**Name Entity Recognition in NLP**

Divya Sri Nunety,

Rishi Dasari,

Teja Sanaka

Department of Data Science

University of Maryland Baltimore County

Data 690 Natural Language Processing

Dr. Antonio Diana

May 2024

**Abstract**

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task that identifies and classifies significant words or phrases in texts, such as names of persons, organizations, locations, etc. In other words, NER helps recognize named entities. This research proposes named entity recognition models using BERT, Bi-LSTM and DistilBERT architecture. BERT is used for a deeper semantic understanding of words considering their context; on the other hand, Bi-LSTM helps to remember word order information. CoNLL-2003 dataset, one of the most widely known ones, was chosen to train the model and AdamW optimizer was adopted for faster learning in our model. Results of BERT, DistilBERT and Bi-LSTM against validation and test set were compared. SpaCy, an open-source natural language processing (NLP) library which comes with pre-trained NER models was also examined to built a streamlit NER app. We suggest a future work direction where both BERT and bi-directional LSTM can be jointly employed in NER systems.

Key Phrases: BERT, Bi-LSTM, Conll-2003 dataset, AdamW, spaCy.

**Introduction**

To identify and tag significant information in text, such as names of individuals, locations or organizations is the main objective of Named Entity Recognition (NER). This is essential for many applications like information retrieval, language translation, question-answering systems, data analysis, etc. In the past, there have been different approaches to NER systems including rule-based methods as well as statistical learning methods. However, these approaches tend to fail when faced with human language that is ambiguous due to context and calls for domain specific adjustments.

A lot of difficult problems have been addressed by deep learning. Understanding language with computers has really changed thanks to models like BERT (Bidirectional Encoder Representations from Transformers).BERT has changed the way the NER task used to be done prior, with its transformer architecture. It enhanced the understanding of the multi-levels of context with bidirectional training and a multi-layered approach; this model sets new benchmarks across various NLP tasks.

DistilBERT is a more efficient but still powerful model that requires much less resource retention than many of BERT's predictive capacities. In other words, it's a small and portable model, so perfect for conditions when the computational performance cannot be given up.

Furthermore, spaCy is a Python library that contain NER predefined model. Using a spaCy model, ensures efficient results for our system while taking advantage of spaCy’s optimized NLP pipelines and pretrained entity types would speed up training time and make deployment easier too.

The training and valuation of this NER models are detailed in this article. It talks about how models were trained  using the CoNLL-2003 dataset, what optimization techniques were used, and what results were obtained. Additionally, it shows where improvements can be made so that the performance level of the models used can rise in future applications concerning named entity recognition (NER).

**Literature review**

An important study in the area of Named Entity Recognition (NER) that involved the use of deep learning models was conducted by Chiu and Nichols (2016), who combined bidirectional LSTM and CNN. They showed that LSTM, which is useful for processing sequences of data, can be effectively paired with CNN, which handles character-level representations to improve recognition of text patterns. This aligns with the current project’s objectives, which seeks to validate the performance Bi-LSTM. Similarly, Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016) show how deep learning models — especially those that use information about characters can improve NER systems. They suggest that automated text processing tools could greatly benefit from these neural architectures and thus become better at information extraction as well as retrieval across various applications. What sets these designs apart from previous ones is their ability to adapt dynamically; therefore, they are able to process data more efficiently than ever before while also making it accessible for further use.

Liu, Z., Jiang, F., Hu, Y., Shi, C., & Fung, P. (2021) compared the performance of custom BERT models trained for different purposes. Although this paper does not focus on custom NERs. Metric used for comparing different models aligns with the research.

Ritter, A., Clark, S., Mausam, & Etzioni, O. conducted an experiment in 2011, their study shows the challenges to make NER systems capable of working well with social media texts to foster instant analysis and useful information extraction. These developments have wide-ranging implications which can revolutionize various fields including market research or even disaster management through more precise and timely examination of contents from social media during natural disasters.

According to Huang, Xu, and Yu (2015), Bidirectional Long Short-Term Memory (Bi-LSTM) with Conditional Random Fields (CRF) models is a big step forward in Natural Language Processing (NLP). Mainly for sequence tagging tasks. Their research shows that the BI-LSTM-CRF model can better recognize entities by considering the context of both previous and next items along a sequence. What stands out about this paper in relation to the paper is BI-LSTM works good when layered over CRF or BERT.

In their research on Joint Word Embeddings (JWE) models Yu, Jian, Xin, and Song (2017) have made a significant contribution to the methodology for improving word embeddings that are tailored specifically to the Chinese language. The knowledge provided by this paper can be extremely useful in BERT-based Named Entity Recognition (NER) research using Bi-LSTM. If we could obtain more semantic features from complex scripts, then it may help BERT or Bi-LSTM with feature extraction capabilities leading towards better recognition and classification of entities according to their morphological as well as semantic properties.

In 2022, Alves-Pinto, A., Demus, C., Spranger, M., Labudde, D., & Hobley, E’s research shows how much better Named Entity Recognition (NER) accuracy can get using Conditional Random Fields (CRF) in iterative training. The systems are trained to have more correct understanding and recognition of the entities by working in human annotations repeatedly. The traditional training methods are far behind this new one because it helps many languages and domains become accurate which means that there is no need for adaptation anymore. This finding may lead to faster NER systems development, thus changing data processing techniques across various industries through accurate identification of events happening within different contexts.

Walha, A., Ghozzi, F., & Gargouri, F (2017) made a significant contribution to the field of sentiment analysis. The study shows that it is effective to use a fixed list of words to measure emotions contained in messages shared through social media, showing the importance of NER in NLP pipelines.

The work done by Satheesh, K., Jahnavi, A., Iswarya, L. (2020), to make Resume Parser using SpaCy NER model shows the efficiency of SpaCy NER models and how we can make custom models using SpaCy can be applied to this research.

**Methodology:**

**Dataset**

CoNLL-2003 dataset is known as one of the most popular datasets for training and testing Named Entity Recognition (NER) systems. It was first presented at the Conference on Natural Language Learning in 2003. The dataset contains texts in both English and German, but most commonly, only the English part is used. It is derived from Reuters news stories published in 1996 and aims to help develop systems that can automatically find specific names or terms in texts such as people’s names, locations, organizations’ names or miscellaneous names such as events or nationalities.

Structure of CoNLL-2003 Dataset:

This dataset consists of three parts:

1. Training Set: This section covers 946 sentences which are used for training NER models to recognize and categorize names/terms.

2. Validation Set: Comprising 203 sentences, this set helps fine-tune models by developers.

3. Test Set: This set has 246 sentences to evaluate how good an NER model works.

Each word in the dataset is denoted to a specify NER tag (9 labels):

['I-MISC', 'I-ORG', 'I-PER', 'I-LOC', 'B-PER', 'B-LOC', 'O', 'B-MISC', 'B-ORG']

B (Begin) marks the beginning of a tag

I (Inside) designate other words within the same tag (following the first word).

O (Outside) is used for words that are not part of a tags in the dataset (ex: and, is, lives).

The names and terms are classified into several categories:

PER (Person): Names of people, e.g., “John”.

LOC (Location): Names of places, e.g., “Paris”, “Africa”.

ORG (Organization): Names of groups or businesses, e.g., “United Nations”, “Google”.

MISC (Miscellaneous): Other terms, including events or products, e.g., “Olympic”, “Windows”.

**NER System Architecture:**

The System Architecture this paper follows to make an NER model is shown in Figure1.

A diagram of a software process

Description automatically generated

*Figure 1: NER system architecture.*

**Data Cleaning and Preprocessing:**

This is a critical phase to ensure data usability and quality. The train, test and validation datasets were in text format (‘EU NNP B-NP B-ORG’), where each line has 4 space separated entities. The first one denotes the word, and the last denotes the corresponding NER tag. So, we built a Dataset class to separate these entities and make a list of sentences, where each sentence is again a list of words.

**Bert Tokenization**

The BERT tokenizer receives text and turns it into tokens for the BERT model to read. It does this using a method known as WordPiece, which splits words unknown to the model into smaller pieces that it can understand. Additionally, special tokens like [CLS] are added to the start and [SEP] between sentences, while [PAD] is used for padding so that all sentences have equal length but still represent different sentences (Liu et al.,2021).

Then each token is given a unique number from BERT’s vocabulary – this changes its form into something that can be processed by our model. This was ensured by tokenizing the data using pretrained BERT tokenizer.

**Model Training:**

**BERT Model:**

The BERT is a Token Classification Model to categorize entities in text. BERT is capable of capturing both the left and right context of a word. To Train the model, the training data is tokenized using the tokenizer function. Then, each token is represented as a high dimensional vector through embedding layers. The model is tuned using AdamW optimizer.

**Bi-LSTM Model:**

The Bi-LSTM model is a kind of LSTM (Long Short-Term Memory) that has been modified to be able to process data in two ways: forward and backward. This, means that the model can now better understand the context by considering what happens both before and after any given point within a sequence. The training dataset is converted as a high-dimensional vector using word embeddings. The training was done in batches for 3 epochs for simplicity.

**DistilBERT Model:**

A fine-tuned DistilBERT model from Hugging Face was used loaded, and trained against tokenized training dataset and evaluated with validation set, and stored the optimal checkpoint.

**SpaCy NER Model:**

SpaCy has a default NER model which can recognize a wide range of named entities. Unlike Conll 2003 Dataset which has only 4 entities (PER, LOC, ORG, MISC).

**Model Evaluation:**

The three models were evaluated with metrics precision, Recall, F1 Score and accuracy.

A lower F1 score indicates that the model is not performing well in both Recall and precision. Which means the model is incorrectly classifying instances. On the contrary, a high F1 score indicates that the model is performing the classification task well.

**Table 1: Models Evaluation Results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Metrics** | **BERT** | **Bi-LSTM** | **DistilBERT** |
| **Recall** | 0.79 | 0.11 | 0.93 |
| **Precision** | 0.80 | 0.09 | 0.91 |
| **F1 Score** | 0.78 | 0.10 | 0.92 |
| **Accuracy** | 0.95 | 0.84 | 0.98 |

**Building NER Web APP using SpaCy:**

Using Python Streamlit library and SpaCy pre-built NER model. The app has a menu containing Tokenization, NER and knowledge graphs. Where the user can input text and can visualize tokens, classify Named Entities, or plot Knowledge graphs.

**Results**

**Dataset EDA:**

**A graph with blue and white bars

Description automatically generated with medium confidence**

*Figure 2: Tag Distribution in Dataset*

**Model Evaluation:**

As one can see from Table 1. The Bi-LSTM model has low F1 Score but high Accuracy. This suggests that the model is not classifying instances. But still has high accuracy this is because the tags distribution is skewed more towards the ‘O’ (others) tag. The model might be biased towards the tag and since its number is humongous compared to other tags, it’s quantity even dominates in validation set. So, in this case, accuracy is misleading. While the DistilBERT has the highest accuracy and F1 score among the three. On the other hand BERT has an appreciable performance balancing both F1 score and accuracy.

**NER Web App:**

A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

*Figure 3: NER Web App interface*

**Conclusion**

Through this paper, a BERT, Bi-LSTM and DistilBERT models for Named entity recognition were trained and valuation was performed against performance metrics. F1score, precision, Recall, and accuracy. It was found that Bi-LSTM was not performing good enough on further evaluation found that the accuracy can be improved by using BERT as word2Vec tokenizer in combination with Bi-LSTM can improve the accuracy. BERT, due to it inherent transformer structure performed well, but due to system limitations it was trained for only 3 epochs. Increasing the epochs may increase the F1 Score and accuracy. On the other hand, DistilBERT has the highest F1 Score and accuracy among the three.

A Web Application was also made utilizing the pretrained NER- model from spaCy using Python Streamlit.

**References**

Alves-Pinto, A., Demus, C., Spranger, M., Labudde, D., & Hobley, E. (2022).

Iterative Named Entity Recognition with Conditional Random Fields. Applied Sciences, 12(1), 330. https://doi.org/10.3390/app12010330

Chiu, J. P. C., & Nichols, E. (2016).

Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 4, 357-370. Retrieved from http://direct.mit.edu/tacl/articlepdf/doi/10.1162/tacl\_a\_00104/1567392/tacl\_a\_00104.pdf.

Huang, Z., Xu, W., & Yu, K. (2015).

Bidirectional LSTM-CRF models for sequence tagging. Retrieved from https://arxiv.org/abs/1508.01991.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016).

Neural architectures for named entity recognition. Retrieved from https://arxiv.org/abs/1603.01360.

Liu, Z., Jiang, F., Hu, Y., Shi, C., & Fung, P. (2021).

NER-BERT: A Pre-trained Model for Low-Resource Entity Tagging. Retrieved from http://arxiv.org/abs/2112.00405.

Moalla, I., Nabli, A., Bouzguenda, L., & Hammami, M. (2016).

Data warehouse design from social media for opinion analysis: the case of Facebook and Twitter. In Proceedings of the IEEE.

Ritter, A., Clark, S., Mausam, & Etzioni, O. (2011).

Named entity recognition in tweets: An experimental study. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 1524-1534.

Satheesh, K., Jahnavi, A., Iswarya, L., Ayesha, K., Bhanusekhar, G., & Hanisha, K. (2020).

Resume Ranking based on Job Description using SpaCy NER model. International Research Journal of Engineering and Technology, 7(5), 74-77.

Walha, A., Ghozzi, F., & Gargouri, F. (2017).

A Lexicon Approach to Multidimensional Analysis of Tweets Opinion. In Proceedings of the IEEE, Sfax, Tunisia.

Yu, J., Jian, X., Xin, H., & Song, Y. (2017).

Joint embeddings of Chinese words, characters, and fine-grained sub character components. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 286–291). Copenhagen, Denmark.