ENSEMBLE MODEL FOR CHUNKING

Nilamadhaba Mohapatra, Namrata Sarraf and Swapna sarit Sahu

Department of Data Science, Zeotap, Bangalore, India

ABSTRACT

Transformer Models have taken over most of the Natural language Inference tasks. In recent times they have proved to beat several benchmarks. Chunking means splitting the sentences into tokens and then grouping them in a meaningful way. Chunking is a task that has gradually moved from POS tag-based statistical models to neural nets using Language models such as LSTM, Bidirectional LSTMs, attention models, etc. Deep neural net Models are deployed indirectly for classifying tokens as different tags defined under Named Recognition Tasks. Later these tags are used in conjunction with pointer frameworks for the final chunking task. In our paper, we propose an Ensemble Model using a fine-tuned Transformer Model and a recurrent neural network model together to predict tags and chunk substructures of a sentence. We analyzed the shortcomings of the transformer models in predicting different tags and then trained the BILSTM+CNN accordingly to compensate for the same.

KEYWORDS

Natural Language Processing- Named Entity Recognition, Chunking, Recurrent Neural networks, Transformer Model.

1. INTRODUCTION

Chunking is the process of splitting the words of a sentence into tokens and then grouping the tokens in a meaningful way. These chunks are our point of interest which are used to solve our relevant NLP tasks[3]. It labels every word of the sentence suitably and thus lays out a basic framework for bigger tasks such as question answering, information extraction, topic modeling, etc[16]. Named Entity Recognition as mentioned in the paper[2][10][11]is the process of extracting and tagging words that signify the names of certain places, people, organizations, time, etc. Several Natural Language Understanding Tasks like POS tagging, Tokenization, and Noun Phrase Identification are all chunking tasks.

In recent times Chunking has also been used in various domain-level projects with an end goal of retrieving custom substructures of a sentence[18]. We have employed the chunking model proposed in a similar use case. Since its wide use and application chunking requires a constant push to the state of the art models. We have hence proposed a model where we use RoBERTa[9], a Transformer Model ensemble with an RNN based BILSTM+CNN [8]The Transformer Model helps in attention maximization and hence expanding the learning abilities of the Model. It provides embeddings for words in a sentence highly correlated with other words in the sentence owing to its multi-head attention mechanism[19], and thus reflects a more relative semantic of a word instead of using just the POS tag for embeddings.[9] We have used pre-trained models that facilitate the understanding of words and their place from a much larger dataset and fine-tuning it further customizes and improves the model understanding of words related to specific tags associated with it.

David C. Wyld et al. (Eds): BIoT, DKMP, CCSEA, EMSA, NLCAI - 2021
pp. 113-119, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110811

In our Recurrent Neural Network section of the Model, we use Bidirectional LSTM in addition to Convolutional Neural Networks CNN[8]. We introduced this segment of the model to compensate for the shortcomings of the Transformer Model and give a more general approach to Tagging and segmentation for chunking. Since LSTMs use the sequential word by word approach of processing, they can be slower in processing, and hence we use simpler shorter custom embeddings targeting exactly the common error points of Transformer Models[9]. The word embedding essentially utilizes a combination of positional embedding, POS Tag, and the word vector after whitespace tokenization. Our analysis of the RoBERTa Model showed an error in differentiating similarly tagged tokens and hence a boost to the Part of speech tagging is given here. The BiLSTM network is complemented with a stack of CNN layers to aid feature extraction from sentences and the feedback loop helps in accurately labelling these features and further segmenting them into chunks.

2. Prior Work

Named Entity Recognition was initially performed using extensive knowledge base systems, its orthographic features, ontological and lexicon rules[4][2][14][15]. However, the new trend has shifted towards neural network-based structures to define entity relations [5][13]. Hence the mentioned top 20 state-of-the-art NER mechanisms are neural network-based including LSTM, GRUs, BERT, CNN, and a combination of them as suited. [2] [6][10][11][12].

Chunking has been done using machine learning-based models such as HMM(Hidden Markov Model) [7][17] and Maximum Entropy model and has gradually seen a shift to Statistical models such as Support Vector Machines and Boosting [8], [3], [7]. In more recent times, Neural Models have been on a rise as a tool for chunking. Neural network models are deployed as a classification system to classify Beginning, Inside, and Outside of the chunks required or also known as BIO tagging which is quite a popular Named Entity Recognition mechanism for segmentation. The latest paper submitted by IBM Watson uses a combination of Bi-LSTM and CNN to label the tokens of the sentence and then chunk them together accordingly [8]. They follow an encoder-decoder-pointer framework while segmenting and labelling chunks sequentially using a pointer. We use POS tag reinforced word vectors as input to this segment of the model. Since the model excelles majorly in sequential labelling and feature extraction, it helps widen the gap between similar tags occurring together frequently.

In this paper, we have deployed an ensemble model using the Transformer-based Model RoBERTa [9] and recurrent neural network[8] for labeling and segmentation, after which we group the hence labeled chunks and map the contexts and phrases together. Our experiment gives an F1 score 97.3 which beats the F1 score of the chunking methods employed in the paper by IBM 94.72. [8] both tested on the common CoNLL 2000 dataset. F1 score here signifies the harmonic mean of precision and recall derived from the confusion matrix built on test dataset to evaluate the classifier trained for entity tag labels which are then further used for chunking.

3. Data

We have evaluated our Model on two main datasets, English Penn treebank[20] corpus and ConLL 2000 Dataset. The English Penn Treebank corpus, is an extensively used corpus for the evaluation of language models. The task includes annotating each word with its Part-of-Speech tag. There are 38,219 sentences, used for training, 5,527 sentences used for validation and 5,462 sentences are used for Testing purposes. ConLL 2000[21] is a widely used dataset for noun phrase chunking with 211727 training tokens and 47377 testing tokens. The annotation of the data is procured from the WSJ corpus by an automated program written by Sabine Buchholz from

Tilburg University. The objective of this task is to introduce machine learning methods which after training recognizes the chunk segmentation of the test data as accurately as possible.

4. Experiments

The Named entity task in our experiment is to classify tokens into IOB annotations of either custom tags, POS tags or Entity tags for example organisation, person etc. These tags are then used for segmentation using the Inside and Outside tags and give a final output of chunked phrases of a sentence. The chunks are made as accurately as the tags are predicted.

Table 1. Comparative Analysis of Algorithms. Dataset (ConLL 2000 english).

Model	**ConLL 2000**
RoBERTa +(Bi-LSTM+CNN)	97.3
RoBERTa	96.76
BERT	95.64
Bi-LSTM+CNN	94.72

4.1. Model I

We deployed two Models to train and test for the Named Entity Recognition task.

The neural model we chose to train our NER downstream task was RoBERTa. This model is a robustly optimized form of BERT(Bidirectional Encoder Representations from Transformers) and is state of the art on 4 out of 9 GLUE (General Language Understanding Evaluation) tasks. The model is a modified version of BERT wherein it targets the flaws of BERT. BERT is pre-trained over 16GB of uncompressed text, which is a combination of Book Corpus[22] and English Wikipedia[23]. However, RoBERTa is trained on five English language corpora summing up to 160GB of uncompressed text. The huge upscale of data for pretraining gives RoBERTa the edge over BERT. In addition, RoBERTa omits the Next Sentence Prediction task from the pretraining, trains on longer sequences, and dynamically revises the masking patterns on training data [12].

We chose a Transformer Model as our neural architecture for generalization to bring in the concept of attention to the structure.[9] In the downstream task of Named Entity recognition, we need the complete semantics of the sentence. For this purpose, the attention weight of every token or time step for every other token present in the sentence is essential.

RoBERTa is consistent with the mathematics of BERT as well as the number of training layers, attention heads, and parameters. For our task, we used the RoBERTa base model which has 12 Layers of deep neural architecture, 16 Attention heads, and 110 training parameters. The model was fine-tuned to train for 6 epochs with a training batch size of 64 and a learning rate of 2^*e^{-5}. The model pre-trained for the Named Entity recognition task is imported.

4.2. Model II

The second model is an ensemble training of RoBERTa and Bi-LSTM + CNN.

The purpose of ensemble training is to boost the learning process of neural architecture. We fed different feature vectors into Model A and Model B to capture the entire meaning of a token. We deploy this model to further increase our F1 score by using two complementary training models.

RoBERTa uses its encoder to tokenize and embed the tokens into feature vectors. In the Bi-LSTM+CNN model, we use a custom tokenizer after analysing the misclassifications of Model I. In our analysis, we found that similar types of Parts of speech e.g.: Symbols, and Punctuations, and Nouns and Proper nouns were getting miss classified. To rectify this error, we built a custom tokenizer and embedding explained in the later sections.

4.2.1. MODEL A

RoBERTa, Refer Figure 1, model takes care of attention and co-dependency of tokens frequently coming together in a sentence. It uses RoBERTa Tokeniser and RoBERTa Embedding to encode the entire sentence and is then trained over for 6 epochs. We have explained the purpose of using RoBERTa in the earlier sections.

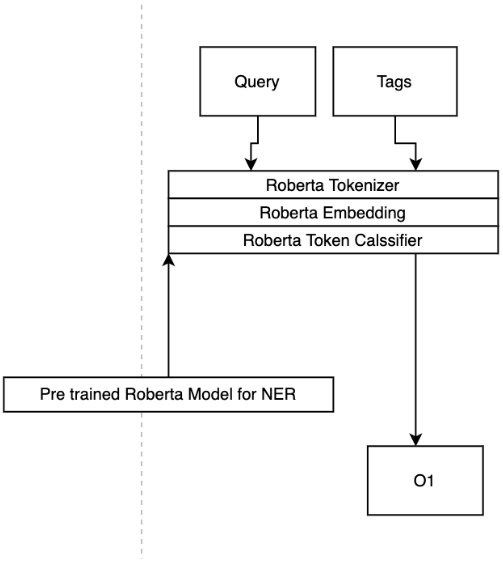

Figure 1. Transfor Model: RoBERTa: This model produces Output 1 or O1

4.2.2. MODEL B

The second leg of the ensemble training, Refer Figure 2, uses a custom tokenizer wherein every query is whitespace tokenizer, every token is passed into a word vectorizer and horizontally stacked with it's one hot encoded part of speech tag. This array is then multiplied with positional encoding of the token which finally outputs the token's embedding. We have used three features of a token namely its word semantics, position at which it shows up in the sentence and it's part of Speech in this Tokenizer which makes up for any misclassification that is brought about by Model A.

$$\textit{word embedding} = \textit{positional encoding}*([\textit{word vector, ohe pos tag}]) \quad \text{...Eqn.1}$$

where,

positional encoding = position of token in query / length of query

ohe pos tag = one hot encoded part of speech tag.

These word embeddings are sent to the BiLSTM layer followed by 1 dimensional convolution, max pooling and the vector is then flattened out before introducing non linearity and classification on top of it.

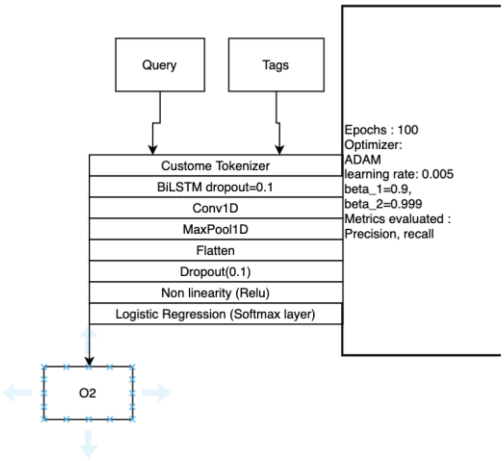

Figure 2. BiLSTM+CNN: This model produces Output 2 or O2 after the query goes through a custom tokenizer and embedder to further pass through the RNN and CNN layers

4.2.3. Ensemble

Weighted Average of Classification scores of Model A and Model B are then used for final Multi class classification.

We trained Model II for 150 epochs using the Adam Optimiser and a learning rate of 0.01. The model converged to an F1 score of 97.3 which is higher than either of the models described individually. We can thus conclude with this experiment that ensemble training using custom tokenizer encapsulates more information required for a Named Entity Recognition Task. The same segmentation and labelling approach is used here as Model I.

$$\textit{final classification scores} = w1*O1 + w2*O2 \qquad \text{.... Eqn. 2}$$

where,
O1, *O2* are outputs from Model A and Model B respectively.
w1, *w2* are the weights associated with it.

Table 2. Comparative Analysis of Algorithms. Dataset (PENN Treebank).

Model	**CoNLL 2000**
RoBERTa +(Bi-LSTM+CNN)	98.4
IntNet + BiLSTM-CRF	95.29
NCRF++	95.06

5. Observations, Results, Benchmarks

The training sentences are preprocessed into a [token]-[tag]-[sentence-id] format for training. Every query is whitespace tokenized, every token hence obtained is mapped to its annotated tag and a sentence id corresponding to the query. A GPU Nvidia Tesla K80 is used for training which takes about 160 minutes per epoch. Naturally, for this kind of task the evaluation metric F1 score is taken into account because the data set has imbalanced tags and accuracy might therefore be biased towards the majority tag.

After training the model, a layer of segmentation is deployed to chunk the phrases and contexts together to further pass onto the last layer of the model to map the relevant context and phrases together. The result of the model hence is a dictionary where key, value pairs are the context phrase(s) pairs.

Comparing our results with the present state of the art for chunking[10], on conll 2000 data set Refer Table 1, the model exceeds the state of the art neural net model which uses Bi-LSTM in conjunction with CNN for encoder-decoder labeling and pointer framework for segmentation as mentioned before. Our model obtains an F1 score of 97.3 exceeding the preceding State of the Art 94.72 for conll 2000 dataset. The comparison is also done on 23 labels for 9000 training sentences and 900 testing sentences. On the Penn treebank dataset, Refer Table 2, our model achieves an F1 score of 98.4 for classifying the chunking tags which exceeds the present state of the art[8] with a score 95.29 which uses IntNet + BiLSTM-CRF..

6. CONCLUSIONS

In this paper, we aim to highlight the significance of the ensemble models. In the first base model, we used the current state-of-the-art model i.e. the transformer-based model. It uses an attention framework to understand the semantics of long-length sentences better. We found that the transformer-based model does the misclassifications at various Symbols, and Punctuations, and Nouns and Proper nouns. To compensate for this in the second base model, we explicitly provided POS tags of the token explicitly to the model. We used a BILSTM+CNN framework with explicit pos tags for each token. The CNN part optimizes the local context well and The BILSTM with pos tag tries to capture the context dependency with the surrounding word better. We experimented with this ensemble model on various open datasets for chunking tasks and found that this ensemble architecture broke the previous state-of-the-art. In future work, we would apply this architecture to different GLUE tasks to achieve similar kinds of results.

REFERENCES

[1] Nothman And James R. Curran And Tara Murphy, “Transforming Wikipedia Into Named Entity Training Data,” 2008.

[2] V. Yadav and S. Bethard, “A survey on recent advances in named Entity Recognition from deep learning models,” arXiv [cs.CL], 2019.

[3] E. Muszyńska, “Graph- and surface-level sentence chunking,” in Proceedings of the ACL 2016 Student Research Workshop, 2016.

[4] A. Siddharthan, "Complex Lexico-syntactic Reformulation of Sentences Using Typed Dependency Representations", ACL Anthology, 2021. [Online]. Available: https://www.aclweb.org/anthology/W10-4213.

[5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” arXiv [cs.CL], 2016

[6] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” in Benjamins Current Topics, Amsterdam: John Benjamins Publishing Company, 2009, pp. 3–28.

[7] T. Kudo and Y. Matsumoto, “Chunking with Support Vector Machines,” J. Nat. Lang. Process., vol. 9, no. 5, pp. 3–21, 2002.

[8] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, “Neural Models for Sequence Chunking,” arXiv [cs.CL], 2017.

[9] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv [cs.CL], 2019.

[10] R. Chalapathy, E. Zare Borzeshi, and M. Piccardi, “An investigation of recurrent neural architectures for drug name recognition,” in Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis, 2016.

[11] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in Proceedings of the 25th international conference on Machine learning - ICML ’08, 2008.

[12] F. Dernoncourt, J. Y. Lee, and P. Szolovits, “NeuroNER: an easy-to-use program for named-entity recognition based on neural networks,” in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2017..

[13] R. Panchendrarajan and A. Amaresan, "Bidirectional LSTM-CRF for Named Entity Recognition", ACL Anthology, 2021. [Online]. Available: https://www.aclweb.org/anthology/Y18-1061.

[14] Y. Li, K. Bontcheva, and H. Cunningham, “SVM based learning system for information extraction,” in Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 319–339.

[15] W. Etaiwi, A. Awajan, and D. Suleiman, “Statistical Arabic name entity recognition approaches: A survey,” Procedia Comput. Sci., vol. 113, pp. 57–64, 2017.

[16] E. J. Otoo, D. Rotem, and S. Seshadri, “Optimal chunking of large multidimensional arrays for data warehousing,” in Proceedings of the ACM tenth international workshop on Data warehousing and OLAP - DOLAP ’07, 2007.

[17] H. Sharma, "Survey of Research on Chunking Techniques", Semanticscholar.org, 2021. [Online]. Available: https://www.semanticscholar.org/paper/Survey-of-Research-on-Chunking-Techniques-Sharma/ced929651b222d3b489df1c25ed9c801c7c3e749.

[18] P. S. Rosenbloom and J. Aasman, “Knowledge level and inductive uses of chunking (EBL),” in Soar: A Cognitive Architecture in Perspective, Dordrecht: Springer Netherlands, 1992, pp. 219–234.

[19] A.Vaswani et al., “Attention is all you need,” arXiv [cs.CL], 2017.

[20] M. Marcus, B. Santorini and M. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank", ACL Anthology, 2021. [Online]. Available: https://www.aclweb.org/anthology/J93-2004.

[21] E. F. Tjong Kim Sang and S. Buchholz, “Introduction to the CoNLL-2000 shared task: Chunking,” in Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning -, 2000.

[22] Y. Zhu et al., “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in 2015 IEEE International Conference on Computer Vision (ICCV), 2015.

[23] Wikipedia: Database download