorrect prediction valid.
- **Granularity:** adversarial examples can be crafted by applying perturbations at the character (e.g. swap, insertion), word (e.g. word replacement, insertion) or sentence level (e.g. paraphrasing).

Some popular attack strategies organized by *granularity* are presented below.

2.2 Attack strategies

At the **character-level** DeepWordBug (Gao et al., 2018) generates at each step candidate adversaries by swapping adjacent characters, substituting a character with a random one, deleting or inserting a character. At the **word-level** TextFooler (Jin et al., 2020) ranks the words in a sample by prediction relevance and replaces the most important ones using a word embedding optimized for synonyms (Mrkšić et al., 2016). BERT-Attack (Li et al., 2020b) and CLARE (Li et al., 2021) operate similarly, but they respectively use BERT and DistillRoBERTa (Sanh et al., 2019) (Liu et al., 2019b) as language models to suggest potential candidates. CLARE supports token replacements, insertions, and merges. Meanwhile, BERT-Attack and TextFooler only support token replacements. All word-level attacks enforce a semantic similarity constraint using the Universal Sentence Encoder (Cer et al., 2018). Finally, at the **sentence-level**, SCPN (Iyyer et al., 2018) generates paraphrases that match one of its built-in syntactic forms.

In comparison to text classification, to the authors' knowledge, adversarial attacks (and training) for NER only appears in two work (Araujo et al., 2020) and (Wang et al., 2020). The former tackles biomedical NER, showing that BERT-based models are susceptible to character swaps, keyboard typo noise and synonym-based entity-word substitutions. The latter integrates adversarial training in the train loop of an LSTM-CNN: at each training step adversarial examples are obtained by perturbing the word embeddings directly. This paper contributes by evaluating a larger number of attack strategies and the portability of adversarial attacks for text classification to token classification problems. Moreover, we provide new insights and a comparison of the samples generated by the different attack strategies.

2.3 Adversarial training

Adversarial training aims to improve a model's robustness using adversarial examples. This task can be achieved mainly in two ways: via data augmentation and by integrating adversarial training within the model train loop.

The first method attacks the victim model using the training set as the attack input and, once obtained enough samples, retrains the model from scratch. One of the first work to use this technique is (Alzantot et al., 2018), in which the authors adversarially train a sentiment classification model on the IMDB dataset without success. Later work, such as (Li et al., 2020b) and (Li et al., 2021) show more interesting results: the former uses adversarial training to make a natural language inference model more robust, gaining 15% after-attack accuracy at the expense of a minimal test accuracy loss. The latter adversarially trains BERT and TextCNN models on the AG news dataset obtaining similar improvements: without loss of test accuracy the authors manage to reduce the attack rate by 12.3% and 3.5% for BERT and TextCNN respectively. The second method is used by (Wang et al., 2020), where adversarial training is integrated in the training loop using a loss function that takes into account adversarial perturbations. Using this technique, the authors improve the model's generalizability by reducing overfitting.

3 The SeqAttack framework

The most popular frameworks for conducting adversarial attacks are `TextAttack` (Morris et al., 2020) and `OpenAttack` (Zeng et al., 2021), but they do not support token classification problems such as named entity recognition, in which each token is either classified as being the beginning of (B), inside (I) or outside an entity (O) according to the inside-outside-beginning (IOB) schema (Ramshaw and Marcus, 1995). In order to attack NER models we developed `SeqAttack`, a framework for conducting adversarial attacks against token classification models. The framework extends `TextAttack` and inherits its design, where attacks are composed of a goal function (the objective to optimize), transformations (how the input text is perturbed), constraints which limit the candidate perturbations and a search method. The framework can be used by NLP practitioners to attack models, for data augmentation and to quickly prototype attack strategies. Inheriting the structure of `TextAttack` also means that its attack strategies can be easily ported and used against NER models. In `TextAttack`, every attack optimizes a goal function, which in the case of text classification is defined as $1 - p_{\hat{y}}$. Where $\hat{y}$ is the ground truth and $p_{\hat{y}}$ is the normalized confidence score for the ground truth. In `SeqAttack`, in order to support NER, the goal function is reformulated as follows:

$$y_{\text{adv}} = \frac{\sum_{i=0}^{N} \text{goal}(y_i, \hat{y}_i)}{\text{countEntities}(x)}$$

$$\text{goal}(y, \hat{y}) = \begin{cases} 0 & \text{if } \hat{y} = 0 \\ 1 - p_{\hat{y}} & \text{if } \hat{y} \neq 0 \ \wedge \hat{y} = y \\ 1 & \text{if } \hat{y} \neq 0 \ \wedge \hat{y} \neq y \end{cases}$$

Where y is the model prediction, N the number of tokens in the sample and x the attacked sample. countEntities(x) returns the number of entity tokens in a sample. We call this function the **untargeted NER** goal function. goal$(y, \hat{y})$ considers valid any incorrect classification of an entity token. It's important to note that this function assigns no score to newly introduced entities. This is due to the fact that the CoNLL2003 metrics consider only the classification of ground truth named entities. We also define the **untargeted-strict NER** goal function, which assigns no score to flips between I-CLS and B-CLS. Figure 3 highlights the difference between the two goal functions.

3.1 Adversarial attacks

This paper employs attack strategies implemented in `TextAttack` that proved to be successful for text classification to attack NER models with minor adaptations. In particular the following modifications were applied:

3.1.1 DeepWordBug

We use two different versions of this attack strategy: **DeepWordBug-I**, true to the original implementation and **DeepWordBug-II**, which is not allowed to modify named entities. Both attacks have a Levenshtein distance constraint, whose maximum allowed distance is specified with a subscript, as in DeepWordBug-I$_5$.

3.1.2 BERT-Attack

The sentence similarity constraint was set to 0.4 and the replacement of numeric tokens with alphanumeric ones was forbidden (i.e. "4" cannot be replaced by "car"). Only non-entity tokens are allowed to be replaced (to avoid the generation of trivial examples, e.g. swapping a location with a person's name) and candidate replacements which are named entities are also rejected (e.g. the candidate replacement "Amsterdam" will be rejected). The attack can perturb up to 40% of the words in a sample.

3.1.3 CLARE

The implementation of CLARE used in this paper only supports replacements and insertions. Similarly to BERT-Attack, the replacement of entity tokens is forbidden and candidate replacements which are named entities are rejected. When a new token is inserted, it is automatically labelled as being outside an entity (O). If a token insertion splits a named entity the beginning/inside labels will be adjusted accordingly.

3.1.4 SCPN

Using the `OpenAttack` implementation, the algorithm iteratively generates candidate paraphrases, using the original sample or a paraphrase as the starting point. The candidates are processed to remove identical consecutive unigrams and bigrams, and only the candidates which preserve at least one named entity are kept. Every token which is not an entity in the original sample is labelled as being outside an entity (O). An example can be seen in Figure 2.

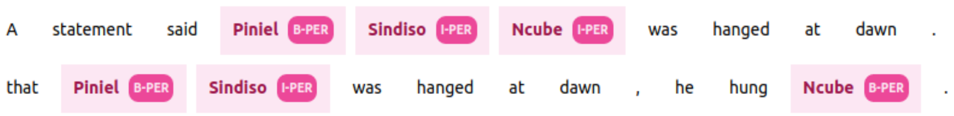
A statement said Piniel B-PER Sindiso I-PER Ncube I-PER was hanged at dawn .
that Piniel B-PER Sindiso I-PER was hanged at dawn , he hung Ncube B-PER .

Figure 2: A paraphrase generated by SCPN (bottom) and its original counterpart (top). Named entities in the paraphrase were re-labelled with the corresponding ground truth and the other tokens were labelled as non-entities. Original sample from CoNLL2003 (Tjong Kim Sang and De Meulder, 2003).

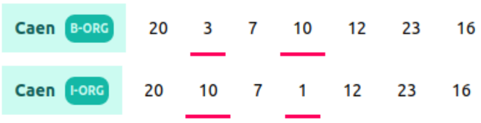

Figure 3: Changing two numbers causes *Caen*'s label to flip from B-ORG to I-ORG. The untargeted goal function would consider *Caen* to be an incorrect classification (and thus a success) meanwhile the untargeted-strict goal function would not. Example from CoNLL2003 (Tjong Kim Sang and De Meulder, 2003).

3.2 Adversarial training

This paper approaches adversarial training using the training dataset augmentation strategy: we attack the model using its training set as the input, generating at most one adversarial example per train sample, and we retrain the model with the augmented dataset. DeepWordBug-I_5 and BERT-Attack were chosen as the attack strategies so as to investigate the different effect of word-level and character-level adversarial training.

4 Experiments

4.1 Adversarial attacks

The attack techniques in section 3 were evaluated on a BERT$_{\text{base}}$ cased model (Devlin et al., 2019), fine tuned on the CoNLL2003 dataset for three epochs using the `transformers` library (Wolf et al., 2020). All attacks use the **untargeted-strict** goal function and target a subset of 256 samples from the test set, selected such that the model incorrectly predicts up to 10% of the entities contained in each sample. For each sample, the attack is allowed up to 120 seconds and a maximum of 512 model invocations (queries).

4.1.1 Evaluation metrics

The attacks are evaluated following previous work (Li et al., 2021), (Jin et al., 2020), (Morris et al., 2020), which employ the following automated metrics (in addition to accuracy, recall and F1 score as in the CoNLL2003 task):

- **Attack Rate (A-Rate)**: percentage of adversarial examples that can fool the model. An adversarial example is considered successful when at least one entity is incorrectly classified.

- **Modification Rate (Mod)**: percentage of modified tokens. Insert operations increase by one the modified tokens count (Li et al., 2021).

- **Δ Grammar Errors (ΔGErr)**: difference in the number of grammar errors between the adversarial example and its original counterpart, calculated with LanguageTool (Naber et al., 2003).

- **Textual similarity (Sim)**: cosine similarity between the adversarial example and the original input calculated via the Universal Sentence Encoder.

We also define the **Labels Score** (L-Score) metric as the percentage of incorrectly classified entities in a sample. All metrics defined above are averaged over the successful samples (with the exception of the attack rate). Table 1 lists the metrics for the original and attacked datasets.

4.2 Adversarial training

Table 1 shows that our victim model is vulnerable to adversarial attacks, which raises the question: is it possible to exploit attacks to make the model more robust while maintaining a reasonable performance on standard data? And what is the difference in model performance between models adversarially trained with word-level and character-level adversarial examples, both in normal conditions and when under attack?

To answer this question we trained a BERT$_{\text{base}}$ cased model, named NER$_{\text{small}}$, on $1/3$ of the CoNLL2003 dataset, equivalent to 5000 examples. A smaller dataset simulates a low-resource scenario

Attack	Acc	Recall	F1	A-Rate ↑	Mod ↓	L-Score ↑	Sim ↑	ΔGErr ↓
Bert-Attack	72%	88%	79%	44%	22%	20%	84%	0.26
CLARE	78%	81%	79%	37%	70%	56%	86%	0.33
DeepWordBug-II$_5$	86%	92%	89%	27%	18%	24%	86%	1.6
DeepWordBug-II$_{30}$	82%	93%	87%	30%	21%	23%	83%	3.05
DeepWordBug-I$_5$	48%	49%	49%	78%	24%	64%	77%	1.4
SCPN	90%	92%	91%	18%	66%	58%	59%	0.92
Original	98%	99%	98%					

Table 1: Comparison of attack strategies on the CoNLL2003 test set using the **untargeted strict** goal function. The metrics were calculated using `seqeval` (Nakayama, 2018). ↑ (↓) indicate whether the higher (or lower) the better from the attack perspective.

Model	Acc	Recall	F1	A-Rate ↑	Mod ↓	L-Score ↑	Sim ↑	ΔGErr ↓
NER$_{\text{small}}$ 500	73%	71%	72%	84%	26%	71%	73%	1.5
NER$_{\text{small}}$ 1000	78%	77%	77%	82%	27%	70%	73%	1.54
NER$_{\text{small}}$ 1500	77%	77%	77%	84%	26%	68%	73%	1.47
NER$_{\text{small}}$ 2000	79%	79%	79%	82%	28%	67%	72%	1.48
NER$_{\text{small}}$ (baseline)	52%	50%	51%	89%	24%	78%	75%	3.83

Table 2: Comparison of CoNLL2003 models against DeepWordBug-I$_5$, trained using a different amount of adversarial examples generated by DeepWordBug (specified next to the model) and the **untargeted** goal function. The attack had up to 45 seconds to successfully attack an input sample. ↑ (↓) indicate whether the higher (or lower) the better from the attack perspective.

and highlights the differences between the two adversarial training strategies. The model achieves a reasonable performance on the test set (Table 4, last row) but it can be fooled by both DeepWordBug-I$_5$ (Table 2) and BERT-Attack (Table 3). The adversarial data augmentation was done by attacking NER$_{\text{small}}$ using its own training set as the attack input. We respectively generated 2000 and 1000 adversarial examples for DeepWordBug-I$_5$ and BERT-Attack, which were then used to train robust models, whose performance on CoNLL2003 is listed in Tables 4 and 5.

4.2.1 Model evaluation

To evaluate the effectiveness of adversarial training we ran the same attacks against NER$_{\text{small}}$ and its robust counterparts, using the same CoNLL2003 subset used for evaluating attack strategies. Both attack strategies were allowed up to 512 model invocations. DeepWordBug-I$_5$ and BERT-Attack were respectively allowed up to 45 and 60 seconds to attack each sample.

5 Results and discussion

5.1 Adversarial attacks

Table 1 lists the after-attack metrics for the various attack strategies. By observing the metrics we can notice that DeepWordBug-I$_5$ is the most effective. Its success is most likely due to the fact that it can modify named entities. In fact, when named entities are preserved as in DeepWordBug-II$_5$, the attack rate drops to 27% and increasing the Levenshtein distance constraint to 30 has little effectiveness. Word-level attacks are less effective than unconstrained character-level attacks, but perform better than similarly constrained character-level attacks, decreasing a model's accuracy by up to 26% in the case of BERT-Attack. Even if less effective, word-level attack strategies may be useful for adversarial training since the generated samples are highly grammatical (introducing less than 0.5 grammar errors per sample), have a low percentage of modified words (except when insertions are used) and maintain a high sentence similarity: 84-86% for BERT-Attack and CLARE versus 77% for DeepWordBug-I$_5$. Some adversarial examples generated respectively by BERT-attack and CLARE can be seen in the appendix (Figures 8 and 9). Future work may attempt to apply word-level attacks also on the entities themselves, making sure to preserve the entity class. This would both speed up the adversarial examples generation (due to the higher sensitivity) and uncover examples highly likely to appear in the real world.

Model	Acc	Recall	F1	A-Rate ↑	Mod ↓	L-Score ↑	Sim ↑	ΔGErr ↓
$\text{NER}_{\text{small}}$ 500	94%	95%	94%	16%	19%	52%	89%	0.08
$\text{NER}_{\text{small}}$ 1000	94%	95%	94%	12%	19%	50%	89%	0.2
$\text{NER}_{\text{small}}$ (baseline)	88%	89%	89%	20%	18%	55%	88%	0.17

Table 3: Comparison of CoNLL2003 models against BERT-Attack, trained using a different amount of adversarial examples generated by BERT-Attack (specified next to the model) and the **untargeted** goal function. The attack had up to 60 seconds to successfully attack an input sample. ↑ (↓) indicate whether the higher (or lower) the better from the attack perspective.

5.2 Adversarial training

Tables 2 and 3 respectively summarize the attack metrics for DeepWordBug-I_5 and BERT-Attack. In line with the adversarial attacks results, DeepWordBug-I_5 obtains a largely better success than BERT-Attack, reducing the model's after-attack accuracy to 52%, where BERT-Attack only manages to reduce the accuracy to 88%.

Adversarial training grants a significant protection from both attacks: in the case of DeepWordBug-I_5 (Table 2) adding only 500 samples to the training set already increases the after attack accuracy by 21%, without affecting the test set metrics, causing at the same time an increase in the modification rate and a decrease in the similarity score. The improvement is statistically significant: a paired t-test with regards to the modification rate and the labels score respectively yields p-values of 0.0086 and 2.42e-09, confirming the added robustness of the adversarially trained model. The improvement is also visible in Figure 4, where the labels score distribution of the attacked dataset for the normal model is more skewed towards the right than its robust counterpart, showing a smaller attack success on individual samples for the robust model. Similarly, the modification rate distribution for the normal model is more skewed towards the left, thus more words need to be perturbed to fool the robust model. Adding more samples further improves the after attack scores at a small cost of the standard metrics (Table 4), but the improvement over the robust model with 500 samples is statistically significant only when 2000 adversarial examples are used, and only in regards to the labels score (p = 0.017).

Similarly, the robust models trained with BERT-attack have a performance similar to $\text{NER}_{\text{small}}$ on the test set, even improving the model's F1 score by 1% (Table 5). Using only 500 samples the after-attack accuracy increases by 6% and the attack-rate drops by 4%. Adding more samples further reduces the attack rate (Table 3). Using only 500 samples causes a significant improvement in the modification rate needed to break the model, yielding a p-value of 2.66e-05, but does not grant significant improvement in the labels score (p = 0.4). The latter improves significantly only when 1000 samples are used, where the p-value for the labels score is 0.011. These results are very encouraging, since the added robustness does not affect the test-set metrics and even improves it, suggesting that this attack method could be used for data augmentation in low-resource scenarios, a potential direction for future research. The difference in the number of samples needed to grant a significant robustness against DeepWordBug-I_5 and BERT-Attack may be explained by the initial effectiveness of the attack strategy: the former reduces the baseline accuracy to 52%, meanwhile the latter only reduces it to 88%.

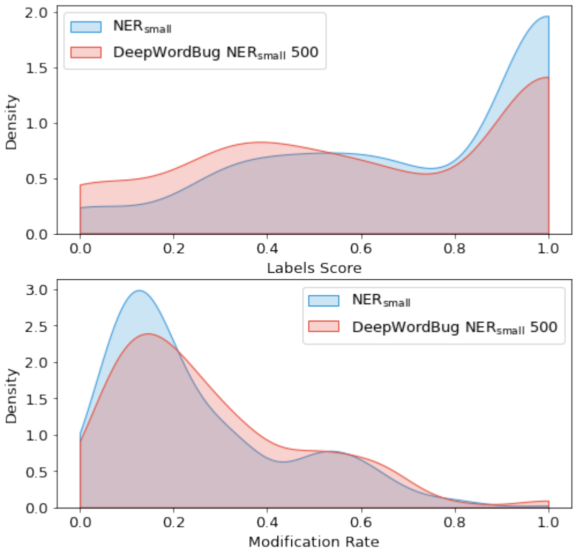

Figure 4: KDE plots for the labels score and modification rate distributions for $\text{NER}_{\text{small}}$ and its robust counterpart, when attacked with DeepWordBug-I_5. To be successful, the attack needs to alter more words for the robust model, and nonetheless achieves a lower labels score on average.

Examples	**Acc**	**Recall**	**F1**
500 examples	91%	90%	90%
1000 examples	90%	89%	90%
1500 examples	90%	90%	90%
2000 examples	90%	89%	90%
NER$_{\text{small}}$	91%	90%	90%

Table 4: CoNLL2003 test set metrics for the adversarially trained models against DeepWordBug. Adding adversarial examples slightly worsens the metrics due to overfitting.

Examples	**Acc**	**Recall**	**F1**
500 examples	91%	90%	91%
1000 examples	91%	90%	91%
NER$_{\text{small}}$	91%	90%	90%

Table 5: CoNLL2003 test set metrics for the adversarially trained models against BERT-Attack. Adversarial examples slightly improve the metrics, potentially covering blind spots in the training set.

6 Conclusion

In this paper we showed that NER models are vulnerable to adversarial attacks at the character, word and sentence level. When allowed to alter named entities, DeepWordBug is the most effective, but it produces highly ungrammatical samples (appendix Figures 6, 7). Thus, character-level attacks are not recommended for adversarial training or data augmentation since the produced samples are unlikely to appear in a real-world setting. Word-level attacks instead produce more fluent adversarial examples (appendix Figures 8, 9), which can be used both for adversarial training and for data augmentation. Finally, with regards to sentence-level attacks, this paper finds that they often produce low-quality samples for this particular dataset (appendix Figure 5). This is probably due to the fact that SCPN paraphrases are generated following a small set of target syntactic forms, which are incompatible with CoNLL2003. Further research in this direction is recommended, as paraphrasing methods produce a richer variety of samples and may reveal weaknesses in a model which cannot be discovered by character-level or word-level attacks.

To help NLP practitioners evaluate and improve their models' robustness and to foster research on adversarial attacks in token classification (and named entity recognition) we release `SeqAttack`[1], a Python library for conducting adversarial attacks against token classification models. The library is accompanied by a web application[2] to inspect the generated adversarial examples and cherry pick them for adversarial training.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Vladimir Araujo, Andres Carvallo, Carlos Aspillaga, and Denis Parra. 2020. On adversarial examples for biomedical nlp tasks. *arXiv preprint arXiv:2004.11157*.

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. 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.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks.

[1] `https://github.com/WalterSimoncini/SeqAttack`

[2] Application available at `https://ner-attack.ashita.nl/`

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.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. 2017. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2664–2669, Copenhagen, Denmark. Association for Computational Linguistics.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019a. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. 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. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Daniel Naber et al. 2003. A rule-based style and grammar checker.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Desislava Petkova and W Bruce Croft. 2007. Proximity-based document representation for named entity retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 731–740.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

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.

Jiuniu Wang, Wenjia Xu, Xingyu Fu, Guangluan Xu, and Yirong Wu. 2020. Astral: adversarial trained lstm-cnn for named entity recognition. *Knowledge-Based Systems*, 197:105842.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing:*

System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. 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 363–371, Online. Association for Computational Linguistics.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

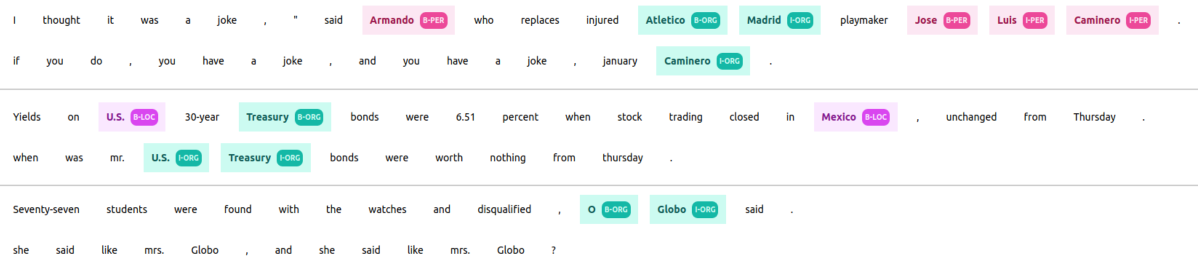
I thought it was a joke , " said Armando B-PER who replaces injured Atletico B-ORG Madrid I-ORG playmaker Jose B-PER Luis I-PER Caminero I-PER .
if you do , you have a joke , and you have a joke , january Caminero B-ORG .
Yields on U.S. B-LOC 30-year Treasury B-ORG bonds were 6.51 percent when stock trading closed in Mexico B-LOC , unchanged from Thursday .
when was mr. U.S. I-ORG Treasury I-ORG bonds were worth nothing from thursday .
Seventy-seven students were found with the watches and disqualified , O B-ORG Globo I-ORG said .
she said like mrs. Globo , and she said like mrs. Globo ?

Figure 5: Adversarial examples generated by SCPN, when attacking a BERT-based model trained on CoNLL2003. For each pair the top row represents the original sample and the bottom row its attacked counterpart. The modified words are underlined in red.

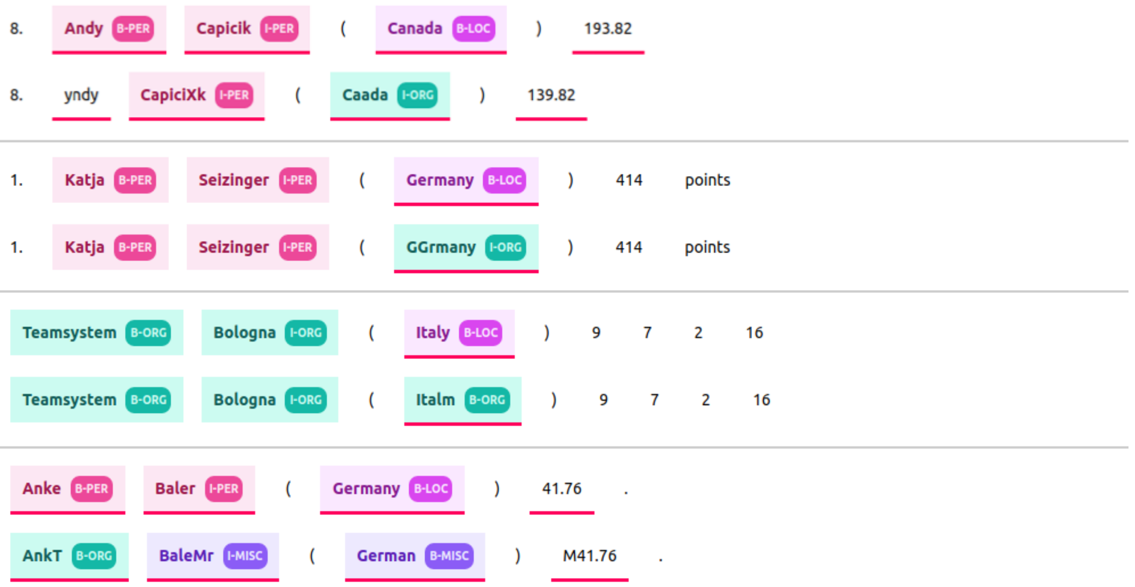
8. Andy B-PER Capicik I-PER (Canada B-LOC) 193.82
8. yndy CapiciXk I-PER (Caada I-ORG) 139.82
1. Katja B-PER Seizinger I-PER (Germany B-LOC) 414 points
1. Katja B-PER Seizinger I-PER (GGrmany I-ORG) 414 points
Teamsystem B-ORG Bologna I-ORG (Italy B-LOC) 9 7 2 16
Teamsystem B-ORG Bologna I-ORG (Italm B-ORG) 9 7 2 16
Anke B-PER Baler I-PER (Germany B-LOC) 41.76 .
AnkT B-ORG BaleMr I-MISC (German B-MISC) M41.76 .

Figure 6: Adversarial examples generated by DeepWordBug-I, when attacking a BERT-based model trained on CoNLL2003. For each pair the top row represents the original sample and the bottom row its attacked counterpart. The modified words are underlined in red.

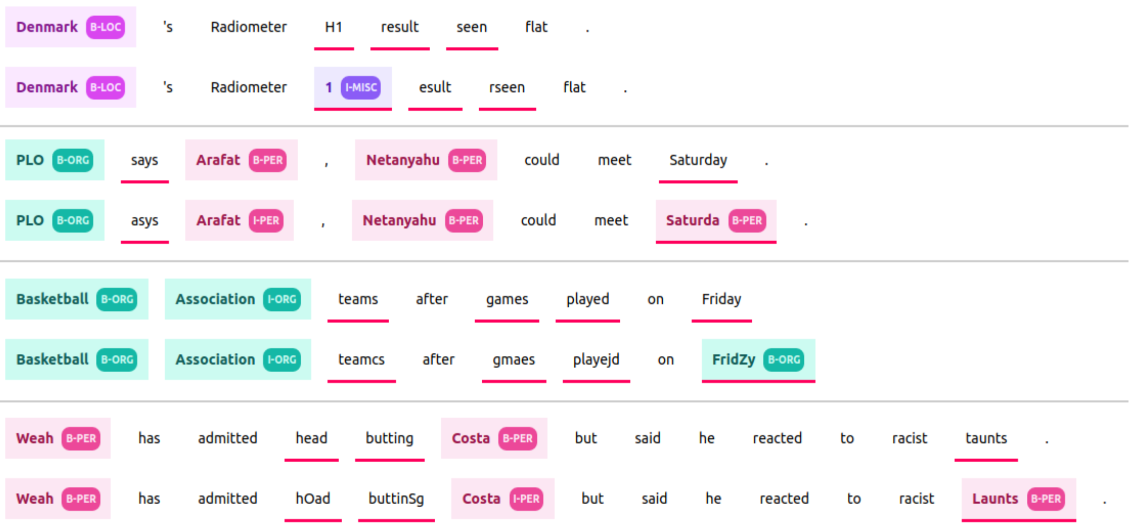
Denmark B-LOC 's Radiometer H1 result seen flat .
Denmark B-LOC 's Radiometer 1 I-MISC esult rseen flat .
PLO B-ORG says Arafat B-PER , Netanyahu B-PER could meet Saturday .
PLO B-ORG asys Arafat I-PER , Netanyahu B-PER could meet Saturda B-PER .
Basketball B-ORG Association I-ORG teams after games played on Friday
Basketball B-ORG Association I-ORG teamcs after gmaes playejd on FridZy B-ORG
Weah B-PER has admitted head butting Costa B-PER but said he reacted to racist taunts .
Weah B-PER has admitted hOad buttinSg Costa I-PER but said he reacted to racist Launts B-PER .

Figure 7: Adversarial examples generated by DeepWordBug-II, when attacking a BERT-based model trained on CoNLL2003. For each pair the top row represents the original sample and the bottom row its attacked counterpart. The modified words are underlined in red.

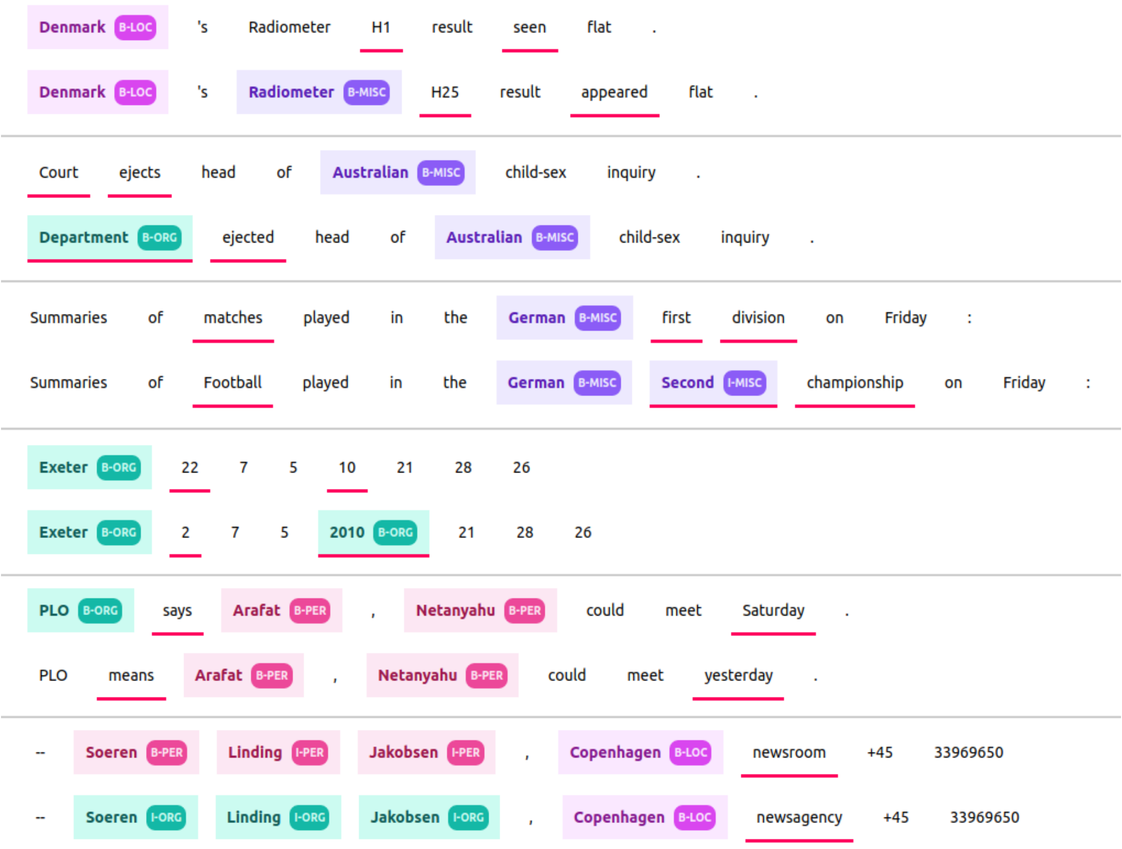
Denmark B-LOC 's Radiometer H1 result seen flat .
Denmark B-LOC 's Radiometer B-MISC H25 result appeared flat .
Court ejects head of Australian B-MISC child-sex inquiry .
Department B-ORG ejected head of Australian B-MISC child-sex inquiry .
Summaries of matches played in the German B-MISC first division on Friday :
Summaries of Football played in the German B-MISC Second I-MISC championship on Friday :
Exeter B-ORG 22 7 5 10 21 28 26
Exeter B-ORG 2 7 5 2010 B-ORG 21 28 26
PLO B-ORG says Arafat B-PER , Netanyahu B-PER could meet Saturday .
PLO means Arafat B-PER , Netanyahu B-PER could meet yesterday .
-- Soeren B-PER Linding I-PER Jakobsen I-PER , Copenhagen B-LOC newsroom +45 33969650
-- Soeren I-ORG Linding I-ORG Jakobsen I-ORG , Copenhagen B-LOC newsagency +45 33969650

Figure 8: Adversarial examples generated by BERT-Attack, when attacking a BERT-based model trained on CoNLL2003. For each pair the top row represents the original sample and the bottom row its attacked counterpart. The modified words are underlined in red.

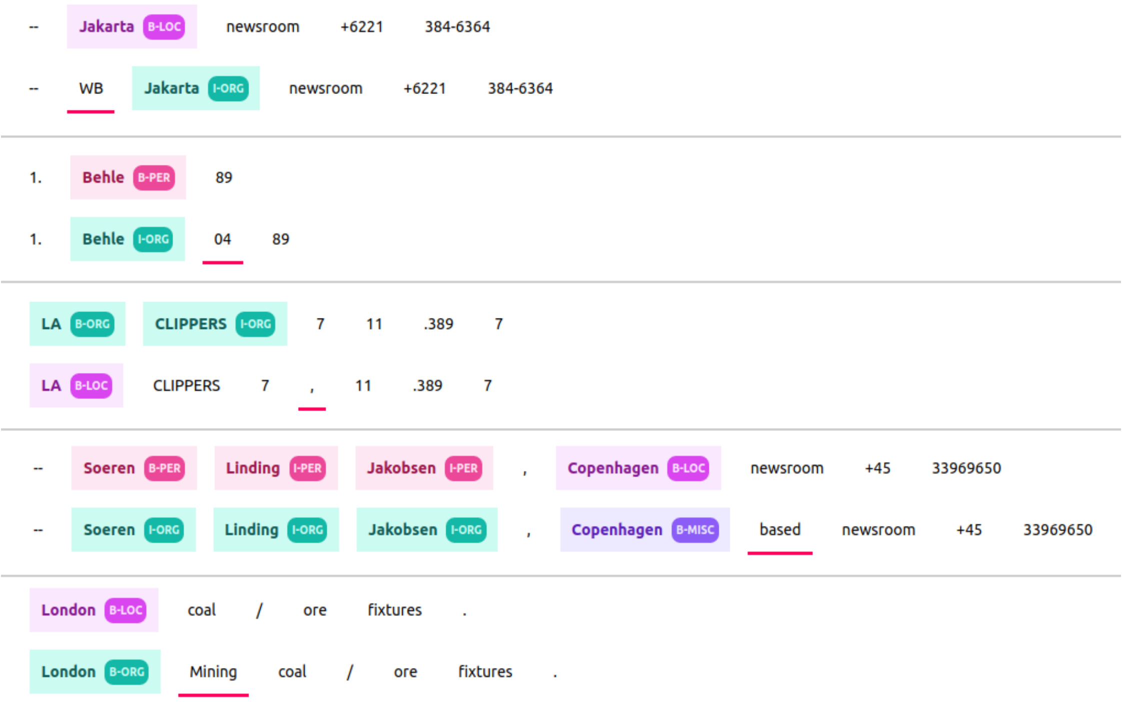
-- Jakarta B-LOC newsroom +6221 384-6364
-- WB Jakarta I-ORG newsroom +6221 384-6364
1. Behle B-PER 89
1. Behle I-ORG 04 89
LA B-ORG CLIPPERS I-ORG 7 11 .389 7
LA B-LOC CLIPPERS 7 , 11 .389 7
-- Soeren B-PER Linding I-PER Jakobsen I-PER , Copenhagen B-LOC newsroom +45 33969650
-- Soeren I-ORG Linding I-ORG Jakobsen I-ORG , Copenhagen B-MISC based newsroom +45 33969650
London B-LOC coal / ore fixtures .
London B-ORG Mining coal / ore fixtures .

Figure 9: Adversarial examples generated by CLARE, when attacking a BERT-based model trained on CoNLL2003. For each pair the top row represents the original sample and the bottom row its attacked counterpart. The modified words are underlined in red.