The Effects of Differing Syntactic Structures on the Performance of Neural Networks in a Named Entity Recognition Task

Matthew Buzzell
matabuzz@iu.edu

Natasha Singh
singhnat@iu.edu

Indiana University
Bloomington, IN

Abstract

In this paper we show that using syntactic structures of training data on neural models can improve the efficacy of said models on Named Entity Recognition tasks. We also show how these different syntactic models predict sub-tags differently, allowing for the use of specific syntactic models for specific sub-tag recognition tasks. Before using a syntactic model to train a neural model it is important to ensure that the specific syntactic model being used is beneficial to the task the neural model is performing. This criteria is also important when considering other languages with their own structure and rules.

1 Introduction

The aim of a Named Entity Recognition (NER) task is to identify the partition of text which has instances of named entities, such as an individual's name, geographic location, age, address, phone number, or company. Although many researchers disagree on the definition of named entities, it is still one of the most researched topics today. The term "named entity" was coined in the 6th MUC conference which focused on Information Retrieval tasks to refer to "unique identifiers of entities" [4]. Petasis defined it as a proper noun, serving as a name for something or someone [5]. NER has gained a lot of interest from researchers because it can be applied to various domains. One can use it to extract information from large databases of unstructured text or to improve sentiment analysis by identifying named entities that are associated with positive or negative sentiments.

2 Related Work

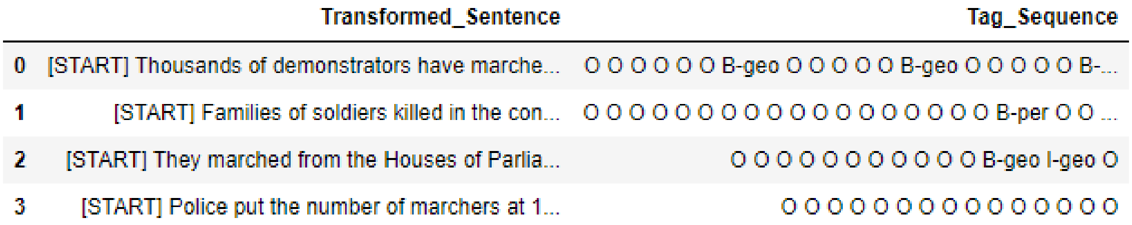
Previous works in this field include Lample et. al. (2016) [6], who noticed that the early SOTA models for NER tasks relied heavily on manually created features and domain knowledge to learn from a small dataset. In their paper, they implemented two neural architectures that use no language specific resource: (i) bidirectional LSTM and CRF's (ii) a transition-based approach inspired by shift-reduce parsers for four languages - English, Dutch, German, and Spanish. Their LSTM-CRF models outperformed all the previous SOTA models in Spanish and German with an F1-score of 85.75 and 78.76 respectively[1].

Following this, Hongyu Lin et. al (2019) [3], highlighted the limitation of sequence labeling-based NER tasks that restricts a word to only one category. In their paper, they tried addressing this issue by leveraging the head-driven phrase structures of entity mentions. To do this, they experimented with three standard English entity mention detection benchmarks (ACE2005, GENIA and TACKBP2017). Their model was able to achieve the highest F1-score of 74.9 and 74.6 on ACE2005 and TACKBP2017 dataset respectively. For the GENIA dataset, their model had an F1-score of 72.3 while the model by Wang and Lu (2018) outperformed them by achieving an F1-score of 75.1.

In a tangentially related domain, Juntao Yu et. al. (2020) [1], stated that most of the research on NER is focused on flat NER and ignored the nested entities like [Bank of [China]] where both [China] and [Bank of China] are named entities. In their paper, they leveraged Dependency Grammar to predict heads and assign relations to head-child pairs. Borrowing the idea of Machine Reading Comprehension, they formulated the NER task to identify the start and end positions of the named entity text spans. They stacked the Biaffine model on top of a Bi-LSTM to assign scores to all possible spans in a sentence. Their model outperformed all the existing models for both nested and flat NER tasks.

To build upon these ideas, Lu Xu et. al (2021) [2] hypothesized that incorporating long-distance

[1] Whether these F1-score were micro-averaged or macro-avaeraged was not states withing these papers.



	Transformed_Sentence	Tag_Sequence
0	[START] Thousands of demonstrators have marche...	O O O O O O B-geo O O O O O B-geo O O O O O B-...
1	[START] Families of soldiers killed in the con...	O O O O O O O O O O O O O O O O O O B-per O O ...
2	[START] They marched from the Houses of Parlia...	O O O O O O O O O O O B-geo I-geo O
3	[START] Police put the number of marchers at 1...	O O O O O O O O O O O O O O O

Figure 1: The BIO tags sequence we extracted from the Kaggle NER dataset.

information can improve the performance of NER models because the contextual information captured by the linear sequences and the structured information captured by the dependency trees complements each other. In their paper, they presented the SynergizedLSTM (Syn-LSTM) model which captured the interaction between the contextual and structural features. Their model achieved the highest precision, recall, and F1-score in the SemEval 2010 task for both Catalan and Spanish.

3 Methodology

3.1 Data

For our project, we used the Kaggle NER[2] dataset for English. From this dataset we extracted the words and BIO tags for each sentence as shown in Figure 1.

The data currently has main tags as Beginning of named entity (B), Inside of named entity (I) and Outside of named entity (O). The data also has subtags as Time entity (tim), Geopolitical entity (gpe), Geographic entity (geo), Economic Value Equity entity (eve), Organizational entity (org), and Person (per).

3.2 Parsing

We generated syntactic information for each sentence using Dependency Grammar (DG) and Head-Driven Phrase Structure (HPSG) Grammar. To generate the syntactic information for DG, we used the StanfordDependencyParser[3], which is embedded in the NLTK module, and for the syntactic information for HPSG, we used the DELPH-IN[4] module.

3.3 Data Transformation

The transformed input sequence had the following structure:

[START] sentence [SEP] syntactic info [END] [PAD]

Syntactic information generated by DG had the head-dependency relationship information of each pair of words in the sentence, while the syntactic information generated by HPSG had information about each lexicon (Tense, Person, Singular/Plural, what predicates it takes as input etc.).

Unique words from the input sequence data are collected to form a vocabulary. Then, several special tokens ([START], [END], [SEP], [UNK] and [PAD]) are added to the vocabulary. The [START] and [END] tokens are used to mark the start and end of the sequence respectively. The [SEP] token is used to separate the original sentence from the syntactic information. The [UNK] token is used to mask any word in the test data that is not present in the vocabulary. And finally, the [PAD] token is used to add additional dummy words to an input sentence to make all the sentences of equal length.

After the special tokens are added, the vocabulary is then used to encode the words in each input sentence (each word is encoded as the index for that word in the vocabulary). All unknown words are mapped to the index of the [UNK] token. Finally, all the sentences are extended by adding the index for [PAD] tokens to the end of each sentence until they are the same length as the sentence with the maximum number of words.

3.4 Model Architecture

We implemented a simple Recurrent Neural Network by stacking an Embedding layer, two bidirectional LSTM layers, and a one-time distributed Dense layer with a softmax activation function to generate BIO labels for the input sequence.

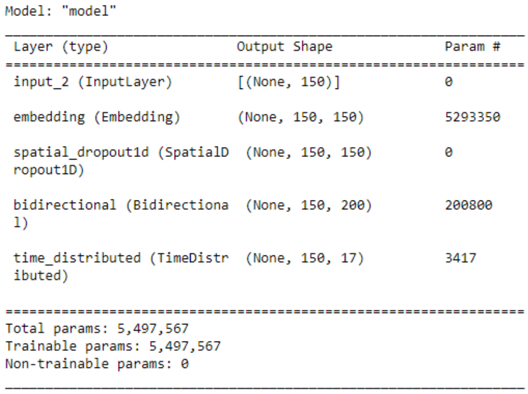
[2] https://www.kaggle.com/datasets/namanj27/ner-dataset

[3] https://stanfordnlp.github.io/CoreNLP/download.html

[4] https://github.com/delph-in

	Baseline (POS)	**DG**	**HPSG**
Model accuracy	96.68%	98.43%	95.47%
Exact match accuracy	15.24%	42.06%	17.05%
Accuracy in predicting B tag	3.2%	99.88%	56.32%
Accuracy in predicting I tag	0%	38.2%	0%
Accuracy in predicting O tag	100%	99.83%	48.87%
Accuracy in predicting tim sub-tag	52%	36.12%	51.06%
Accuracy in predicting gpe sub-tag	0%	40.13%	0%
Accuracy in predicting geo sub-tag	0%	99.84%	32.5%
Accuracy in predicting eve sub-tag	0%	5.36%	0%
Accuracy in predicting org sub-tag	76%	85%	56.32%
Accuracy in predicting per sub-tag	84%	100%	54.5%

Table 1: Percent accuracy across different metrics for model trained using differing syntactic information.



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 150)]             0

 embedding (Embedding)       (None, 150, 150)          5293350

 spatial_dropout1d (SpatialD  (None, 150, 150)         0
 ropout1D)

 bidirectional (Bidirectiona  (None, 150, 200)         200800
 l)

 time_distributed (TimeDistr  (None, 150, 17)          3417
 ibuted)

=================================================================
Total params: 5,497,567
Trainable params: 5,497,567
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: The parameters of our RNN model.

The loss function and optimizer used for the training was Binary Cross entropy and Adam respectively.

3.5 Evaluation metric for BIO labels

Simply evaluating based on accuracy does not make sense in this task because there are very few named entities in each sentence. Thus a very small fraction of the tags will be B or I and the majority will be O's. Given this, our model can learn nothing and still achieve a high accuracy by predicting all the tags as O's.

We thought that a better metric for evaluation would be to calculate how many times our model correctly identifies the main tag (i.e. a B, an I, or an O tag). On top of this, we also analyzed how many times our model correctly identified the correct subcategory of the entity (i.e. tim, gpe, geo, eve, org, or per).

4 Results

In Table 1, we show that using the model using the DG syntactic structure improves the model's overall accuracy when compared to our baseline model which was only trained using the Part of Speech (POS) tags for each word. On the other hand, using the model using the HPSG syntactic structure did not improve the model's overall accuracy. However, both the model using DG syntactic structures and the model using HPSG syntactic structures outperformed the baseline model in terms of exact match accuracy (the metric for when all the tags in an output sequence match the tags of the expected sequence exactly).

In terms of the main tags, both the model using DG syntactic structures and the model using HPSG syntactic structures received the same accuracy or outperformed the baseline model on both B-tags and I-tags. However, the baseline model outperformed the model using DG syntactic structures and the model using HPSG syntactic structures on O-tags. Additionally, the model using the DG syntactic structure outperformed the model using HPSG syntactic structure across all three main tags.

Finally, in terms of sub-tags, the model using the DG syntactic structure outperformed the baseline and the model using the HPSG syntactic structure across all sub-tags excluding the Time entity sub-tag (tim). On the other hand, the model using the HPSG syntactic structure received the same accuracy or underperformed against the baseline across all sub-tags except the Geographical entity sub-tag (geo). When compared against the model using the DG syntactic structure, the model using the HPSG syntactic structure underperformed across all sub-tags except the Time entity sub-tag (tim).

5 Conclusion

In this project we demonstrated that the use of syntactic information boosts the performance of the Named Entity Recognition model significantly in terms of correctly identifying the B, I and O main labels. We also illustrated that the model accuracy is not the correct metric to evaluate the performance of the model for the NER task. Further, we investigated the model's ability to accurately identify the category of a given named entity. We conclude that using syntactic information is better at identifying the main tags as well as sub-tags when compared to models only using POS tags.

6 Future Work

In the future, we want to experiment with other syntactic parsers like Lexical Functional Grammar. In addition, we want to investigate if this framework is able to perform better on the standard dataset for nested NER tasks. And most importantly, we would like to study the effects of these different syntactic frameworks on differing languages.

7 References

1. Yu, Juntao, Bernd Bohnet, and Massimo Poesio. "Named entity recognition as dependency parsing." arXiv preprint arXiv:2005.07150 (2020).

2. Xu, Lu, et al. "Better feature integration for named entity recognition." arXiv preprint arXiv:2104.05316 (2021).

3. Lin, Hongyu, et al. "Sequence-to-nuggets: Nested entity mention detection via anchor-region networks." arXiv preprint arXiv:1906.03783 (2019).

4. Chinchor, Nancy, and Patricia Robinson. "MUC-7 named entity task definition." Proceedings of the 7th Conference on Message Understanding. Vol. 29. 1997.

5. Petasis, Georgios, et al. "Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods." Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. 2000.

6. Lample, Guillaume, et al. "Neural architectures for named entity recognition." arXiv preprint arXiv:1603.01360 (2016).