Journal of Information Systems and Telecommunication
Vol.11, No.3, July-September 2023, 176-184

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Deep Transformer-based Representation for Text Chunking

Parsa Kavehzadeh[1], Mohammad Mahdi Abdollah Pour[1], Saeedeh Momtazi[1*]

Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran

Received: 27 Jul 2021/ Revised: 20 May 2022/ Accepted: 18 Jun 2022

Abstract

Text chunking is one of the basic tasks in natural language processing. Most proposed models in recent years were employed on chunking and other sequence labeling tasks simultaneously and they were mostly based on Recurrent Neural Networks (RNN) and Conditional Random Field (CRF). In this article, we use state-of-the-art transformer-based models in combination with CRF, Long Short-Term Memory (LSTM)-CRF as well as a simple dense layer to study the impact of different pre-trained models on the overall performance in text chunking. To this aim, we evaluate BERT, RoBERTa, Funnel Transformer, XLM, XLM-RoBERTa, BART, and GPT2 as candidates of contextualized models. Our experiments exhibit that all transformer-based models except GPT2 achieved close and high scores on text chunking. Due to the unique unidirectional architecture of GPT2, it shows a relatively poor performance on text chunking in comparison to other bidirectional transformer-based architectures. Our experiments also revealed that adding a LSTM layer to transformer-based models does not significantly improve the results since LSTM does not add additional features to assist the model to achieve more information from the input compared to the deep contextualized models.

Keywords: Text Chunking; Sequence Labeling; Contextualized Word Representation, Deep Learning; Transformers.

1- Introduction

Assigning appropriate labels to each particular token in text has always been a critical issue in Natural Language Processing (NLP). One of the most challenging tasks in sequence labeling is text chunking which entails detecting different phrases in unlabeled data. Finding distinct phrases in text could play an important role in various semantic and contextual analysis since each chunk usually contains a precious piece of information. This means that extracting the phrases from a particular corpus would endow the main idea and purpose of it. However, in spite of other sequence labeling tasks, detecting phrases in text is sometimes so sophisticated that it requires language experts' assistance.

Generally, phrases follow special semantic and syntactic patterns in text, which enables models to predict them automatically in large corpuses. To be more specific, linear statistical models like Hidden Markov Models (HMM) [10] or CRF [16] have been used frequently in sequence labeling tasks. There are many papers in previous years that combined CRF with more complicated models such as Long short-term memory (LSTM) [14] and Bidirectional LSTM (BiLSTM) [11]. Over the last couple of years, the emergence of transformer-based models has assisted researchers to take a huge step in handling various complicated NLP tasks more accurately and achieve state-of-the-art results on challenging datasets.

In this article, we employed state-of-the-art pretrained transformer-based models for sequence chunking. As these models are trained on a huge amount of textual data, they offer valuable contextual information that could be useful for various NLP tasks. We also use CRF and LSTM-CRF, a recent state-of-the-art model for sequence labeling tasks, after getting output from transformer-based models to evaluate the effect of classification module together with transformers. This research provides a comprehensive comparative study on the impact of transformer-based representations on chunking. Although different studies focused on text chunking, we still suffers from the lack of information from different perspectives: (1) the available models have not provided information about the differences between different representation models in text chunking, (2) in case of any difference, which type of representation performs the best is not clear, (3) in case of using any transformer-based representation model, it is not studied well if we still need the well-known LSTM-CRF architecture or we can benefit from a simpler module for classification.

In the rest of the paper, first, we introduce related works about sequence chunking in Section 2. Our model will be specified in Section 3. In Sections 4 and 5, we will

✉ **Saeedeh Momtazi**
momtazi@aut.ac.ir

elaborate the training procedure and experimental results. Finally, we will draw the conclusion in Section 6.

2- Related Works

Considering the sequential behavior of textual data, various NLP tasks, such as parts of speech tagging, named entity recognition and chunking work based on sequence labeling models.

Most traditional sequence labeling approaches use rudimentary language-specific methods [13].

Probabilistic graphical models, such as HMM and CRF have been widely used for these tasks. Different types of Recurrent Neural Networks (RNN) are more recent approaches that have played an important role in sequence labeling including chunking tasks. RNN was first designed by Rumelhard et al. [31] to carry the information of a sentence by passing through it. Hochreiter et al. [14] and Cho et al. [4] introduced more complicated versions of RNN, called LSTM and GRU, which enables the model to keep crucial information of certain tokens in sentence without the problem of vanishing or exploiting gradients occurred in previous RNNs. To benefit from the information of both sides of a sentence, BiLSTM was introduced by Graves et al. [11], enabling models to capture more information during training which achieved better results in the chunking task.

Ma et al. [23] and Huang et al. [15] used BiLSTM to obtain word representations with respect to both right and left context and a subsequent CRF layer to consider sentence level tag information. Huang et al. [15] also used SENNA [6] pre-trained embeddings. Moreover, in the proposed model of Ma et al. [23], a max-pooling and a convolutional layer were used to obtain character embeddings for each word. They also used the concatenation of character representations, linguistic features like POS and NER labels, and word embeddings to create a general embedding before feeding to BiLSTM.

An attention-based RNN was used for chunking by Li et al [19]. A context window was defined to generate embeddings and this assisted RNN to consider more local dependency. Attention component c_t helps to selectively obtain information from encoding layers instead of totally relying on a particular hidden state.

Attention segmental recurrent neural networks (ASRNN) was used by Lin et al. [20] for text chunking. The hierarchical architecture employed in their model helps the model to capture both character and word-level information from the text and also separate important and less information while building the segmental representation. Wei et al. [39] used a novel attention-based model called position-aware self-attention to extract both successive and discrete dependencies of each word in sequences.

There are many works employing semi-supervised and unsupervised learning to enhance their results. In the proposed model of Rei et al [30], a secondary unsupervised section for language modeling was used to enhance performance in sequence labeling tasks including chunking by learning more complicated features. Peters et al. [25] designed a bidirectional language model TagLM and trained it on unlabeled data. Then, they used pre-trained embeddings achieved from TagLM for the chunking task. Clark et al. [5] also proposed a semi-supervised approach working on both unlabeled and labeled data.

Wang et al. [38] employed a meta self-training approach for sequence labeling tasks in order to overcome the lack of annotated data challenge. Using meta self-training, they only needed to use a small amount of labeled data along a large unlabeled corpus, which helped their model to benefit from the information within the huge amount of unlabeled data.

Multi-task learning was another approach used by Liu et al. [21], Sogaard et al. [34], and Hashimoto et al. [13] to improve the performance of the model in chunking. Liu et al. [21] fine-tuned word-level pre-trained GLOVE [1] embeddings and also used character-level embeddings to build a language model alongside handling chunking. Sogaard et al. [34] used BiRNNs [33] for different sequence labeling tasks including shallow parsing by employing SENNA embeddings. Hashimoto et al. [13] devised a Joint Many-Task (JMT) model whose goal is handling different NLP tasks in a deep neural architecture. They also used two types of embeddings, Skip-gram [24] and character embeddings.

Zhai et al. [41] designed three models to handle segmentation and labeling tasks simultaneously. They also concatenated two embeddings: SENNA and embeddings gathered by adopting CNN on character embeddings of words to achieve final embeddings. Xin et al. [40] designed IntNet to learn the internal structure of words by their composing characters. Afterwards, in order to capture context information and handle sequence labeling and chunking, they feed these embeddings to LSTM-CRF.

Character-level language models have been also used to obtain highly contextualized word embeddings for sequence labeling tasks including chunking. After training the character-level language model, Akbik et al. [1] concatenated their own word embeddings to pre-trained GLOVE embeddings and passed them to a BiLSTM-CRF network.

Akhundov et al [2] combined byte embeddings extracted by Byte BiLSTM with word embeddings and fed the result to another BiLSTM to get word-level scores and use a CRF layer to handle sequence tags.

[1] http://nlp.stanford.edu/projects/glove/

Obviously, most well-known works on the chunking tasks are based on statistical models like CRF beside various kinds of recurrent neural networks such as LSTM, BiLSTM, as well as pre-trained static word embeddings which are appropriate to be fed into the aforementioned models.

Transformer-based models have also been studied in recent works in sequence labeling tasks. Although the proposed models by Tsai et al. [36] and Chawla et al. [3] benefit from state-of-the-art representation models, they only focused on one specific representation and a specific classification model. They did not compare the impact of different available models for each of the representation and classification components of text chunking. %However, the novel contextualized transformer-based models started with Vawsani et al. [37] which have revolutionized many NLP tasks in recent years have not been explored for chunking.

In the next section, we analyze the performance of the newest transformer-based pre-trained models in the chunking task by providing a comparative analysis on different models.

3- Models

Our approach consists of two major parts: (1) a pre-trained transformer-based model to capture contextual features of tokens, (2) a probabilistic graphical model and a neural network model, such as CRF and LSTM-CRF, which receives the output of pre-trained models to learn the labels and their dependency. In this part, we first briefly introduce the pre-trained models which build the former part of our architecture. Then we continue with the latter part.

3-1- Transformer-based Models

With the emergence of transformers [37], a great step has been taken in the NLP area for achieving outstanding results in different tasks. The unique architecture of transformers assists models to learn much more sophisticated contextual information from text and outperform embedding models like Word2Vec and ELMo [26], and other RNN based architectures like LSTM. Over the last couple of years, some special transformer-based architectures have been developed, which enhanced experimental results in various NLP tasks. We used those versions of transformer-based models that have between 300 to 500 million parameters in order to compare their performance in a relatively equal situation. Here, we describe the models that are used in our proposed architecture for shallow parsing.

3-1-1 BERT

Devlin et al. [9] designed a deep bidirectional transformers architecture pre-trained on a tremendous amount of unlabeled text, BookCorpus [42] and Wikipedia, for two specific tasks of masked language modeling and next sentence prediction. At the beginning stage, BERT sums three embeddings: token, segment, and position embeddings to create final input embeddings. It helps the model to consider the position of input tokens in sentence and the segmentation part of them before feeding them into the bidirectional transformer layers. In the masked language modeling pre-training task, the model is forced to predict some masked tokens in input sequence by the context of other inputs. Another pre-training task for BERT is next sentence prediction in which two sequential sentences were fed to the model and the model is expected to predict whether two sentences are contextually related or not. Pre-trained parameters could be fine-tuned for various NLP tasks such as sequence labeling by giving labeled data to the model.

3-1-2 XLM

Lample et al. [17] proposed a bidirectional transformer-based model trained on both supervised and unsupervised tasks. Like BERT, masked language modeling is the unsupervised objective for XLM which forces the model to predict some masked input tokens by considering other words in the sentence. Predicting the next token is another unsupervised task designed for XLM to be trained on monolingual data. By considering the previous input tokens in a sentence, the model is forced to predict the next token. They also trained the model by translation language modeling, a more flexible version of masked language modeling that uses multilingual parallel sentences in two different languages to predict masked tokens. For instance, to handle translation language modeling, the model could use the tokens in the French sentence to predict masked inputs in the English sentence. In this way, the model was forced to learn how to use translations for predicting masked words. XLM is another transformer-based model that encouraged us to be used for the shallow parsing task.

3-1-3 GPT2

GPT2 is a unidirectional transformer-based model first introduced by Radford et al. [27]. The purpose of GPT2 is predicting the next word which is called causal language modeling. Due to the special unidirectional architecture and causal language modeling task, GPT2 is an ideal model for text generation and predicting next words of a sentence by considering previous ones. The original GPT2 model is trained on 8 million web pages and contains 1.5 billion parameters. We chose the medium version of GPT2 as one of our transformer-based models to create appropriate embedding for each word. It should be mentioned that the medium version of GPT2 consists of 345 million parameters, which has roughly the same number of parameters that large versions of other transformer-based models have.

3-1-4 RoBERTa

RoBERTa is introduced by Liu et al. [22] to improve some aspects of BERT. The RoBERTa architecture is completely similar to BERT. However, Liu et al. [22] designed some different scenarios during the pre-training phase of RoBERTa. The next sentence prediction objective is removed from pre-training targets since they proved that eliminating next sentence prediction would result in improving the model downstream task performance. Masked language modeling is kept as a major pre-training objective for the model. RoBERTa was trained on a larger amount of data rather than BERT including CC-News[1], OpenWebText[2], and STORIES [35] in addition to Book Corpus [42] and English Wikipedia data. RoBERTa also has some differences in implementation. Liu et al. [22] used larger batches for RoBERTa than BERT. RoBERTa uses a different tokenizing scheme than BERT called byte-pair encoding which is similar to GPT2’s tokenizer. Similarities and differences between RoBERTa and BERT motivated us to opt RoBERTa as one of candidate transformer-based models in order to compare its performance on the chunking task with other pre-trained models specially BERT.

3-1-5 BART

Lewis et al. [18] proposed an architecture composed of a bidirectional transformer-based encoder (like BERT) and a unidirectional decoder (like GPT2). BART is pre-trained by five major tasks including masked language modeling, token deletion, token infilling, sentence permutation, and document rotation. In the masked language modeling task, like other models such as BERT, BART is forced to predict the masked words in the sentence. In the token deletion task, in contrast to masked language modeling, some tokens are deleted in a sentence and the model duty is to determine the positions of deleted tokens. Another innovative task, token infilling, is replacing spans of tokens with one single mask token and obliging the model to learn how many words are in that single masked token. Sequence permutation shuffles the sentences in the document and feeds them to the model. In the document rotation task, each time, a single token is randomly selected and the document is rotated in a way to set that particular token as the beginning token of the document. This task teaches the model to predict the starting of each document. Although the main goal of the BART model is text generation and its related tasks, other NLP tasks like sequence labeling and shallow parsing could be handled by employing BART as well. We decided to use the large BART as a candidate of state-of-the-art transformer-based models for our chunking task.

3-1-6 XLM-RoBERTa

XLM-RoBERTa was proposed by Conneau et al. [7]. They set masked language modeling for their model to predict masked tokens of the input. Lample et al. [17]'s model is employed to enhance the model in some particular issues. To be more specific, they designed the model to be multilingual and they trained it on 100 different languages, requiring a great amount of data. For this, they also built a huge dataset, CommonCrawl, containing 2 terabytes of text data from 100 languages. The similarities between XLM and XLM-RoBERTa and the huge multilingual data that it was trained on motivated us to include XLM-RoBERTa in our transformer-based models for text chunking.

3-1-7 Funnel Transformer

Funnel Transformer is a new-brand bidirectional transformer-based model proposed by Dai et al. [8]. It consists of two major parts, an encoder and a decoder. In encoder, there are pooling layers between transformer layers, reducing the size of the initial input and endowing lower computation cost to the whole model. In tasks like sentence summarization or text classification, using just the encoder part of the Funnel Transformer model could be sufficient. To handle token classification and sequence labeling tasks like chunking, designers added a decoder module to the model in order to resize the reduced input by upsampling from encoded layers. Similar to BERT, the objective of the Funnel Transformer is masked language modeling; i.e., the purpose of pre-training is predicting the masked tokens in input sequence. Consequently, this unique architecture makes Funnel Transformer an appropriate option for sequence labeling tasks.

3-2- Learning with LSTM and CRF

After receiving the outputs from transformer-based models, we fed them to a sole CRF layer or a sequence of LSTM-CRF layers. CRF considers past and future labels in a sequence to predict the label of a particular token. CRF computes the best possible tag sequence between all possible sequences by minimizing the objective function presented in Equation 1.

$$E = -s(y) + \sum_{\tilde{y} \in Y} e^{s(\tilde{y})} \qquad (1)$$

[1] http://commoncrawl.org/2016/10/newsdataset-available
[2] http://Skylion007.github.io/OpenWebTextCorpus

where Y represents all possible tag sequences and the goal is to find the sequence that minimizes the formula for a given input sequence.

CRF has two matrix parameters $A^{\{k \times k\}}$, $P^{\{n \times k\}}$, where k represents the number of tags and n is the length of each sequence. A is transition scores between different tags and P determines the probability of each tag in a position. The score of a tag sequence $y = \{y_1, \ldots, y_n\}$, $y_i \in \{1, \ldots, K\}$ and a given input sequence $x = \{x_1, \ldots, x_n\}$ is calculated based on Equation 2.

$$s(x\,,y) = \sum_{i=1}^{n} A_{y_i,y_{i+1}} + \sum_{i=1}^{n} P_{i,y_i} \qquad (2)$$

We also study another architecture, LSTM-CRF, which has been a great success story in sequence labeling tasks, such as parts of speech tagging, named entity recognition, and text chunking for several years [15,40,23,2].

We passed the outputs of transformer-based models to LSTM to observe the impact of combining recurrent neural models with state-of-the-art transformers.

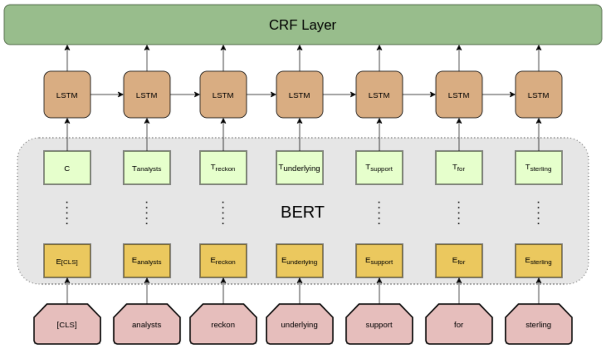

Fig. 1 BERT LSTM-CRF model. After passing through the BERT transformer-based model, token embeddings are given to a LSTM-CRF layer to detect their chunk labels.

Figure 1 shows the combination of one of our transformer-based models, BERT, with a LSTM-CRF layer. In the first stage, a pre-trained BERT model with multi-layer bidirectional transformer-based architecture converted tokens to fine input embeddings and passed them through the transformer layers. We feed the outputs of BERT to an LSTM layer to evaluate how the combination of LSTM and BERT could affect the quantity and the quality of information extracted from input. At the last layer, CRF is utilized to enhance the model performance in predicting output chunk labels by considering the dependency between chunk tags.

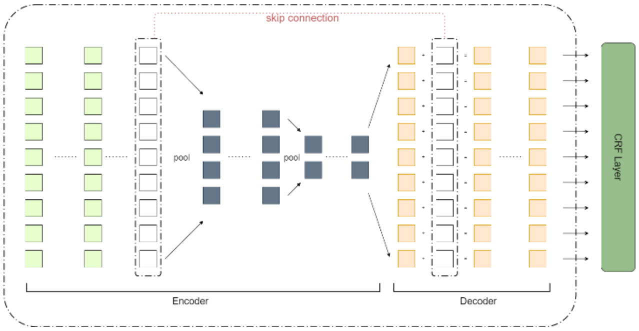

Fig. 2 Funnel Transformer CRF model. After giving tokens to an encoder-decoder transformer-based architecture, outputs will be given to a CRF layer in order to predict chunk labels.

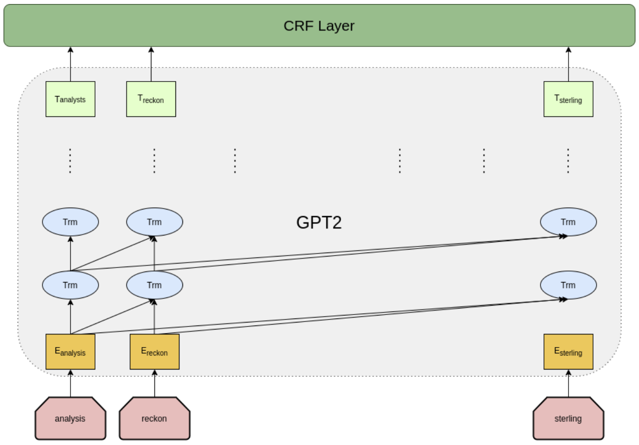

Fig. 3 GPT2 CRF model. Unidirectional transformer-based architecture of the GPT2 model caused a relatively poor performance against other transformer-based models BERT, XLM, and Funnel Transformer.

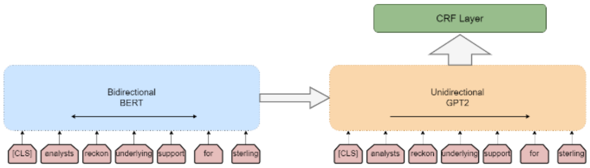

Fig. 4 Bart CRF model. Bart is composed of two parts, a bidirectional encoder (BERT) and an unidirectional decoder (GPT2). For fine-tuning, uncorrupted inputs should be given to both encoder and decoder and final hidden states of the decoder should be considered as output.

In Figure 2, the encoder-decoder architecture of Funnel Transformer is shown. By passing through the encoder section, pooling layers gradually reduce the dimension of transformer layers to encode the information in relatively smaller size. The decoder part resizes the encoded information to their original size, which makes the model appropriate for token classification tasks such as chunking and named entity recognition. At the final stage, we use a CRF layer for predicting target labels.

Figure 3 shows the combination of Open AI GPT2 model with a CRF layer to predict chunk sequences. The unidirectional transformer-based architecture of GPT2 is also shown in the diagram. Although this characteristic has made GPT2 a great smart in text generation, we will see in the next section how this architecture affects the performance of the model in chunking.

In Figure 4, BART-CRF model is shown. Two major parts of BART, bidirectional encoder and unidirectional decoder, are very important during pre-training of the model. To handle the span masking objective, corrupted inputs were fed to the encoder (BERT) and uncorrupted to the decoder (GPT2). Then, BART had to learn masked spans by the information that bidirectional and unidirectional architectures of BERT and GPT2 represent.

4- Experiments

4-1- Dataset

We evaluate our approaches on the standard chunking dataset from CoNLL2000 [32] which contains 11 distinct phrase types. The detailed statistics of the dataset is presented in Table 1. Chunk labels in CoNLL2000 are based on the IOB scheme introduced by Ratnaparkhi et al. [29]. In the next section, we explain how we changed the IOB scheme to IOBES [28]. In Table 2, we present one example of each type of phrase existing in CoNLL2000.

Table 1: Size of sentences, tokens, and labels in CoNLL2000 datasets

train	# of sentences	8936
	# of tokens	211727
test	# of sentences	2012
	# of tokens	47377
	# of labels	22

Table 2: One phrase example for every label existing in the CoNLL2000 dataset.

Phrase	***Label***
their current 15% level	B-NP I-NP I-NP I-NP
has been eroded	B-VP I-VP I-VP
because of	B-PP I-PP
even though	B-SBAR I-SBAR
similar and conservative	B-ADJP I-ADJP I-ADJP
at least	B-ADVP I-ADVP
have to serve	B-VP I-VP I-VP
Good morning	B-INTJ I-INTJ
as well as	B-CONJP I-CONJP I-CONJP

4-2- Setup of Experiments

Due to the many reported experiments, replacing the IOB scheme with IOBES would yield better final accuracy [25,19,2]. As a result, we decided to use the IOBES scheme to evaluate the impact of transformer-based models for sequence chunking. In addition to regular Outside (O), Inside (I), and Beginning (B) labels, we added two End (E) and Single (S) tags, which contain more detailed information in terms of the labeling scheme. To this aim, in the IOB format, in case of having B label with no I label, we converted the B label to S. Also the last I label in each chunk is converted to E label.

For all experiments, we trained our model on the train set and reported the final results on the test set. Since CoNLL2000 does not give an explicit validation set, we randomly selected 10% of the sentences from train data as our validation set.

To employ pre-trained models and build CRF and LSTM-CRF layers, we used the transformers package [1] and Pytorch[2]. Due to the purpose of BERT, XLM and Funnel Transformer pre-trained models, TokenClassification module has been already implemented for these models and we used it to generate embeddings of input tokens. We also manually add the TokenClassification module to GPT2 and BART models to see their performance on the chunking task.

We set the probability of the dropout layer to 0.3, 0.4, 0.5 and injected it between the pre-trained model and target CRF or LSTM-CRF layer. Adam optimizer was used for training parameters of CRF and LSTM-CRF layers and fine-tuning the parameters of pre-trained transformer-based models. Learning rate was fixed in 1e-5 during the training process. Max length of sentences was fixed to 110 and batch-sizes were set to 4, 8, 16 regarding the pre-trained model used to split data during training.

[1] https://huggingface.co/

[2] https://pytorch.org/

4-3- Experimental Results

Table 3 presents the results of all described models. Our BERT-CRF model achieved 96.72 F1 score and BERT-LSTM-CRF received 96.70 F1 score. RoBERTa's performance on shallow parsing was slightly better. RoBERTa-CRF model's F1 score is 96.82 and RoBERTa-LSTM-CRF layer achieved 96.84. The performance of XLM on the chunking task outperformed our BERT-based models despite their similar bidirectional transformer-based architecture except XLM's translation language modeling objective during pre-training. XLM-CRF and XLM-LSTM-CRF models both achieved 96.83 F1 scores.

Table 3: One phrase example for every label existing in the CoNLL2000 dataset.

Pre-trained Models CRF LSTM-CRF	***CRF***	***LSTM-CRF***	***Dense Layer***
BERT	96.72	96.70	96.07
RoBERTa	96.82	96.84	96.34
XLM	96.83	96.83	96.50
XLM-RoBERTa	**96.92**	**96.92**	96.62
Funnel Transformer	96.83	96.53	96.50
BART	96.30	96.03	93.58
GPT2	85.00	84.46	82.57

XLM-RoBERTa achieved the highest scores between our transformer-based models. Both XLM-RoBERTa-CRF and XLM-RoBERTa-LSTM-CRF models received 96.92 F1 score. Funnel Transformer with its unique encoder-decoder transformer-based architecture achieved 96.83 F1 score in combination with CRF layer. The Funnel-LSTM-CRF model achieved 96.53 F1 score on IOBES mode.

The worst performance is reported based on GPT2. Due to the unidirectional architecture of GPT2, the model is adapted to learn left-to-right context of the text and its main goal is for text generation tasks; while in sequence labeling models, need to obtain left-to-right and right-to-left contextual information in order to achieve an acceptable result. The GPT2-CRF and the GPT2-LSTM-CRF models achieved 85 and 84.46 F1 scores, respectively.

Likewise, the unidirectional architecture of BART's decoder results in a relatively lower F1 score regarding bidirectional transformer-based models BERT, RoBERTa, XLM, XLM-RoBERTa, and Funnel Transformer. The results of BART, however, are significantly better than GPT2. The BART-CRF model achieved 96.30 and the BART-LSTM-CRF model result is 96.03.

We also tried a third possible architecture with a dense layer at the final stage. In this scenario, BERT achieved 96.07, RoBERTa's score is 96.34 and both Funnel Transformer and XLM received 96.50 and XLM-RoBERTa achieved 96.62, higher than others in this mode. BART achieved 93.58 and GPT2's score is 82.57.

As we expected, adding a LSTM layer to our CRF target layer not only did not improve our results, but caused them to drop in most cases. Since contextualized transformer-based models are stronger in extracting context information of text, adding a LSTM layer could not improve the ability of our model in capturing more information. In other words, the LSTM layer models similar information as transformers and it does not capture additional information compared to transformers. Moreover, considering the more advanced architecture of transformers in sequence modeling, using an LSTM network besides a transformer does not provide any benefit in the architecture. On the other hand, the CRF layer captures another type of information from the sequences of words by considering the relation between labels. As a result, models with just one CRF layer mostly outperformed their LSTM-CRF counterparts. Better results of the models with CRF in comparison with the models with just a dense layer in output was another expected observation as CRF has valuable information about the dependencies between tags which is not available in the other parts of the architecture.

In the next step, we compared our results with the state-of-the-art models in the literature as presented in Table 4. As can be seen, our model outperforms state-of-the-art models in the field, except the proposed model by Clark et al. [5] which is marginally better than ours. The main reason is that this model works based on multi-task learning and benefits from training data from other sequence modeling approaches in text chunking.

Overall, by comparing various transformer-based models and different architectures that are used after the embedding part, the following observations from the reported results are notable:

- Using novel transformer-based pre-trained models enhances the overall F1 score on text chunking due to the fact that they endow the precious information that they had learned by being trained on a huge amount of data to the model.
- Different pre-trained models which benefit from different architectures (except GPT2) have close performances on chunking and it is difficult to define an absolute winner.
- Despite other transformer-based models, GPT2's performance was relatively poor on text chunking, due to the fact that although the unidirectional architecture of GPT2 has made it a perfect model for text generation, it causes GPT2

to achieve lower results because predicting chunk labels require information from both sides of a particular token.

- A CRF layer in the final stage assists the contextualized pre-trained embedding models to consider the labels of other tokens when they are predicting the label of each particular word and enhances the overall performance.
- In the models in which contextualized pre-trained models were used, adding an LSTM layer does not improve the performance of the model, which could imply that removing LSTM could be a wiser decision since it does not help the model to capture more from the input tokens.

Overall the best results achieved by the XLM-RoBERTa model followed by a CRF layer, which achieved superior results compared to the state-of-the-art models in the field.

Table 4: One phrase example for every label existing in the CoNLL2000 dataset.

Model	***F1 score***
Collobert et al. [2011]	94.32
Huang et al. [2015]	94.46
S_gaard and Goldberg [2016]	95.28
Rei [2017]	93.88
Zhai et al. [2017]	94.72
Liu et al. [2017]	95.96
Peters et al. [2017]	96.37
Xin et al. [2018]	95.29
Akbik et al. [2018]	96.72
Clark et al. [2018]	97.00
Akhundov et al. [2018]	94.74
Lin et al. [2021]	93.70
Wei et al. [2021]	95.15
Ours: XLM-RoBERTa + CRF	96.92

5- Conclusions

We provided various architectures based on state-of-the-art transformer-based models for the chunking task. Well-known CoNLL2000 dataset was used for evaluating our models by F1 score. We compared our models' performance with other works done for sequence labeling tasks. Most previous models employed different types of RNNs such as LSTM and BiLSTM, CNN and character and word embeddings. We used novel transformer architectures like Funnel Transformer, XLM-RoBERTa, and BART as well as BERT, RoBERTa, XLM, and GPT2 to evaluate the effect of contextual embeddings and the combination of them with CRF and LSTM-CRF layers in shallow parsing.

In future, we will use the transformer-based models for other similar sequence labeling tasks like name entity recognition and parts of speech tagging. Due to the similarity of most sequence labeling tasks and the models that have been proposed for them, combining transformer-based models with previous state-of-the-art models could yield significant performance specially on rare languages.

References

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649, 2018.

[2] Adnan Akhundov, Dietrich Trautmann, and Georg Groh. Sequence labeling: A practical approach. arXiv preprint arXiv:1808.03926, 2018.

[3] Avi Chawla, Nidhi Mulay, Vikas Bishnoi, and Gaurav Dhama. Improving the performance of transformer context encoders for ner. In 2021 IEEE 24th International Conference on Information Fusion (FUSION), pages 1–8. IEEE, 2021.

[4] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014.

[5] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. arXiv preprint arXiv:1809.08370, 2018.

[6] Ronan Collobert, Jason Weston, L´eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE):2493–2537, 2011.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116, 2019.

[8] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. arXiv preprint arXiv:2006.03236, 2020.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[10] Sean R Eddy. Hidden markov models. Current opinion in structural biology, 6(3):361–365, 1996.

[11] Alex Graves and Jurgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks, 18(5-6):602–610, 2005.

[12] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. arXiv preprint arXiv:1611.01587, 2016.

[13] Zhiyong He, Zanbo Wang, Wei Wei, Shanshan Feng, Xianling Mao, and Sheng Jiang. A survey on recent advances in sequence labeling from deep learning models. arXiv preprint arXiv:2011.06727, 2020.

[14] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[15] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991, 2015.

[16] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[17] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291, 2019.

[18] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461, 2019.

[19] Bofang Li, Tao Liu, Zhe Zhao, and Xiaoyong Du. Attention-based recurrent neural network for sequence labeling. In Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, pages 340–348. Springer, 2018.

[20] Jerry Chun-Wei Lin, Yinan Shao, Youcef Djenouri, and Unil Yun. Asrnn: a recurrent neural network with an attention model for sequence labeling. Knowledge-Based Systems, 212:106548, 2021.

[21] Liyuan Liu, Jingbo Shang, Frank F Xu, Xiang Ren, Huan Gui, Jian Peng, and Jiawei Han. Empower sequence labeling with task-aware neural language model. arXiv preprint arXiv:1709.04109, 2017.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[23] Chunping Ma, Huafei Zheng, Pengjun Xie, Chen Li, Linlin Li, and Luo Si. Dm nlp at semeval-2018 task 8: neural sequence labeling with linguistic features. In Proceedings of The 12th International Workshop on Semantic Evaluation, pages 707–711, 2018.

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.

[25] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108, 2017.

[26] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

[28] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), pages 147–155, 2009.

[29] Adwait Ratnaparkhi. A linear observed time statistical parser based on maximum entropy models. arXiv preprint cmp-lg/9706014, 1997.

[30] Marek Rei. Semi-supervised multitask learning for sequence labeling. arXiv preprint arXiv:1704.07156, 2017.

[31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. nature, 323(6088):533–536, 1986.

[32] Erik F Sang and Sabine Buchholz. Introduction to the conll-2000 shared task: Chunking. arXiv preprint cs/0009008, 2000.

[33] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing , 45(11):2673–2681, 1997.

[34] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 231–235, 2016.

[35] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847, 2018.

[36] Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. Small and practical bert models for sequence labeling. arXiv preprint arXiv:1909.00100, 2019.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[38] Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. Meta self-training for few-shot neural sequence labeling. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 1737–1747, 2021.

[39] Wei Wei, Zanbo Wang, Xianling Mao, Guangyou Zhou, Pan Zhou, and Sheng Jiang. Position-aware self-attention based neural sequence labeling. Pattern Recognition, 110:107636, 2021.

[40] Yingwei Xin, Ethan Hart, Vibhuti Mahajan, and Jean-David Ruvini. Learning better internal structure of words for sequence labeling. arXiv preprint arXiv:1810.12443, 2018.

[41] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. arXiv preprint arXiv:1701.04027, 2017.

[42] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27, 2015.