Named Entity Recognition for Fictional, Nonfiction Texts with Different Approaches

Hüseyin Yağız Devre

Abstract:

Natural Language Processing is a part of Machine Learning, Artificial Intelligence, and computer science that works with words and sentences as well as languages, corpus instead of tables and images like other branches of Artificial Intelligence. Named Entity Recognition (NER) is one of the fundamental branches of Natural Language Processing(NLP). Named Entity Tagging aims to analyze texts and classify the words entities as some categories including Person(Per), Location(Loc), Organization(Org), Miscellaneous(Misc) etc. and encode those entities in Before(B), Inside(Inside), No Chunk(O), End(E), Single(S) or BIOES for short. Named Entity Recognition can be used in different parts of the Natural Language Processing tasks such as Sentiment Analysis, Speech Recognition, Machine Translations, etc..

This paper focuses on the performance of three learning methods, Global Context Enhanced Deep Transition Architecture (GCDT) without BERT embedding, Flair, Bidirectional Long Short Term Memory (LSTM) + Convolutional Neural Networks (CNN) and Adapting Transformer Encoder for Named Entity Recognition Architecture (TENER) in Fictional, Nonfictional and fantasy corpus. All of these algorithms are trained on the same data, Conll 2003 corpus, and tested on the same data collected from Wikipedia Articles for Nonfictional Data, Wikia Articles for Fictional Data. This paper tests the state of art Named Entity Recognition Algorithms and suggests a new and better hybrid algorithm for named entity recognition for both Fictional and Nonfictional texts.

Keywords:

Named Entity Recognition, Natural Language Processing, Global Context Enhanced Deep Translation, BERT, Flair, Conll 2003, Fictional, Nonfictional, Wikipedia, Wikia

1. Introduction:

What is Named Entity Tagging? To answer this question another question should be answered first. What is language? Or how do humans know that the word “Jack” is a name but the word “apple” is a fruit? The answer to the first question is obvious. A language is an essential tool for communication between humans composed of different sounds and words having meaning. This is the answer to the second question: Meaning. If it is about the meaning then, what are words, and how does a machine learn the meaning of a word and a sentence? The answer is simple. Machine Learning and lots of data. Natural Language Processing (NLP). NLP is a part of Machine Learning that works with words and sentences as well as languages and corpus instead of tables and images like other branches of Artificial Intelligence. The machine is given a huge dataset, annotated (Supervised), or unannotated (Unsupervised) depending on the learning method.

Figure 1: *An Example of Named Entity Recognition*

Named Entity Recognition(NER, or entity extraction) is a part of Natural Language Processing that aims to label and identify the elements of a sentence; Enamex, Numex, and Timex. Enamex is the term for organizations, names, and places; Numex is the term for numbers and Timex is the term for time and dates. The machine is identifying Person(Per), Location(Loc), Organization(Org), etc. and encode those entities in Before(B), Inside(Inside), No Chunk(O), End(E), Single(S) or BIOES for short regarding their positions and usages in the sentence. Named Entity Recognition is a tool that can be used for Text to Speech and Speech to Text algorithms, Machine Translations, and in Sentiment Analysis.

NER tasks require different algorithms for better results. With the recent technological breakthroughs, these algorithms improved significantly and got %93.5 accuracy These algorithms Even though these algorithms are trained on large datasets and powerful computers, some of the algorithms get lower accuracy scores on a given text from a different domain such as fictional and fantasy texts. Thus it can be said that Fictional and Fantasy domains are one of the shortcomings in current Named Entity Recognition algorithms and architectures and need more comprehensive yet adaptive architectures and models for the classification and identification for all domains. To accomplish this ideal machine learning model, a brand-new and possibly a hybrid model should be trained and tested using the current state of art models and corpus.

2. Related Work:

a. Natural Language Processing And Machine Learning:

Currently, Machine Learning is one of the rapidly evolving and enlarging fields of Computer Science. With the newest innovations in science and technology, the algorithms improved significantly in the tasks compared to the algorithms developed 50 years ago like Perceptron which is a type of classification algorithm. Even though these models for instance self-play reinforcement learning models achieve a superhuman level on specific tasks such as chess and go, they are inefficient and mostly worse on other tasks. These algorithms are called Artificial Narrow Intelligence. Researches all around the globe are trying to achieve a goal of Artificial General Intelligence. Similar to Machine Learning models Natural Language Processing models improved significantly.

b. Named Entity Recognition:

As previously stated, Named Entity Recognition is a subdirectory, yet one of the fundamentals of the Natural Language Processing. It is one of the building blocks of Natural language processing thus used in several applications and implications of Natural Language Processing. The most recent and the state of art named entity recognition models depend on Recurrent Neural Networks(RNN), Hybrid Bidirectional Long Short-Term Memory (LSTM), Convolutional Neural Networks, Perceptron models, and architectures. Even though these models work accurately some of them such as Recurrent Neural Networks have some problems. These models using RNN have shallow connections between the "consecutive hidden states of RNNs". Moreover, Recurrent Neural Network architectures have insufficient and inefficient modeling of global information. Not only the architectures and models have problems but also the data given is a problem at the fundamental level as well such as Figure 2.

SOCCER - [PER BLINKER] BAN LIFTED . [LOC LONDON] 1996-12-06 [MISC Dutch] forward [PER Reggie Blinker] had his indefinite suspension lifted by [ORG FIFA] on Friday and was set to make his [ORG Sheffield Wednesday] comeback against [ORG Liverpool] on Saturday . [PER Blinker] missed his club's last two games after [ORG FIFA] slapped a worldwide ban on him for appearing to sign contracts for both [ORG Wednesday] and [ORG Udinese] while he was playing for [ORG Feyenoord].

Figure 2: *An Example of Problematic Data for Named Entity Recognition*

In this particular instance, it is not obvious that Wednesday is an Enamex(Organization) or Timex. Similarly, it is not clear that Blinker is an Enamex(Person) in this instance. This suggests that external and prior knowledge is needed to make accurate predictions in every situation and datasets. Similar to this instance, the current Named Entity Recognition models need to be trained in various domains such as fantasy and fiction apart from the nonfiction domains used in most of the training and evaluations of various networks.

Named entity Recognition is commonly referred to as a sequential labeling and prediction task. Similar to the normal machine learning algorithms, in named entity recognition there are features, words and outputs so it can be written as x and y. A sequential labeling task can be written like this:

$$x = (x_1,\ x_2,\ x_3,\ x_4 \ .... \ x_N)$$

Similarly, y can be defined as follows:

$$y = (y_1,\ y_2,\ y_3,\ y_4 \ .... \ y_N)$$

So the sequence labeling task becomes as follows, the estimate of probabilities:

$$P(y_i \mid x_{i-k} \ ... \ x_{i-l} \ , \ y_{i-m} \cdots y_{i-1})$$

In this form; k, l, and m are very small numbers that are used to prevent overfitting, similar to the bias term used in the perceptron algorithm. In this situation y_{i-1} is the previous prediction and y_{i-2} is the prediction before that and x_i is the current word and the tokens are:

$$c = (x_{i-2},\ x_{i-1},\ x_i,\ x_{i+1},\ x_{i+2})$$

c. Algorithms:

In this section, some of the states of art algorithms will be explained. These algorithms are all trained on the same dataset Conll 2003 which is one of the widely used datasets in the Named Entity Extraction Tasks.

i. Background in Global Context Enhanced Deep Transition (GCDT) Architecture:

The Global Context Enhanced Deep Translation Architecture GCDT is a type of architecture for sequence labeling presented by the researchers Beijing Jiaotong University. Unlike the RecurrentNeural Networks used in several other algorithms, this architecture uses a different approach as follows:

Similar to the other algorithms the tokens in GCDT can be represented as $x = (x_1, x_2, x_3, x_4 .... x_N)$. On the contrary, the tokens are dependent on the embedding as $x_t = [c_t ; w_t ; g]$ where c is the character level word embedding, w is the pre-trained word embedding and g is the global contextual embedding.

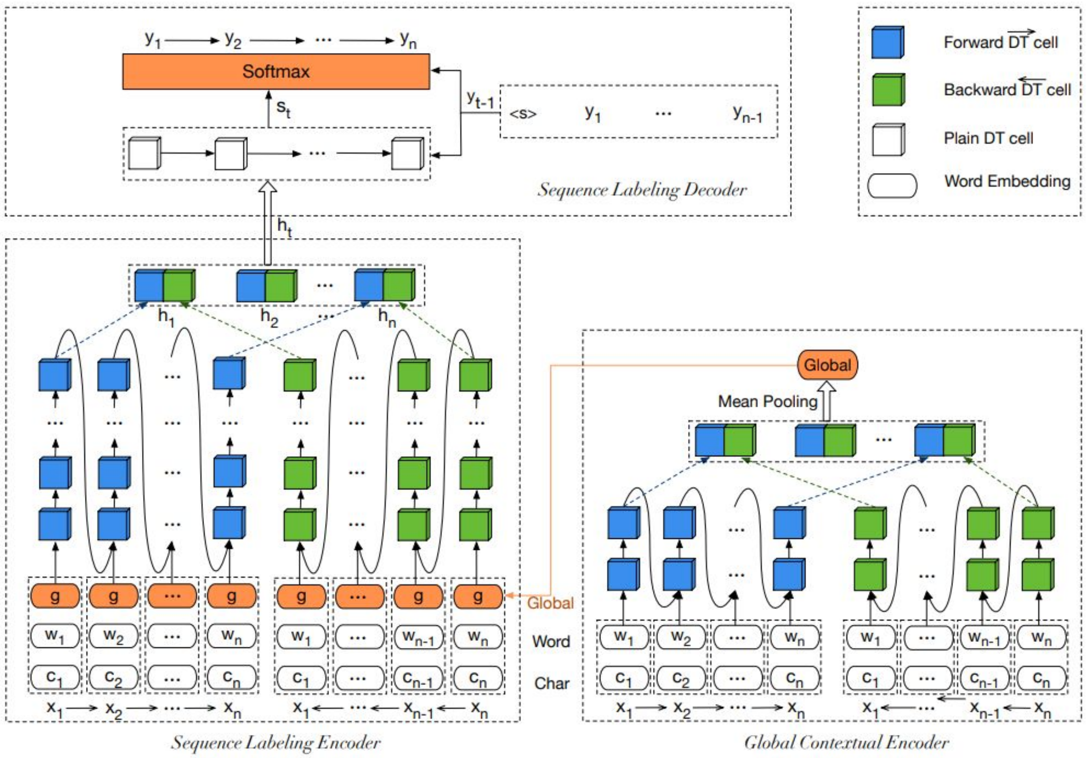

Figure 3: *An Overview of Global Context Enhanced Deep Transition Architecture*

g, the global contextual embedding is calculated by the mean pooling of the hidden states as follows: $\{h_1^g, h_2^g, \cdots, h_N^g\}$

Thus g cana be calculated as:

$$\mathbf{g} = \frac{1}{N}\sum_{t=1}^{n}\mathbf{h}_t^g$$

$$\mathbf{h}_t^g = [\overrightarrow{\mathbf{h}}_t^g ; \overleftarrow{\mathbf{h}}_t^g]$$

$$\overrightarrow{\mathbf{h}}_t^g = \overrightarrow{\mathbf{DT}}_g(\mathbf{c_t}, \mathbf{w_t}; \theta_{\overrightarrow{DT}_g})$$

$$\overleftarrow{\mathbf{h}}_t^g = \overleftarrow{\mathbf{DT}}_g(\mathbf{c_t}, \mathbf{w_t}; \theta_{\overleftarrow{DT}_g})$$

The Sequence Labelling Encoder is computed as follows and the word embeddings x_t is fed into this encoder:

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$$
$$\overrightarrow{\mathbf{h}_t} = \overrightarrow{\mathbf{DT}}_{en}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}; \theta_{\overrightarrow{DT}_{en}})$$
$$\overleftarrow{\mathbf{h}_t} = \overleftarrow{\mathbf{DT}}_{en}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t-1}; \theta_{\overleftarrow{DT}_{en}})$$

Then the output of the sequence labelling encoder, h_t is given to the sequence labelling decoder with the following equation and the label of the current word is predicted with a probability function:

$$\mathbf{s}_t = \mathbf{DT}_{de}(\mathbf{h}_t, \mathbf{y}_{t-1}; \theta_{DT_{de}})$$
$$\mathbf{l}_t = \mathbf{s}_t \mathbf{W}_l + \mathbf{b}_l$$
$$P(y_t = j|\mathbf{x}) = softmax(\mathbf{l}_t)[j]$$

GCDT got an F1 Score of 93.47 with BERT Embedding and 91.96 without BERT Embedding(This paper's architecture) in Conll 2003 shared task.

ii. Background in Flair:

Algorithm 1 Compute pooled embedding

Input: *sentence*, *memory*

```
for word in sentence do
    emb_context ←
        embed(word) within sentence
    add emb_context to memory[word]
    emb_pooled ← pool(memory[word])
    word.embedding ←
        concat(emb_pooled, emb_context)
end for
```

The method used in the Flair Algorithm is a combination of different yet useful operations and uses dynamic memory. The operations are:

pool(): As the name suggests the pool operation şs used for pooling embedded vectors (Akbik, Bergmann, Vollgraf; 2019

embed(): which contextualizes an embedding for a given word in a sentence and the memory is used for each unique word contextual embeddings. Similar to Algorithm 1 an embedding is made for a word and that embedding is saved to the memory. Flair got an F1 Score of 93.18 in Conll 2003 shared task.

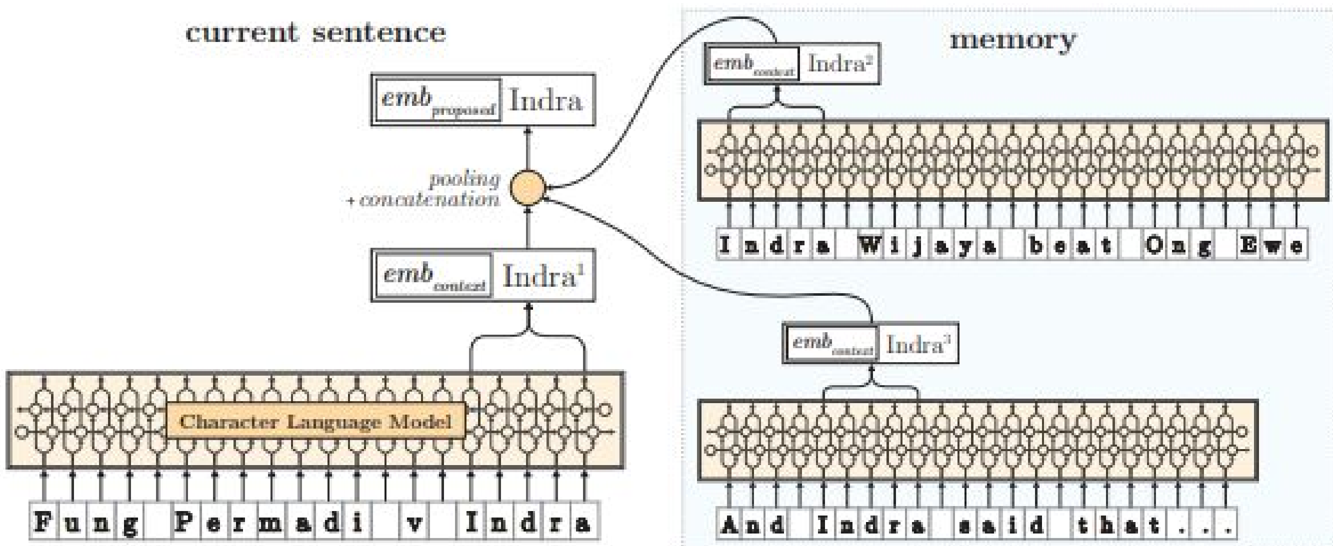

Figure 4: *An Overview of Pooled Contextualized Embeddings with Flair Architecture*

iii. Background in Bidirectional Long Short Term Memory (BiLSTM) with Convolutional Neural Networks(CNN) Architecture:

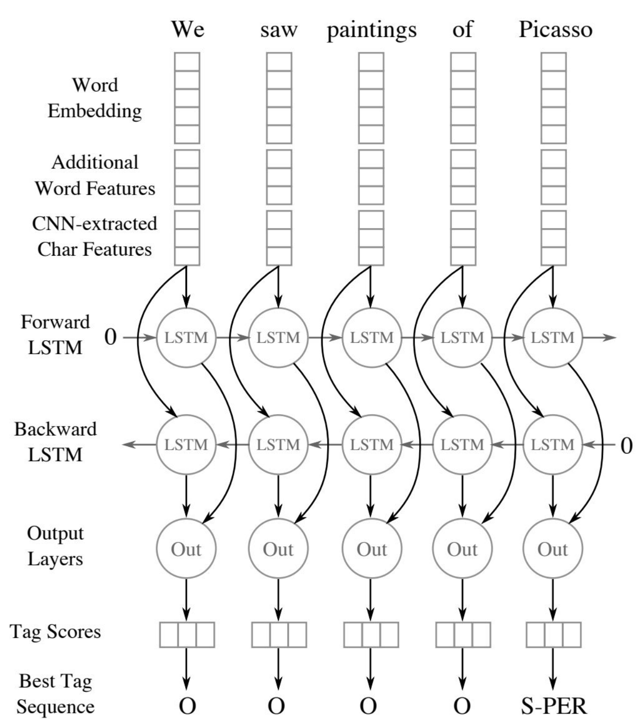

Figure 5: *An Overview of Bidirectional LSTM Architecture*

In this particular algorithm, lookup tables turn features such as words and characters into continuous vectors and then these vectors are given to the neural network. Moreover, instead of using a feed forward architecture, this algorithm uses bi directional Long Short Term Memory(LSTM) and a Convolutional Neural Network is used to induce character level features.

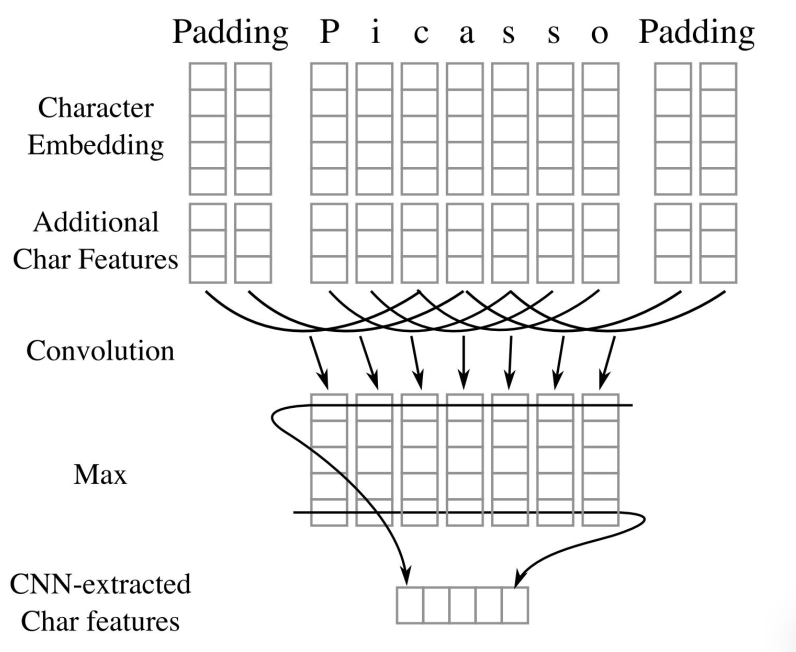

Figure 6: *An Overview of CNN Architecture*

The sequence labelling in this bidirectional LSTM with CNN architecture is as follows:

The extracted features of each discrete word and character is given to the forward and backward LSTM networks as shown in Figure 7. The output of both the layers are decoded by Linear and Log-Softmax layers. This makes a log-probability for each label category. Finally the vectors of these operations are added together. The BiLSTM+CNN model got an F1 score of 91.62 on the Conll 2003 shared task.

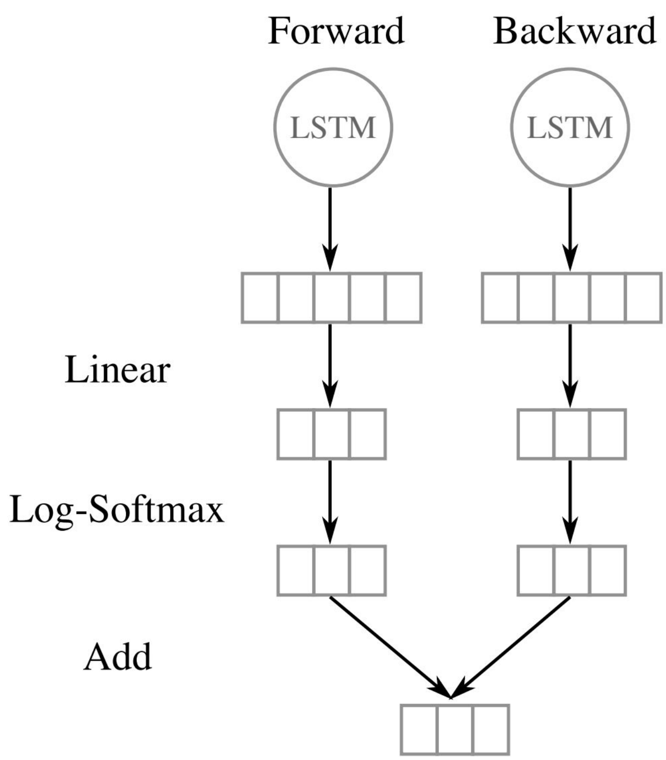

Figure 7: *An Overview of Sequence Labelling in BiLSTM Architecture*

iv. Background in Adapting Transformer Encoder for Named Entity Recognition Architecture (TENER):

To understand Adapting Transformer Encoder for Named Entity Recognition the Transformer should be understanded. Transformer model which was introduced in 2017 depends on self attention. As opposed to using sinusoid position embedding, Transformer uses the distance between to features, tokens should be computed as their attention score. This decreased the time and computational complexity from $O(l^2 d)$ to $O(ld)$ where d is the hidden size and l is the length of the sequence. In this case the Pool Embedding of the t'th token is as follows:

$$PE_{t,2i} = sin(t/10000^{2i/d}),$$

$$PE_{t,2i+1} = cos(t/10000^{2i/d}),$$

i is the between $[0, \frac{d}{2}]$ and d is the input dimension. From these equations it can be deduced that position embedding of the t'th token is as follows:

$$PE_t = \begin{bmatrix} \sin(c_0 t) \\ \cos(c_0 t) \\ \vdots \\ \sin(c_{\frac{d}{2}-1} t) \\ \cos(c_{\frac{d}{2}-1} t) \end{bmatrix}$$

Where d is the input dimension and c is a constant that is $1/10000^{2i/d}$ and depends on i. TENER got an F1 Score Score of 92.62 in Conll 2003 shared task.

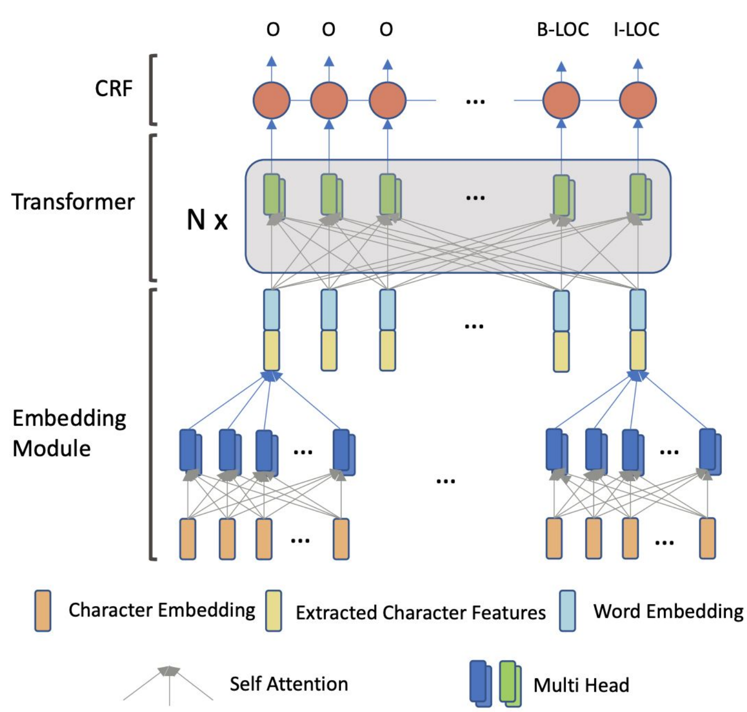

Figure 8: *An Overview Adaptive Transformer Encoder for Named Entity Recognition Architecture*

v. Background in Conll 2003

Conll 2003 is a shared task on Named Entity Recognition which is independent of the language. Conll 2003 dataset is composed of 4 main entity classes; Person, Organization, Location and Miscellaneous which is for entities that do not belong to the other entity classes; and O for the not found entities.

The original data has 2 Languages, English and German, with 4 different data files; training, development, testing and a file with unannotated data; for each of the languages. The data is taken from Reuters news corpus. Conll 2003 dataset has become one of the building blocks of the Named Entity Recognition. It has become one of the most used tasks for different Named Entity Recognition algorithms.

vi. Background in Fictional and Fantasy Named Entity Recognition:

As stated previously, identification and tagging of fictional and fantasy text has been a challenge for the current state of art algorithms and architectures. On the contrary there are some algorithms that are specifically developed and trained for fictional and fantasy text. Entity Typing in Fictional Texts (ENTYFI) is one of the modern Named Entity Recognition tools that was designed for the Fictional and Fantasy dataset such as Wikia Articles. In this case the algorithm needs to identify a relatively new piece of Text and language that needs a fictional background knowledge if the algorithm is only trained on non fiction texts.

"After Melkor's defeat in the First Age, Sauron became the second Dark Lord and strove to conquer Arda by creating the Rings" is an instance that could show the difference between nonfiction and fiction texts. In this case "Arda" is a Location, "First Age" is a Timex, "Sauron" is a Person.

3. Data:

a. Conll 2003:

All of the algorithms used in this article is trained on the same piece of dataset which depends on the Conll 2003 dataset shared task on Named Entity Recognition. This data consists of 2 languages, German and English. The English data was originated form Reuters News corpus. The English training data includes 203,621 Tokes as shown in the Table 1. On the contrary the German training data consists of 206,931 tokens with less articles and sentences than English Training data as shown in Table 2.

English Data	Articles	Sentences	Tokens
Training	946	14,987	203,621
Development	216	3,466	51,362
Testing	231	3,684	46,435

Table 1: *The distribution of the Articles, Sentences and Tokens for files in the Conll 2003 task English data*

German Data	Articles	Sentences	Tokens
Training	553	12,705	206,931
Development	201	3,068	51,444
Testing	155	3,160	51,943

Table 1: *The distribution of the Articles, Sentences and Tokens for files in the Conll 2003 task German data*

As Previously stated, Conll 2003 task depends on 4 main entity classes; Person (Per), Organization (Org), Location (Loc) and Miscellaneous (Misc) which is for entities that do not belong to the other entity classes; and O for the not found entities as shown in the Table 3.

English Data	Location	Miscellaneous	Organization	Person
Training	7140	3438	6321	6600
Development	1837	922	1341	1842
Testing	1668	702	1661	1617

Table 3: *The distribution of the labels for files in the Conll 2003 task English data*

Moreover Conll 2003 also includes German data which has the same labels for English Dataset as shown in Table 4.

German Data	Location	Miscellaneous	Organization	Person
Training	4363	2288	2427	2773
Development	1181	1010	1241	1401
Testing	1035	670	773	1195

Table 4: *The distribution of the labels for files in the Conll 2003 task German data*

In Conll 2003 a sentence is represented as format in the Table 5, where each line contains a tag of one word and/or a character of the sentence. If the sentence consists of N features including characters and words, then the Conll 2003 data of the same sentence contains N lines, each line containing the word and 3 taggs of the word for the various Natural Language Processing Applications and Implementations:

Word	**Part of Speech Tag**	**Chunk Tag**	**Named Entity Recognition Tag**
U.N	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekaus	NNP	I-NP	I-PER

heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

Table 5: *The format of the Conll 2003 Data*

b. Fictional and Fantasy Text:

To evaluate the 4 state of art named entity recognition algorithms and architectures a new testing corpus has been developed. This corpus involves 44 sentences 11 from each folder; Wikia, Shakespeare, Lord of the Rings/Hobbitand Harry Potter respectively; from different resources including books, websites etc.

i. Wikia (Fictional-Fantasy) Corpus:

Wikia is an online website that involves fictional and fantasy fon pages and origin stories of the characters. This part of the dataset contains the piece of text that was coming from Wikia Fandom pages; Anakin Skywalker, Obi Wan Kenobi, Yoda, Luke Skywalker, Harry Potter, Voldemort, Arrow, Green Lantern and Lex Luthor respectively. These corpus was then labelled by hand to compare with the machine output and get an F1 score for analysis

ii. Lord of the Rings and Hobbit(Fantasy)Corpus:

This part of the dataset contains the piece of text that was coming from famous Hobbit and Lord of the Rings quotes; both from the book and from the movies. These corpus was then labelled by hand to compare with the machine output and get an F1 score for analysis

4. Experiments:

a. Training:

All of the algorithms are trained on the same dataset which is Conll 2003. 2 of the 4 algorithms: Global Context Enhanced Deep Transition Architecture

(GCDT), Bidirectional Long Short Term Memory with Convolutional Neural Networks (BiLSTM+CNN) were trained on the same hardware. On the contrary Adapting Transformer Encoder for Named Entity Recognition Architecture (TENER) was trained on a separate hardware due to the CUDA GPU requirements. Flair Embeddings with Pooling algorithm on the other hand was already trained and thus it was only tested

	Number of Epochs	Batch Size	Training Accuracy
GCDT	30000	1024	
BiLSTM+CNN	50		
TENER	100	8	

Table 6: *The Training Settings of the Algorithms trained on Conll 2003 Data*

b. Evaluation:

Apart from the Global Context Enhanced Deep Transition Architecture (GCDT) and Flair Embeddings with Pooling algorithm, other 2 algorithms: TENER and BiLSTM+CNN were evaluated with the training. GCDT on the other hand was evaluated separately after the training due to the training efficiency. The model was evaluated and tested through the checkpoints saved.
-------Here comes the evaluation graphs and charts I have the accuracy and loss per epoch for 3 of the models to be added today-------

c. Testing:

To test the 4 different state of art models, 5 different corpus were collected from different sources including novels and internet fan websites and labelled by hand. The first corpus is the same corpus that the algorithms were trained with. This corpus is the control case. The second corpus is the texts collected from Wikia. This corpus does not involve as many fantastic elements as the Lord of the Rings/Hobbit and Harry Potter corpus involves. But it involves more fantasy elements than the Shakespeare corpus. The third corpus is the Lord of the Rings and Hobbit Corpus which is collected from the books. This corpus is one of the 2 fantasy corpuses tested in this paper. The forth corpus is the Harry Potter Corpus which is collected again from the Harry Potter book series and is the other fantasy corpus tested in this paper. The last data that is tested is the Shakespeare Corpus which is collected from the plays and books of Shakespeare. This corpus is more realistic than other fictional corpuses. For the comparison and scoring between the models; F1 score, precision and recall will be used.

5. Results:

➢ Conll 2003 Corpus

	F1 Macro	F1 Micro	F1 Average	F1 None	Precision	Recall
GCDT						
Flair						
BiLSTM+ CNN						
TENER						

Table 7: *The Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Conll 2003 Corpus*

➢ Wikia Corpus

	F1 Macro	F1 Micro	F1 Average	F1 None	Precision	Recall
GCDT						
Flair						
BiLSTM+ CNN						
TENER						

Table 8: *The Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Wikia Corpus*

➢ Lord of the Rings and Hobbit Corpus

	F1 Macro	F1 Micro	F1 Average	F1 None	Precision	Recall
GCDT						

Flair						
BiLSTM+ CNN						
TENER						

Table 9: *The Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Lord of the Rings and Hobbit Corpus*

- Shakespeare Corpus

	F1 Macro	F1 Micro	F1 Average	F1 None	Precision	Recall
GCDT						
Flair						
BiLSTM+ CNN						
TENER						

Table 10: *The Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Lord of the Rings and Hobbit Corpus*

- Harry Potter Corpus

	F1 Macro	F1 Micro	F1 Average	F1 None	Precision	Recall
GCDT						
Flair						
BiLSTM+ CNN						
TENER						

Table 11: *The Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Harry Potter Corpus*

6. **Analysis:----Will be written today after the full results---**
7. **Conclusion:----Will be written today after the full results---**
8. **Acknowledgements:----Will be written today after the full results---**
9. **References:----Will be written today after the full results---**