The Role of Global and Local Context in Named Entity Recognition

Arthur Amalvy
Laboratoire Informatique d'Avignon
arthur.amalvy@univ-avignon.fr

Vincent Labatut*
Laboratoire Informatique d'Avignon
vincent.labatut@univ-avignon.fr

Richard Dufour*
Laboratoire des Sciences du Numérique de Nantes
richard.dufour@univ-nantes.fr

Abstract

Pre-trained transformer-based models have recently shown great performance when applied to Named Entity Recognition (NER). As the complexity of their self-attention mechanism prevents them from processing long documents at once, these models are usually applied in a sequential fashion. Such an approach unfortunately only incorporates local context and prevents leveraging global document context in long documents such as novels, which might hinder performance. In this article, we explore the impact of global document context, and its relationships with local context. We find that correctly retrieving global document context has a greater impact on performance than only leveraging local context, prompting for further research on how to better retrieve that context.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP), and is often used as a building block for solving higher-level tasks. Recently, pre-trained transformer-based models such as BERT (Devlin et al., 2019) or LUKE (Yamada et al., 2020) showed great NER performance and have been able to push the state of the art further.

These models, however, have a relatively short range because of the quadratic complexity of self-attention in the number of input tokens: as an example, BERT (Devlin et al., 2019) can only process spans of up to 512 tokens. For longer documents, texts are usually processed sequentially using a rolling window. Depending on the document, this local window may not always include all the context needed to perform inference, which may be present at the global document level. This leads to prediction errors (Stanislawek et al., 2019): In NER, this often occurs when the type of an entity cannot be inferred from the local context. For instance, in the following sentence from the fantasy novel *Elantris*, one cannot decide if the entity `Elantris` is a person (`PER`) or a location (`LOC`) without prior knowledge:

> *"Raoden stood, and as he did, his eyes fell on Elantris again."*

In the novel, this prior knowledge comes from the fact that a human reader can recall previous mentions of `Elantris`, even at a very long range. A sequentially applied vanilla transformer-based model, however, might make an error without a *neighboring* sentence clearly establishing the status of `Elantris` as a city.

While some works propose to retrieve external knowledge to disambiguate entities (Zhang et al., 2022; Wang et al., 2021), external resources are not always available. Furthermore, external retrieval might be more costly or less relevant than performing document-level context retrieval, provided the document contains the needed information, which depends on the type of document.

Therefore, we wish to explore the relevance of document-level context when performing NER. We place ourselves at the sentence level, and we distinguish and study two types of contexts:

- *local context*, consisting of surrounding sentences. This type of context can be used directly by vanilla transformer-based models, as their range lies beyond the simple sentence. Fully using surrounding context as in Devlin et al. (2019) is, however, computationally expensive.
- *global context*, consisting of all sentences available at the document level. To enhance NER prediction at the sentence level, we retrieve a few of these sentences and provide them as context for the model.

We seek to answer the following question: is local context sufficient when solving the NER task,

*These authors contributed equally.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 714–722

or would the model obtain better performance by retrieving global document context?

To answer this question, we conduct experiments on a literary NER dataset we improved from its original version (Dekker et al., 2019). We release the annotation process, data and code necessary to reproduce these experiments under a free license[1].

2 Related Work

2.1 Sparse Transformers

Since the range problem of vanilla transformer-based models is due to the quadratic complexity of self-attention in the number of input tokens, several works on *sparse transformers* proposed alternative attention mechanisms in hope of reducing this complexity (Zaheer et al., 2020; Wang et al., 2020; Kitaev et al., 2020; Tay et al., 2020b,a; Beltagy et al., 2020; Choromanski et al., 2020; Katharopoulos et al., 2020; Child et al., 2019). While reducing self-attention complexity improves the effective range of transformers, these models still have issues processing very long documents (Tay et al., 2020c).

2.2 Context retrieval

Context retrieval in general has been widely leveraged for other NLP tasks, such as semantic parsing (Guo et al., 2019), question answering (Ding et al., 2020), event detection (Pouran Ben Veyseh et al., 2021), or machine translation (Xu et al., 2020).

In NER, context retrieval has mainly been used in an external fashion, for example by leveraging names lists and gazetteers (Seyler et al., 2018; Liu et al., 2019), knowledge bases (Luo et al., 2015) or search engines (Wang et al., 2021; Zhang et al., 2022). Meanwhile, we are interested in document-level context retrieval, which is comparatively seldom explored. While Luoma and Pyysalo (2020) study document-level context, their study is restricted to neighboring sentences, i.e. local context.

3 Method and Experiments

3.1 Retrieval Heuristics

We wish to understand the role of both *local* and *global* contexts for the NER task. We split all documents in our dataset (described in Section 3.3) into sentences. We evaluate both local and global simple heuristics of sentence retrieval in terms of NER performance impact. We study the following *local* heuristics:

- `before`: Retrieves the closest k sentences at the left of the input sentence.
- `after`: Same as `before`, but at the right of the input sentence.
- `surrounding`: Retrieves the closest $\frac{k}{2}$ sentences on both sides of the input sentence.

And the following *global* heuristics:

- `random`: Randomly retrieves a sentence from the whole document.
- `samenoun`: Randomly retrieves a sentence from the set of all sentences that have at least one common noun with the input sentence[2]. Intuitively, this heuristic will return sentences that contain entities of the input sentence, allowing for possible disambiguation. We use the NLTK library (Bird et al., 2009) to identify nouns.
- `bm25`: Retrieves sentences that are similar to the input sentences according to BM25 (Robertson, 1994). Retrieving similar sentences has already been found to increase NER performance (Zhang et al., 2022; Wang et al., 2021).

It has to be noted that global heuristics can sometimes retrieve local context, as they are not restricted in which sentences they can retrieve at the document level. For all configurations, we concatenate the retrieved sentences to the input. During this concatenation step, we preserve the global order between sentences in the document.

3.2 Oracles

For each heuristic mentioned in Section 3.1, we also experiment with an *oracle* version. The oracle version retrieves 16 sentences from the document using the underlying retrieval heuristic, and retain only those that enhance the NER predictions the most. We measure this enhancement by counting the difference in numbers of NER BIO tags errors made with and without the context. In essence, the oracle setup simulates a perfect re-ranker model, and allows us to study the maximum performance of such an approach.

[1] https://github.com/CompNet/conivel/tree/ACL2023

[2] If the set of sentences with a common noun is empty, the `samenoun` heuristic does not retrieve any sentence.

3.3 Dataset

To evaluate our heuristics, we use a corrected and improved version of the literary dataset of Dekker et al. (2019). This dataset is comprised of the first chapter of 40 novels in English, which we consider long enough for our experiments.

Dataset corrections The original dataset suffers mainly from annotation issues. To fix them, we design an annotation guide inspired by CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and apply it consistently using a semi-automated process:

1. We apply a set of simple rules to identify obvious errors[3] (for example, non capitalized entities annotated as PER are often false positives). Depending on the estimated performance of each rule, we manually reviewed its choices before application.

2. We manually review each difference between the predictions of a BERT (Devlin et al., 2019) model finetuned on a slightly modified version of the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003)[4] and the existing annotations.

3. We manually correct the remaining errors.

Further annotations The original dataset only consists of PER entities. We go further and annotate LOC and ORG entities. The final dataset contains 4476 PER entities, 886 LOC entities and 201 ORG entities.

3.4 NER Training

For all experiments, we use a pretrained $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019) model, consisting in 110 million parameters, followed by a classification head at the token level to perform NER. We finetune BERT for 2 epochs with a learning rate of $2 \cdot 10^{-5}$ using the `huggingface transformers` library (Wolf et al., 2020), starting from the `bert-base-cased` checkpoint.

3.5 NER evaluation

We perform cross-validation with 5 folds on our NER dataset. We evaluate NER performance using the default mode of the `seqeval` (Nakayama, 2018) python library to ensure results can be reproduced.

[3] See Appendix A.2 for details.

[4] We modified the CoNLL-2003 dataset to include honorifics as part of PER entities to be consistent with our annotation guidelines.

4 Results

4.1 Retrieval heuristics

The NER performance for retrieval heuristics can be seen in Figure 1. The `samenoun` heuristic performs the best among global heuristics, whereas the `surrounding` heuristic is the best for local heuristics. While the top results obtained with both heuristics are quite similar, we consider global heuristics as naive retrieval baselines: they could be bested by more complex approaches, which might enhance performance even more.

Interestingly, the performance of both `before` and `bm25` heuristics decrease strongly after four sentences, and even drop behind the `no retrieval` baseline. For both heuristics, this might be due to retrieving irrelevant sentences after a while. The `bm25` heuristic is limited by the similar sentences present in the document: if there are not enough of them, the heuristic will retrieve unrelated ones. Meanwhile, the case of the `before` heuristic seems more puzzling, and could be indicative of a specific entity mention pattern that might warrant more investigations.

4.2 Oracle versions

NER results with the oracle versions of retrieval heuristics can be found in Figure 2.

It is worth noting that the performance of the oracle versions of the heuristics always peaks when retrieving a single sentence. This might indicate that a single sentence is usually sufficient to resolve entity type ambiguities, but it might also be a result of the oracle ranking sentences individually, thereby not taking into account their possible combinations.

Global heuristics perform better than local ones overall, with the oracle version of the `random` heuristic even performing better than both the `before` and `after` heuristics. These results tend to highlight the benefits of using global document context, provided it can be retrieved accurately.

Retrieved sentences To better understand which sentences are useful for predictions when performing global retrieval, we plot in Figure 3 the distribution of the distance between sentences and their retrieved contexts for the oracle versions of heuristics `samenoun` and `bm25`. We find that 8% and 16% of retrieved sentences (for `samenoun` and `bm25`, respectively) are comprised within 6 sentences of their input sentence, while the other are

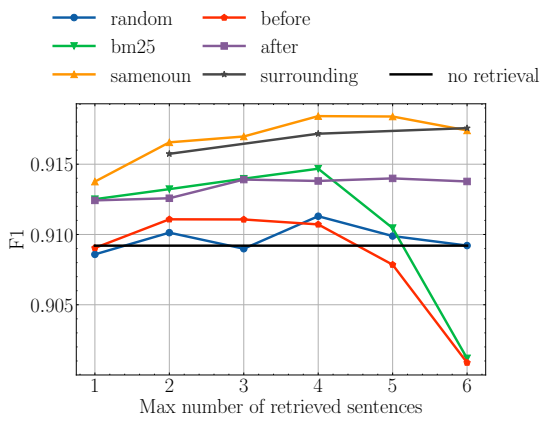

Figure 1: Mean F1 score versus max number of retrieved sentences for all retrieval heuristics across 3 runs.

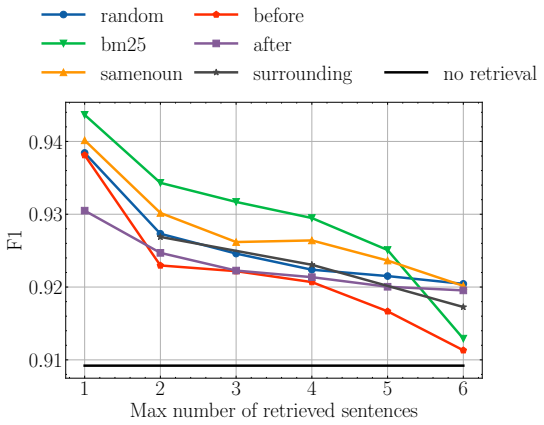

Figure 2: Mean F1 score versus max number of retrieved sentences across 3 runs for oracle versions of all retrieval heuristics.

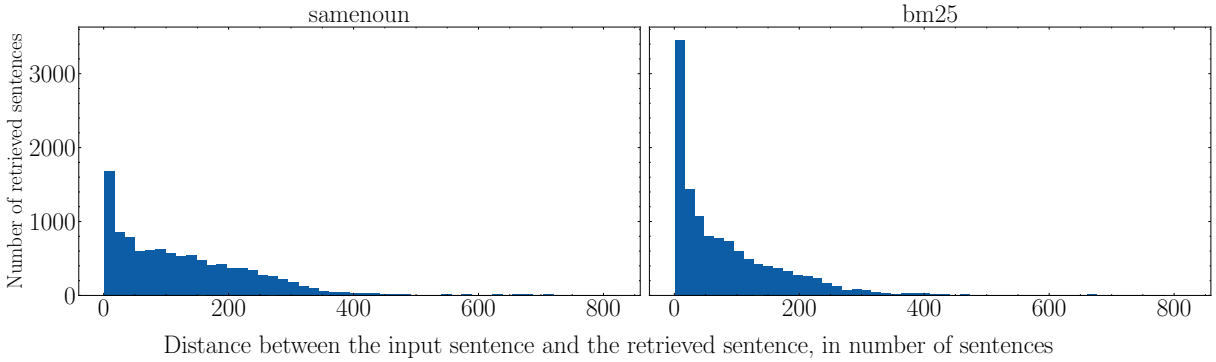

Figure 3: Distribution of the distance of retrieved sentences using the oracle versions of the `samenoun` and `bm25` heuristics. The `samenoun` heuristic retrieves fewer sentences overall, since it is possible for some sentence to not have a common noun with any other sentence of its document.

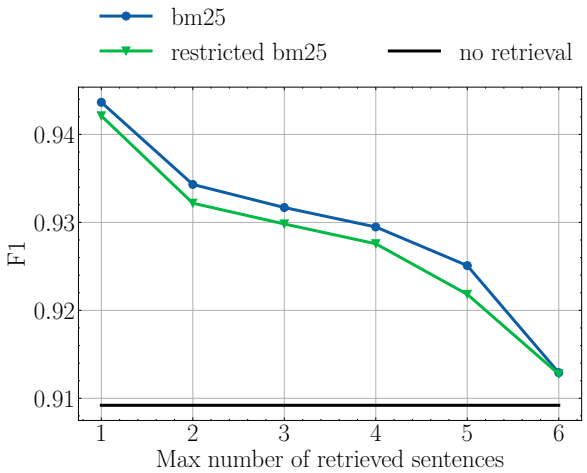

Figure 4: Mean F1 score versus number of retrieved sentences across 3 runs for the oracle version of the `bm25` heuristic, and the same heuristic restricted to distant context.

further away, highlighting the need for long-range retrieval.

Local context importance To see whether or not local context is an important component of NER performance, we perform an experiment where we restrict the oracle version of the `bm25` heuristic from retrieving local surrounding context. Results can be found in Figure 4. NER performance remains about the same without local context, which tends to show that local context is not strictly necessary for performance.

5 Conclusion and Future Work

In this article, we explored the role of local and global context in Named Entity Recognition. Our results tend to show that, for literary texts, retrieving global document context is more effective at enhancing NER performance than retrieving only local context, even when using relatively simple retrieval heuristics. We also showed that a re-ranker model using simple document-level retrieval heuris-

tics could obtain significant NER performance improvements. Overall, our work prompts for further research in how to accurately retrieve global context for NER.

6 Limitations

We acknowledge the following limitations of our work:

- While the oracle selects a sentence according to the benefits it provides when performing NER, it does not consider the interactions between selected sentences. This may lead to lowered performances when the several sentences are retrieved at once.
- The retrieval heuristics considered are naive on purpose, as the focus of this work is not performance. Stronger retrieval heuristics may achieve better results than presented in this article.
- The studied documents only consist in the first chapter of a set of novels. Using complete novel would increase the number of possible information to retrieve for the presented global heuristics.

References

I. Beltagy, M. E. Peters, and A. Cohan. 2020. Longformer: The long-document transformer. *arXiv*, cs.CL:2004.05150.

S. Bird, E. Loper, and E. Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

R. Child, S. Gray, A. Radford, and I. Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv*, cs.LG:1904.10509.

K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller. 2020. Rethinking attention with performers. *arXiv*, cs.LG:2009.14794.

N. Dekker, T. Kuhn, and M. van Erp. 2019. Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ Computer Science*, 5:e189.

J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. 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, pages 4171–4186.

M. Ding, C. Zhou, H. Yang, and J. Tang. 2020. CogLTX: Applying bert to long texts. In *Advances in Neural Information Processing Systems*, volume 33, pages 12792–12804.

D. Guo, D. Tang, N. Duan, M. Zhou, and J. Yin. 2019. Coupling retrieval and meta-learning for context-dependent semantic parsing. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 855–866.

A. Katharopoulos, A. Vyas, N. Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20.

N. Kitaev, Ł. Kaiser, and A. Levskaya. 2020. Reformer: The efficient transformer. *arXiv*, cs.LG:2001.04451.

T. Liu, J. Yao, and C. Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307.

G. Luo, X. Huang, C. Lin, and Z. Nie. 2015. Joint entity recognition and disambiguation. In *2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

J. Luoma and S. Pyysalo. 2020. Exploring cross-sentence contexts for named entity recognition with BERT. In *28th International Conference on Computational Linguistics*, pages 904–914.

H. Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation.

A. Pouran Ben Veyseh, M. V. Nguyen, N. Ngo Trung, B. Min, and T. H. Nguyen. 2021. Modeling document-level context for event detection via important context selection. In *Conference on Empirical Methods in Natural Language Processing*, pages 5403–5413.

S. E. W. Robertson. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, pages 232–241.

D. Seyler, T. Dembelova, L. Del Corro, J. Hoffart, and G. Weikum. 2018. A study of the importance of external knowledge in the named entity recognition task. In *56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 241–246.

T. Stanislawek, A. Wróblewska, A. Wójcicka, D. Ziembicki, and P. Biecek. 2019. Named entity recognition - is there a glass ceiling? In *23rd Conference on Computational Natural Language Learning*, pages 624–633.

Y. Tay, D. Bahri, D. Metzler, D. Juan, Z. Zhao, and C. Zheng. 2020a. Synthesizer: Rethinking self-attention in transformer models. *arXiv*, cs.CL:2005.00743.

Y. Tay, D. Bahri, L. Yang, D. Metzler, and D. Juan. 2020b. Sparse sinkhorn attention. *arXiv*, cs.LG:2002.11296.

Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler. 2020c. Long range arena: A benchmark for efficient transformers. *arXiv*, cs.LG:2011.04006.

E. F. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *7th Conference on Natural Language Learning*, pages 142–147.

S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv*, cs.LG:2006.04768.

X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu. 2021. Improving named entity recognition by external context retrieving and cooperative learning. In *59th Annual Meeting of the Association for Computational Linguistics and 11th International Joint Conference on Natural Language Processing*, volume 1, pages 1800–1812.

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, and A. M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

J. Xu, J. Crego, and J. Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590.

I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Conference on Empirical Methods in Natural Language Processing*, pages 6442–6454.

M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297.

X. Zhang, Y. Jiang, X. Wang, X. Hu, Y. Sun, P. Xie, and M. Zhang. 2022. Domain-specific NER via retrieving correlated samples. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2398–2404.

A Dataset Details

A.1 Document Lengths

Our NER dataset is composed of documents longer that typical NER datasets such as CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003).

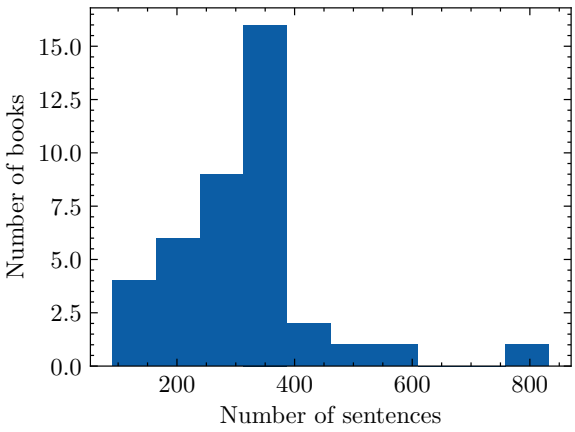

Figure 5: Distribution of the number of sentences in our enhanced version of the dataset from Dekker et al. (2019).

Figure 5 shows the distribution of the number of sentences of our NER dataset.

A.2 Automatic Correction Rules

We use the following rules to automatically identify obvious errors in the original dataset from Dekker et al. (2019). The original dataset only contained `PER` entities, so these rules only apply to them:

- If a span appears in the list of characters from its novel but is not annotated as an entity, we investigate whether or not this is a false negative.
- Similarly, if a span annotated as an entity does not appear in the list of characters from its novel, we investigate whether or not it is a false positive.
- Finally, if a span is annotated as an entity but all of its tokens are not capitalized, we check if it is a false positive.

B Heuristics Results Breakdown by Precision/Recall

Figures 6 and 7 show precision and recall for all retrieval heuristics. Interestingly, retrieval only has a positive effect on recall, with precision being lower than the baseline except for the `surrounding` heuristic.

B.1 Oracle Versions

Figures 6 and 7 show precision and recall for the oracle versions of all retrieval heuristics. While retrieval benefits recall more than precision, precision is still increased using retrieval. Together with

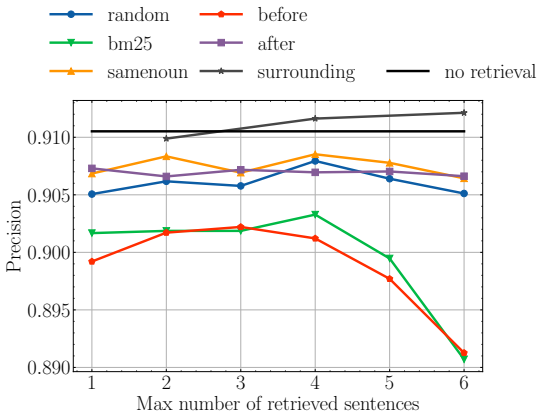

Figure 6: Mean precision versus max number of retrieved sentences for all retrieval heuristics across 3 runs.

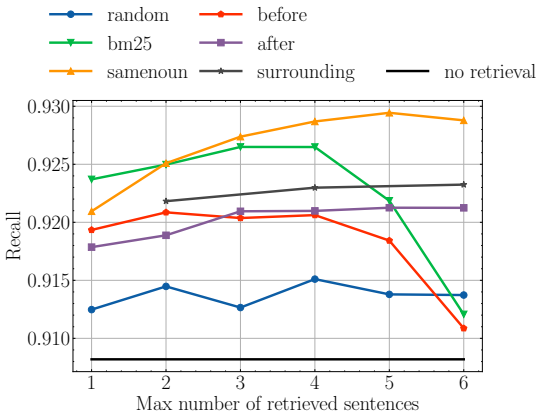

Figure 7: Mean recall versus max number of retrieved sentences for all retrieval heuristics across 3 runs.

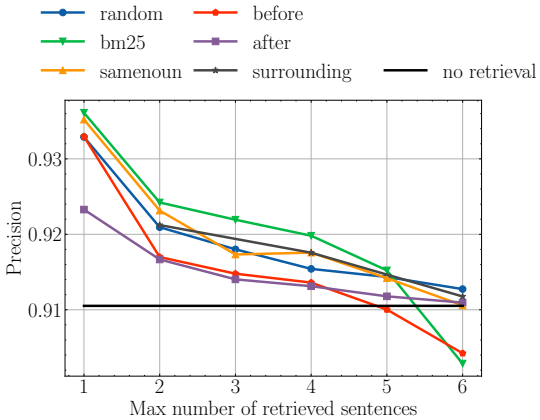

Figure 8: Mean precision versus max number of retrieved sentences across 3 runs for oracle versions of all retrieval heuristics.

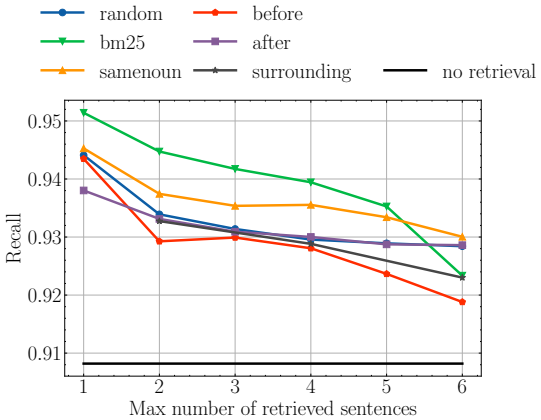

Figure 9: Mean recall versus max number of retrieved sentences across 3 runs for oracle versions of all retrieval heuristics.

the results from the regular heuristics, these results again highlight the potential performance gains of using a suitable re-ranker model to retrieve context.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Yes, limitations are discussed in Section 6

☒ A2. Did you discuss any potential risks of your work?
We do not think our work presents any direct risk

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

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

B ☑ Did you use or create scientific artifacts?

in Section 3.4, we indicate that we use a BERT checkpoint. We also use a previous NER dataset, see Section 3.3. We distribute an enhanced version of this dataset and code to reproduce our experiments.

☑ B1. Did you cite the creators of artifacts you used?
See Section 3.3 and 3.4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We specify the license in the Github repository given at the end of section 1.

☒ 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)?
We use a dataset published for research purposes.

☒ 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?
Collected datas do not include information that can be used to identify individuals

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We specify that the distributed dataset covers english literature (section 3.3). The reader can refer to Dekker et al., 2019 for more informations on the dataset.

☑ 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.
We include the number of document of our dataset in Section 3.3 We also include statistics about the length of these document in the Appendix

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

C ☑ Did you run computational experiments?

See Section 3.4 and Section 3.5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
See Section 3.4

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
We include training hyperparameters in Section 3.4

☑ 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?
Our results are reported in Section 4. We indicate that, for Figure 1 and 2, each point is the mean F1 of 3 runs.

☑ 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.)?
See Section 3.1 (nltk), Section 3.4 (huggingface transformers), Section 3.5 (seqeval)

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

Section 3.3

☒ 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.?
The experiments were free of any risks

☒ 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)?
The authors annotated the dataset themselves

☒ 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?
The authors annotated the dataset themselves

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
The authors annotated the dataset themselves

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
This is not relevant since annotation was done by the authors