Data Augmentation for Robust Character Detection in Fantasy Novels

Arthur Amalvy, Vincent Labatut, Richard Dufour

Data Augmentation for Robust Character Detection in Fantasy Novels

Arthur Amalvy[1], Vincent Labatut[1] and Richard Dufour[2]

[1]Laboratoire d'Informatique d'Avignon (LIA), Avignon University, France
[2]Laboratoire des Sciences du Numérique de Nantes (LS2N), Nantes University, France

Abstract

Named Entity Recognition (NER) is a low-level task often used as a foundation for solving higher level NLP problems. In the context of character detection in novels, NER false negatives can be an issue as they possibly imply missing certain characters or relationships completely. In this article, we demonstrate that applying a straightforward data augmentation technique allows training a model achieving higher recall, at the cost of a certain amount of precision regarding ambiguous entities. We show that this decrease in precision can be mitigated by giving the model more local context, which resolves some of the ambiguities.

Keywords

data augmentation, named entity recognition, character detection

1. Introduction

The *Character Detection* task is concerned with detecting which characters appear in a text, and where. We decompose it in two subtasks:

- Named Entity Recognition (NER), whose goal is to find character occurrences in a text.
- Named Entity Disambiguation (NED), which maps character occurrences to their respective normalized character form.

The result of the character detection task can be used to solve other higher-level problems. Since its output can affect the performance of other tasks relying on it, such as character network extraction [1], its degree of success is of importance.

Dekker et al. [2] perform a study where they evaluate several NER models for the purpose of extracting character networks from literary texts. According to their results, performance varies greatly across novels, ranging from great to pretty low. These authors find that quite a few of the false negatives produced by the evaluated models stem from two specific types of names: word names (names that are also regular words, such as `Valor` or `Mercy`) and apostrophed names (such as `Rand al'Thor`). Replacing them with regular names allows for better detection, showing that the issue regarding these names is linked to their form and not to their surrounding context.

We suspect that these shortcomings come from the datasets used to train the NER systems assessed by Dekker et al. [2]. Indeed, NER systems are usually trained on the main publicly available datasets (such as Ontonotes [3], CoNLL-2003 [4] or WikiGold [5]), which come from a few domains (news, web...). Among these mainstream datasets, none contain texts of the literary domain[1], and annotating new datasets is costly. This means applying off-the-shelf NER systems to literary texts suffers from these systems integrating knowledge specific only to their training data.

More precisely, there may be a mismatch in style between typical person names from the training dataset and from literary texts. This is particularly the case in the fantasy genre, where character names can have very unusual styles depending on the setting.

In order to try and fix this issue, we showcase the application of a specific data augmentation technique, *mention replacement* [7], to inject new character names in the training dataset. Using this technique, we demonstrate performance improvements for the NER task on a dataset of texts from the fantasy genre. We release our code and data, that can be used to reproduce our results, under a free license[2].

The rest of this article is organized as follows: first, Section 2 provides a few examples of related data augmentation works. Then, in Section 3, we present mention replacement and the experiments we perform to assess its ability to resolve the name mismatch issue we highlighted. We discuss the results of these experiments in Section 4, and perform further analysis to shed light on the precision decrease we observe when using mention

COMHUM 2022
arthur.amalvy@univ-avignon.fr (A. Amalvy);
vincent.labatut@univ-avignon.fr (V. Labatut);
richard.dufour@univ-nantes.fr (R. Dufour)
0000-0002-0877-7063 (A. Amalvy); 0000-0002-2619-2835 (V. Labatut); 0000-0003-1203-9108 (R. Dufour)

[1]While the Litbank dataset [6] specifically contains only literary texts, its annotations concern *nested* NER, an arguably harder task.
[2]https://github.com/CompNet/ddaugNER

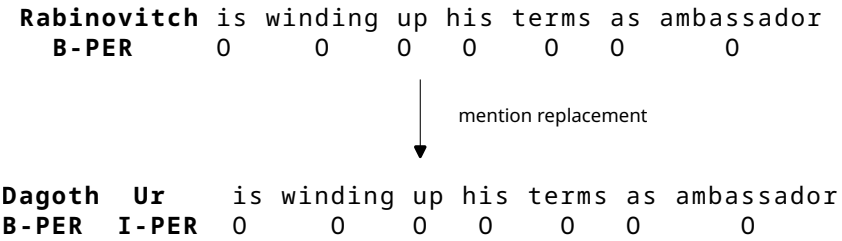

Figure 1: Example of mention replacement. Note that in this case, the sentence labels have to be slightly modified to account for the replacement entity higher number of tokens.

replacement. Finally, we conclude with some perspectives regarding NER performance in literary texts.

2. Data Augmentation and Cross-Domain NER

Generally speaking, data augmentation is the generation of new synthetic training examples. Data augmentation techniques can be used to address data scarcity problems, or to increase the diversity of the training dataset to reduce overfitting. While these techniques are ubiquitous in image processing [8], they are less commonly explored in natural language processing [9], and even less for NER [7].

A few techniques have been used to try and fix the domain discrepancy between training and testing datasets. Ding et al. [10] train a language model to generate new examples by directly including NER tags in the generated text. Chen et al. [11] train a neural architecture to transform examples from the training domain to examples closer to the test domain. Very recently, Yang et al. [12] propose *FactMix*, a two-step data augmentation process performing mention replacement followed by masked token replacement. However, as far as we know, previous works are not specifically concerned with literary texts or the mismatch in person names between domains.

3. Method

In this section, we first describe mention replacement and how we apply it to texts of the fantasy genre. We then introduce the datasets we use to assess the performance of this data augmentation technique. Finally, we describe the specific setting of the experiment we carry to evaluate its performance, including the model we used and its training parameters.

3.1. Mention Replacement

Dai and Adel [7] survey a few simple data augmentation schemes for NER, including mention replacement. It consists in generating new examples by replacing tokens from an entity mention found in the training dataset by tokens from another entity of the same type. Figure 2 shows an example of this process. Based on the same principle, we propose to randomly replace training entities with ones from a list of fantasy names to increase the training dataset coverage of this type of names. If the same entity mention is present multiple times in a training example, we replace every occurrence with the same fantasy mention when performing augmentation to avoid inconsistencies. We hope that injecting typical fantasy names into the training dataset using mention replacement will allow the model to better detect them.

To perform mention replacement, we compose a list of replacement entities that do not come directly from the evaluation dataset. We scrap the entirety of the names from *The Elder Scrolls* series of video games that are mentioned on the *Unofficial Elder Scrolls Pages*, a wiki dedicated to *The Elder Scrolls* universe[3]. In total, we retrieve 22,748 first names, 4,879 last names, 647 suffixes and 37 prefixes. We combine these to form new mentions: when performing mention replacement, we randomly compose a new mention by first sampling from a set of valid forms (`[first name]+[last name]`, `[prefix]+[first name]+[last name]`, ...). We weight this sampling by a rough approximation of each form's frequency. When a form has been selected, we uniformly sample a name part for each of the form's elements.

Since we are interested in knowing the impact of the number of generated examples, we compare the performance of models trained with different augmentation rates. We define the *augmentation rate* as the ratio of examples generated over the dataset size, so an augmentation rate of 1 would mean generating as many examples as there are examples in the dataset.

3.2. Datasets

Training Dataset We train our model on a modified version of CoNLL-2003 [4], which is one of the best known and used NER dataset. CoNLL-2003 is composed

[3] https://en.uesp.net/wiki/Main_Page

of 14,041 sentences from news articles. Our modifications consist in including honorifics as part of entities to be consistent with our evaluation dataset (see below). The training dataset contains annotations for four classes: persons (PER), organizations (ORG), locations (LOC) and miscellaneous (MISC). Despite restricting ourselves to character detection (PER class), we keep annotations for every class since we observe that training the model with all classes increase the performance of NER.

Evaluation dataset We use a corrected version of the subset of fantasy novels from the dataset of Dekker et al. [2], where only persons entities (PER) are annotated. We found the dataset had to be corrected because we noticed a number of encoding, tokenization and annotation issues. We fixed the encoding and tokenization issues manually. In order to consistently correct annotation errors, we designed an annotation guide and applied a semi-automated correction process. It consists of 3 steps:

1. We apply a number of simple heuristics to identify obvious errors:
   - When a span is not annotated as a person occurrence, but it appears in the list of character names for the book, it might be a false negative.
   - When a span is annotated as a person occurrence, but it does not appear in this list, it might a false positive.
   - When a span is annotated as a person occurrence, but the first letter of all of its tokens is not capitalized, it might be a false positive.

   Each time one of these heuristics recommends an annotation change, we manually check if it is correct before accepting it.
2. We then consider the differences between the existing annotations and the predictions of a BERT model [13] fine-tuned for the NER task, and manually reconcile them.
3. Finally, we manually correct the few remaining errors that we could detect.

The dataset consists in the first chapter of 17 novels. Together, these chapters are composed of 5,518 sentences and contain 360 unique person names. Inspired by Taillé et al. [14], we report in Table 1 the name overlap between our evaluation dataset and:

- The train portion of the CoNLL-2003 [4];
- Our list of names from *The Elder Scrolls*, used for mention replacement.

Name Set	Exact Match	Partial Match	Unseen
CoNLL-2003 (train)	9.62%	76.97%	13.41%
The Elder Scrolls	12.25%	76.38%	11.37%

Table 1
Overlap of different name sets with the set of character names from our evaluation dataset. The "exact match" column reports the proportion of PER labeled tokens from our evaluation dataset that are also in the studied name set. The "partial match" column is the proportion of PER labeled tokens containing a wordpiece that exists in the studied name set. Finally, the "unseen" column indicates the rest of PER labeled tokens from the evaluation dataset.

Vin	Camon	Bilbo	Pug	Arlen
Galladon	Bran	Selia	Bug	Cob
Raoden	Gandalf	Szeth	Weasel	Chivalry
Logen	Theron	Frodo	Jory	Jezrien

Table 2
Top 20 character names that were better detected by a model trained with data augmentation (10% augmentation rate) but not by a model trained with the original dataset only.

3.3. Experiment

We fine-tune a BERT model [13] with an added classification head on our training dataset, and report its performance on our evaluation dataset depending on the augmentation rate. In order to obtain stable results, we report the mean of the results from 10 fine-tuning runs for each augmentation rate that we test. Models without augmentations are trained for two full epochs, while we adjust the number of training steps for model with augmentation so that they learn on the same number of examples as models without augmentation. We use a learning rate of $2 \cdot 10^{-5}$ [13], and we initialize each model with the `bert-base-cased` checkpoint of the *transformers* library [15]. At inference time, each model predicts NER tags separately for each sentence, and we do not provide it with any additional context. Since the evaluation dataset only contains annotations for the PER class, we simply discard predictions made for other classes.

4. Results

Results can be seen in Figure 2. We report performance according to the CoNLL-2003 guidelines [4] and using the Python `seqeval` library [16]. We observe a notable increase in recall using augmentation, while precision decreases. The recall increase can be attributed to the model picking up on more unusual character names, as Table 2 shows.

To try to understand the decrease in precision when using data augmentation, we perform some additional

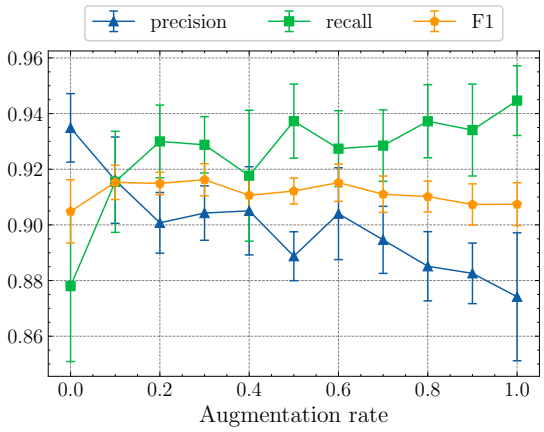

Figure 2: Precision, recall and F1-score against augmentation rate. Error bars represent the 95% confidence interval.

experiments. Based on our manual investigation of the false positives, we formulate two hypotheses that we want to test:

h_1 Adding more examples with PER entities modifies the class distribution in the dataset, progressively leading to a situation of class imbalance where these entities are way more frequent than those of the other classes (LOC, ORG and MISC). As we observe that removing other entity classes from the training dataset is detrimental to PER detection performance (possibly because having more classes means the model is better at distinguishing between them), we suppose that class imbalance can be detrimental as well.

h_2 Increasing name variety in the training set encourages the model to try and make PER predictions for tokens whose classes are ambiguous given a specific sentence. We deem a token "ambiguous" when the context given to the model does not suffice to infer its correct class (for example, when it is not possible to distinguish between a PER and a LOC entity using only an input sentence).

4.1. Class Imbalance

To check if the decrease in precision is due to class imbalance, we test two different methods to inject names in the training dataset that should not suffer from the class imbalance problem:

- `Upsample and balance`: After adding newly generated examples to the dataset, we copy some of the original examples and add them to the dataset to restore the original class distribution.
- `Replace`: Instead of *adding* newly generated examples to the dataset, we *substitute* them for the original training samples.

Figure 3 shows the performance of a model trained on a dataset modified with the above augmentation methods. The *upsample and balance* strategy still increases recall, but does not fix the precision issue. Meanwhile, the `replace` strategy still makes precision decrease, and also generally decreases performance for an augmentation rate greater than 20%.

As none of these two methods are able to fix the precision issue, we conclude that class imbalance is not a plausible cause of precision decreasing.

4.2. Name Variety and Ambiguous Occurrences

In order to check h_2, we analyze the difference in false positives between a model trained with augmentation and a model trained without it. As errors can vary between runs, we perform three training runs for each model and keep the false positives that are consistent between the runs. We then observe the set of false positives of the model trained with augmentation that are not present in the set of false positives of the model trained without augmentation. This allows us to analyze errors that are specific to our augmentation scheme.

We manually find that for 62% (58/93) of these false positives, the model attributes a PER label to some tokens even though the context alone is not sufficient to know the correct label of the entity, such as in the following example from *Elantris*:

```
Raoden stood, and as he did, his eyes fell on
Elantris again.
```

Here, the model correctly predicts that `Elantris` is a named entity, but gives it the class PER while `Elantris` is actually a city (which corresponds to the LOC class). Of course, predicting the correct class of `Elantris` in this example would only be a matter of luck even for a perfect model, as the given context alone is not sufficient.

Overall, this result shows a bias towards the PER class, that was introduced by increasing the variety of possible names at training time.

To try and mitigate this effect, we supply the model with broader context, in hope that this context can help resolving some ambiguities. In our previous example, the next sentence gives enough context to predict the correct entity type for `Elantris`:

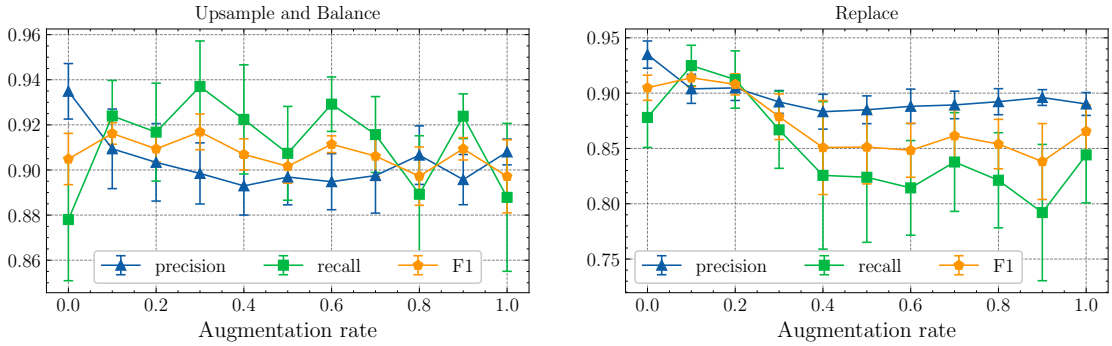

Figure 3: Precision, recall and F1-score against augmentation rate for different augmentation methods. Error bars represent the 95% confidence interval.

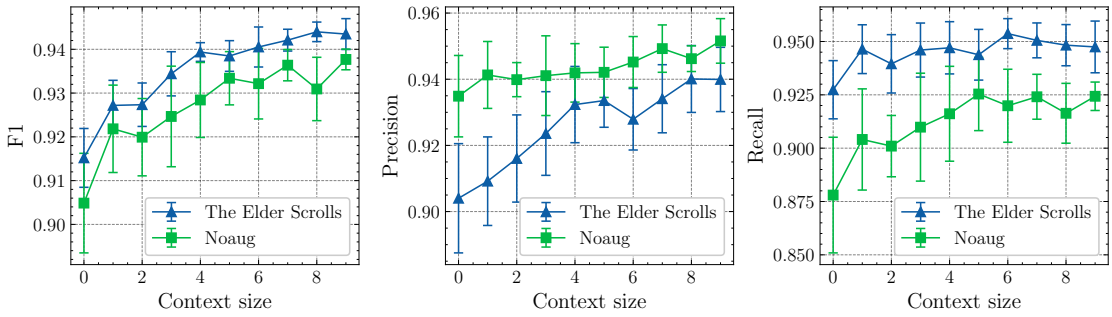

Figure 4: F1-score, precision and recall against different context sizes for a model trained without augmentation (`Noaug`) and a model trained with augmentation (`The Elder Scrolls`) (60% augmentation ratio). Error bars representent the 95% confidence interval.

```
Raoden stood, and as he did, his eyes fell on
Elantris again. Resting in the great city's
shadow, Kae seemed like an insignificant
village by comparison.
```

We define a context size of n as giving the model the n sentences that *precede* the considered sentence, as well as the n sentences that *follow* it.

Figure 4 shows the effect of giving more context to models trained with augmentation (`The Elder Scrolls`) or without it (`Noaug`). As can be seen, context is beneficial to both models, steadily increasing F1-score. Precision increases a lot for the model trained with augmentation, as previously ambiguous entities are now correctly labeled. Using our previously described method for error analysis, we find that only 48% (30/63) of the entities found in false positives are deemed ambiguous for a context size of 1. Meanwhile, the recall augmentation for the `Noaug` model can be attributed to it predicting `PER` entities in previously ambiguous cases. While adding more context results in greater recall for the `Noaug` model, it still never surpasses the recall of the `The Elder Scrolls` model.

5. Conclusion

We demonstrated the usage of a simple data augmentation technique to better detect fantasy names when performing named entity recognition. While this technique greatly increases model recall, precision decreases as the model sometimes does not have enough information to predict the correct class of some entities. We showed that this issue can be mitigated by giving the model more local information. However, for the same context size, our technique always result in greater recall but lower precision. In the context of character detection, we argue that this trade is beneficial: while false positives can be filtered after performing NER (automatically or manually), false negatives are not easily recovered. We also note that increasing NER context has a positive impact on recall in general.

Further work might investigate the role of context in NER. While we showed that it can be used to disambiguate some mentions, we were only interested in local context. Nevertheless, necessary disambiguation infor-

mation is not always present in the local context, And might exist in other places for each novel.

References

[1] V. Labatut, X. Bost, Extraction and analysis of fictional character networks : A survey, ACM Computing Surveys 52 (2019) 89. doi:10.1145/3344548.

[2] N. Dekker, T. Kuhn, M. van Erp, Evaluating named entity recognition tools for extracting social networks from novels, PeerJ Computer Science 5 (2019) e189. doi:10.7717/peerj-cs.189.

[3] R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, N. Xue, OntoNotes: A large training corpus for enhanced processing, 2011. URL: https://www.cs.cmu.edu/~hovy/papers/09OntoNotes-GALEbook.pdf.

[4] E. F. Tjong Kim Sang, F. De Meulder, Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, in: 7th Conference on Natural Language Learning, 2003, pp. 142–147. URL: https://aclanthology.org/W03-0419.

[5] D. Balasuriya, N. Ringland, J. Nothman, T. Murphy, J. R. Curran, Named entity recognition in Wikipedia, in: Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources, 2009, pp. 10–18. URL: https://aclanthology.org/W09-3302.

[6] D. Bamman, S. Popat, S. Shen, An annotated dataset of literary entities, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1, 2019, pp. 2138–2144. doi:10.18653/v1/N19-1220.

[7] X. Dai, H. Adel, An analysis of simple data augmentation for named entity recognition, in: International Conference on Computational Linguistics, 2020, pp. 3861–3867. doi:10.18653/v1/2020.coling-main.343.

[8] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (2019) 60. doi:10.1186/s40537-019-0197-0.

[9] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for NLP, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 968–988. doi:10.18653/v1/2021.findings-acl.84.

[10] B. Ding, L. Liu, L. Bing, C. Kruengkrai, T. H. Nguyen, S. Joty, L. Si, C. Miao, DAGA: Data augmentation with a generation approach for low-resource tagging tasks, in: Conference on Empirical Methods in Natural Language Processing, 2020, pp. 6045–6057. doi:10.18653/v1/2020.emnlp-main.488.

[11] S. Chen, G. Aguilar, L. Neves, T. Solorio, Data augmentation for cross-domain named entity recognition, in: Conference on Empirical Methods in Natural Language Processing, 2021, pp. 5346–5356. doi:10.18653/v1/2021.emnlp-main.434.

[12] L. Yang, L. Yuan, L. Cui, W. Gao, Y. Zhang, Factmix: Using a few labeled in-domain examples to generalize to cross-domain named entity recognition, arXiv cs.CL (2022) 2208.11464. doi:10.48550/ARXIV.2208.11464.

[13] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[14] B. Taillé, V. Guigue, P. Gallinari, Contextualized embeddings in named-entity recognition: An empirical study on generalization, in: Advances in Information Retrieval, 2020, pp. 383–391. doi:10.1007/978-3-030-45442-5_48.

[15] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[16] H. Nakayama, seqeval: A python framework for sequence labeling evaluation, 2018. URL: https://github.com/chakki-works/seqeval.