CIAD System for Geographical Entity Detection at TextMine'24

Pauline Armary, Cheikh Brahim El Vaigh, Ouassila Labbani Narsis, Christophe Nicolle

CIAD System for Geographical Entity Detection at TextMine'24

Pauline Armary[*,**] Cheikh-Brahim El-Vaigh, [*] Ouassila Labbani Narsis[*] Christophe Nicolle[*]

[*] CIAD, UMR 7533, *Université de Bourgogne, UB*,
64 rue de Sully 21000 Dijon
firstname.lastname@u-bourgogne.fr
[**]Anabasis-Assets
https://www.anabasis-assets.com/fr/

Abstract. This paper outlines CIAD's approach to Named Entity Recognition (NER) in a corpus of nautical instruction. Employing a conventional entity detection approach, our system tackles the NER task through token classification. We handled this classification task using two approaches: 1. NER is performed using different BERT models (BERT-base, Tiny-BERT, and CamemBERT); 2. We also used a GCN-based token classification where a graph is built connecting words in the same context. Our results suggest that within these token classification setups, our models are bounded to an accuracy of around 92% on the test data regardless of the complexity of the used BERT model.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in natural language processing that involves the identification and classification of named entities within textual data (Liu et al., 2022). Named entities are specific entities such as persons, organizations, locations, dates, and more, which carry significance in the given context. NER plays a crucial role in information extraction, and the process typically involves analyzing and annotating text to locate and classify named entities (Liu et al., 2022; Wang et al., 2022).

NER has undergone significant advancements, with various approaches and techniques employed for identifying and classifying named entities in text. Initially, rule-based NER systems were used relying on predefined linguistic rules and patterns to identify named entities in text (Collins and Singer, 1999). Then, machine learning algorithms, such as Conditional Random Fields (CRF) and Hidden Markov Models (HMM), have been applied to learn patterns and features from labeled data (Patil et al., 2020; Morwal et al., 2012). The advent of deep learning models, especially Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformers has ushered in a new era of state-of-the-art performance for NER tasks (Peters et al., 2018; Chiu and Nichols, 2016). Finally, with the success of transformers despite their huge training cost, fine-tuning pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) and GPT (Generative Pre-trained Transformer) has become a common practice.

In the context of the initiative TextMine 2024 [1], focusing on the identification of geographical entities through named entity processing from a French nautical instruction corpus (Rawsthorne et al., 2024), the application of NER becomes crucial for extracting entities from the text. TextMine 2024 is a EGC 2024 [2] evaluation Lab. However, the inherent challenges of the geographical entities context make the NER task more intricate, given that the texts are of a small size and noisy. Moreover, the target entities are not fine-grained types but rather ambiguous. For example, in Tab. 1 which shows the statistics of the different entity types, the annotation *name* refers to a proper noun being it a Person, a Place, or an Organization, while in the standard NER these three categories are separated. Moreover, while NER is often seen as a multi-class classification task, in this evaluation the NER is both a multi-class and a multi-label classification problem, and for example, in Tab. 1, we have 2123 tokens that are both *name* and *geogName*.

This paper proposes two approaches for the NER task in Sec. 2. We first studied several pre-trained BERT models (Devlin et al., 2018) that we fine-tuned on the TextMine corpus and compared their accuracy (see Sec. 2.1). We also devised a graph convolutional network (GCN) (Kipf and Welling, 2017) based approach that considers the tokens in the corpus as nodes and connects these nodes based on their co-occurrence in the same context. Moreover, labels are also seen as nodes and a token is connected to its labels from the train dataset. We then tackle NER as a node classification task using the standard GCN architecture (see Sec. 2.2). The code of our system is on GitHub [3].

Type	geogFeat	geogFeat geogName	geogName	name	name geogName	tokens without label
annotations in the corpus	4167	1469	4490	2118	2123	32668

TAB. 1 – *The different annotations present in the corpus.*

2 CIAD's Systems

2.1 Fine-tuning BERT models

Fine-tuning a BERT model for a specific task is nowadays the state-of-the-art option for several NLP tasks, especially for NER. A BERT (Devlin et al., 2018) model is a Large Language Model (LLM) trained with an Encoder architecture on a very large corpus of documents. As with any LLM, the principle is to transform each word of the corpus into a vector within a large-dimensional space. This vector is reduced using a neural network architecture by optimizing the result to a specific task while reducing the number of dimensions of the vector space.

As part of the Transformer family, BERT uses an Encoder architecture trained on a masked-word prediction task. The Encoder Neural Network is trained to predict what is the correct word which was masked within a sentence. As such, the model uses the overall context surrounding the word to create a vector representing its meaning. For NER task, the model is fine-tuned as a Token Classification or Sequence classification task, with a corpus associating a label of each specific kind of entity (Person, Organization, Place, etc) to each word.

[1] https://textmine.sciencesconf.org/resource/page/id/8
[2] https://iutdijon.u-bourgogne.fr/egc2024/
[3] https://github.com/elvaigh/textmine-ciad

TAB. 2 – *Comparison of the different models.*

	1	2	3	4	5
Model	Bert-base NER	CamemBert NER	GPT2	TinyBert	GCN
Accuracy	0.92771	0.92679	0.91931	0.91508	0.77936

For predicting the label of the corpus, we used a pre-trained BERT model on Sequence Classification for NER, which was pre-trained for recognizing different kinds of entities within the text. We fine-tuned the model to recognize the specific label of our corpus ("geoFeat", "geoName", "name"). We choose a Sequence Classifier as the data is presented as an ordered list of tokens. We fined-tuned the following models: BERT-Base-NER (BERT trained for NER in English), CamemBERT-NER (French BERT trained for NER in French), TinyBERT (refinement of the BERT model), and a NER model using GTP2 within a prompt-based learning setup.

2.2 Using GCN for NER

The Graph Convolutional Network (GCN) architecture was first proposed by (Kipf and Welling, 2017) for node classification. In the context of Natural Language Processing, the text is first transformed into a graph representing the connections between the words that appear in the same context (a window of size 2). For NER, the words are also associated with their labels, each label and token representing a node in the graph. This architecture is often presented as more resilient to multi-label classification, as a node can easily be associated with multiple labels. Within the GCN, an embedding of each node is learned and a softmax layer is used to perform classification based on these embeddings. We used the GCN model to our classification task for geographical entities.

3 Experiments

Experimental validation was conducted on the TextMine 2024 French corpus to assess the quality of our system. We evaluate our system using the accuracy metric. The description of the TextMine corpus can be found in the challenge website. [4]

We discuss hereunder the performance of NER classifiers that we described in Sec. 2.1 and Sec. 2.2. We gathered in Tab. 2 the accuracy of the different classifiers on the test dataset (a.k.a "public score" on the Kaggle page of the challenge [5]). One can see that BERT-based classifiers are the best, reaching an accuracy of 92.7%. Meanwhile, the difference between the smallest model (TinyBERT column 4 in Tab 2) and the largest model (BERT-base column 1 in Tab 2) is rather small (1.2%). This conclusion can be explained by the fact that, for the fine-tuning task, only a subset of the parameters is used (unfrozen parameters), therefore most the parameters of the model are kept as they are. We also show the accuracy of the GPT2 model using prompt learning (column 3 in Tab 2). We notice that this model shows a comparable performance to the BERT-based ones. Finally, the GCN NER classifier did not perform very well on the test dataset and its accuracy is only 77.9% (column 5 in Tab 2). The GCN classifier showed

[4] https://textmine.sciencesconf.org/resource/page/id/8

[5] https://www.kaggle.com/competitions/defi-textmine-2024/

promising results on the training dataset, as most of the nodes are known and the underlying graph creation is straightforward. However, we noticed that this GCN tends to over-fit on the train data and could not generalize well. We believe this is due to the way the graph is created (connecting nodes that appear in the same context), as with the provided TextMine dataset, each word has only limited context. This approach is not suitable for small datasets as it will require different contexts for each word.

4 Conclusion

We built an entity processing system based on a BERT model for the NER task, exploiting the existing pre-trained models. Our system was evaluated on the TextMine 2024 French dataset. The proposed model achieved an accuracy of 92.7% being only 5% points away from the best model. We also proposed a GCN model that did not perform well due to the small size of the data. These results open new perspectives for taking into account BERT embedding in a GCN model for NER.

References

Chiu, J. P. and E. Nichols (2016). Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics 4*, 357–370.

Collins, M. and Y. Singer (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT conference on empirical methods in natural language processing and very large corpora*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kipf, T. N. and M. Welling (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Liu, X., H. Chen, and W. Xia (2022). Overview of named entity recognition. *Journal of Contemporary Educational Research 6*(5), 65–68.

Morwal, S., N. Jahan, and D. Chopra (2012). Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol 1*.

Patil, N., A. Patil, and B. Pawar (2020). Named entity recognition using conditional random fields. *Procedia Computer Science 167*, 1181–1188.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). Deep contextualized word representations. In M. Walker, H. Ji, and A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, pp. 2227–2237.

Rawsthorne, H., N. Abadie, A. Guille, P. Cuxac, V. Lemaire, and C. Lopez (2024). Reconnaissance d'entités géographiques dans un corpus des instructions nautiques (2024) défi textmine'24). *Conférence Extraction et Gestion des Connaissances 2024 (EGC'24)*.

Wang, Y., H. Tong, Z. Zhu, and Y. Li (2022). Nested named entity recognition: a survey. *ACM Transactions on Knowledge Discovery from Data (TKDD) 16*(6), 1–29.