Named Entity Recognition for Fictional, Nonfiction and Fantasy 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 significant branches of Natural Language Processing(NLP). Named Entity Tagging aims to analyse texts and classify the words entities as some categories including Person(Per), Location(Loc), Organization(Org), Miscellaneous(Misc) etc. and encode those entities in Begin(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..

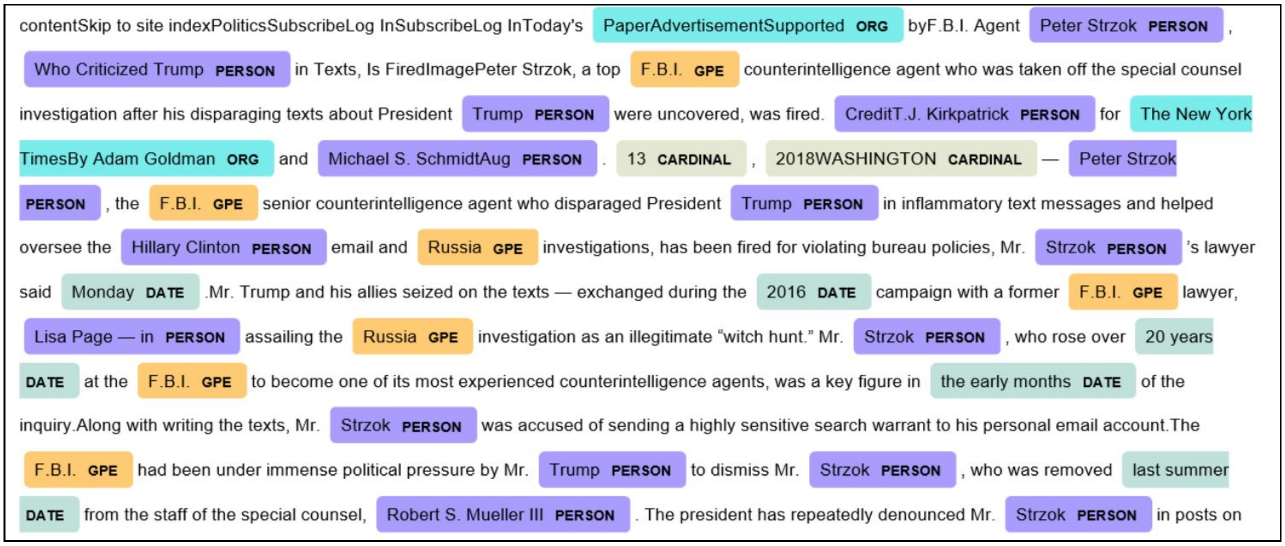
This paper focuses on the performance of three learning methods, Global Context Enhanced Deep Transition Architecture (GCDT) without BERT embedding, Flair, Bidirectional Long Short Term Memory (LSTM) + Convolutional Neural Networks (CNN) and Adapting Transformer Encoder for Named Entity Recognition Architecture (TENER) in Fictional, Non-Fiction 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 state of the art Named Entity Recognition Algorithms and suggests a new and better-synthesized 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, Bidirectional LSTM+CNN, TENER

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 definition 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 vast dataset, annotated (Supervised), or unannotated (Unsupervised) depending on the learning method.



contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported ORG byF.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top F.B.I. GPE counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump PERSON were uncovered, was fired. CreditT.J. Kirkpatrick PERSON for The New York TimesBy Adam Goldman ORG and Michael S. SchmidtAug PERSON . 13 CARDINAL , 2018WASHINGTON CARDINAL — Peter Strzok PERSON , the F.B.I. GPE senior counterintelligence agent who disparaged President Trump PERSON in inflammatory text messages and helped oversee the Hillary Clinton PERSON email and Russia GPE investigations, has been fired for violating bureau policies, Mr. Strzok PERSON 's lawyer said Monday DATE .Mr. Trump and his allies seized on the texts — exchanged during the 2016 DATE campaign with a former F.B.I. GPE lawyer, Lisa Page — in PERSON assailing the Russia GPE investigation as an illegitimate “witch hunt.” Mr. Strzok PERSON , who rose over 20 years DATE at the F.B.I. GPE to become one of its most experienced counterintelligence agents, was a key figure in the early months DATE of the inquiry.Along with writing the texts, Mr. Strzok PERSON was accused of sending a highly sensitive search warrant to his personal email account.The F.B.I. GPE had been under immense political pressure by Mr. Trump PERSON to dismiss Mr. Strzok PERSON , who was removed last summer DATE from the staff of the special counsel, Robert S. Mueller III PERSON . The president has repeatedly denounced Mr. Strzok PERSON in posts on

Figure 1: *An Example of Named Entity Recognition (Printed from SpaCy’s Named Entity Recognition Parser)*

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 organisations, 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 Begin(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 synthesised model should be trained and tested using the current state of art models and corpus.

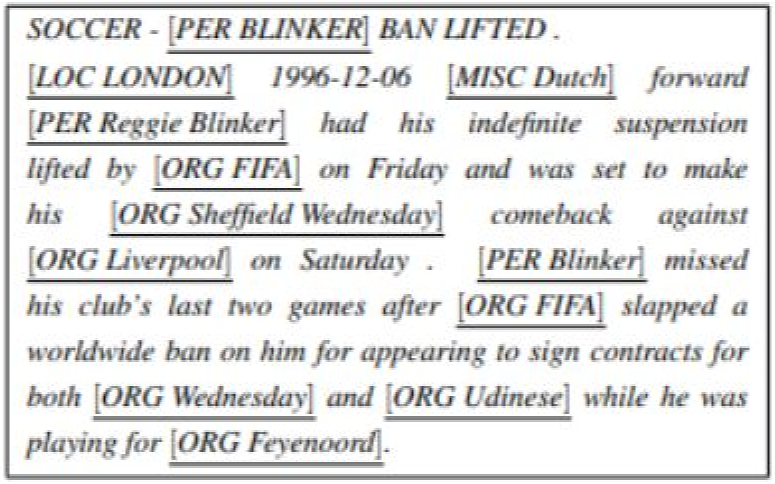
2. Related Work:

2.1 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.

2.2 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 state of the 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 modelling 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*

(Printed from Ratinov, Lev, and Dan Roth. “Design Challenges and Misconceptions in Named Entity Recognition.” Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL '09, 2009, doi:10.3115/1596374.1596399.)

In this particular instance, it is not apparent 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 multiple networks.

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

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

Similarly, y can be defined as follows:

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

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

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

In this form; k, l, and m are tiny 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:

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

2.2.1 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.

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

The Global Context Enhanced Deep Translation Architecture GCDT is a type of architecture for sequence labelling presented by the researchers Beijing Jiaotong University. Unlike the Recurrent Neural 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 $x = (x_1,\ x_2,\ x_3,\ x_4 \ldots\ x_N)$. 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.

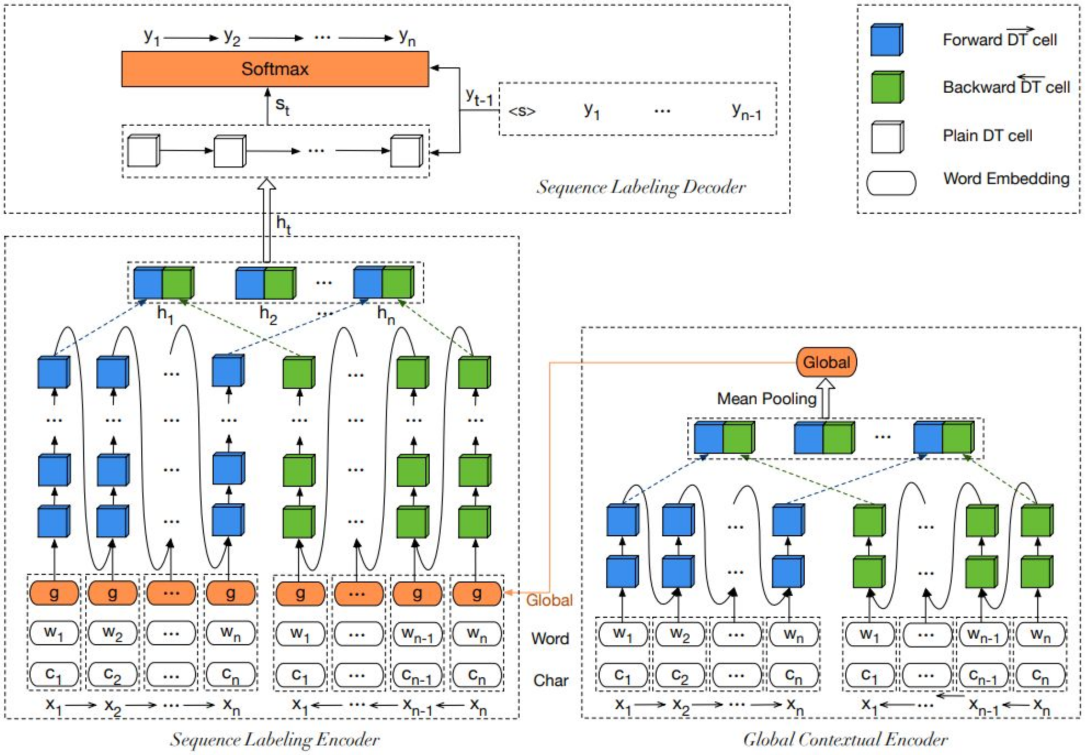


Figure 3: *An Overview of Global Context Enhanced Deep Transition Architecture*

(Printed from Liu, Yijin, et al. "GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling." Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, doi:10.18653/v1/p19-1233.)

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 can be calculated as:

$$\begin{aligned}
&\mathbf{g} = \frac{1}{N}\sum_{t=1}^{n}\mathbf{h}_t^g \\
&\mathbf{h}_t^g = [\overrightarrow{\mathbf{h}}_t^g; \overleftarrow{\mathbf{h}}_t^g] \\
&\overrightarrow{\mathbf{h}}_t^g = \overrightarrow{\mathbf{DT}}_g(\mathbf{c_t}, \mathbf{w_t}; \theta_{\overrightarrow{DT}_g}) \\
&\overleftarrow{\mathbf{h}}_t^g = \overleftarrow{\mathbf{DT}}_g(\mathbf{c_t}, \mathbf{w_t}; \theta_{\overleftarrow{DT}_g})
\end{aligned}$$

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

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

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:

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

GCDT got an F1 Score of 93.47 with BERT Embedding and 91.96 without BERT Embedding(Architecture used in this Paper) in Conll 2003 shared task.

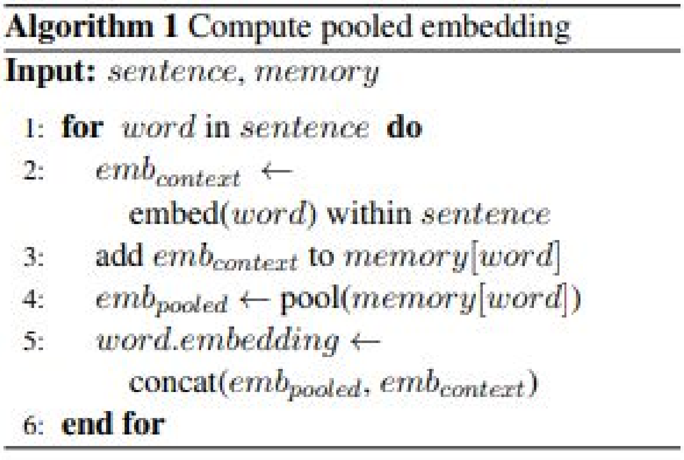
2.2.1.2 Background in Flair:

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 contextualises 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.



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
```

__Algorithm 1:__ An Overview of Pooled Contextualized Embeddings with Flair Architecture Algorithm

(Printed from Akbik, Alan, et al. "Pooled Contextualized Embeddings for Named Entity Recognition." Proceedings of the 2019 Conference of the North, 2019, doi:10.18653/v1/n19-1078.)

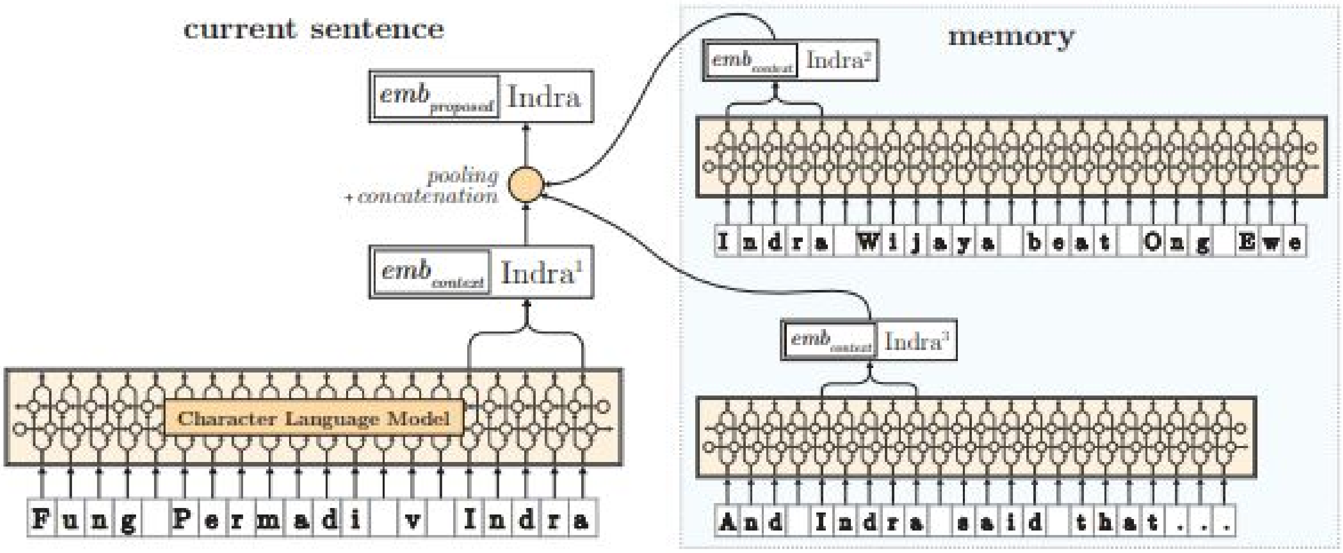


__Figure 4:__ An Overview of Pooled Contextualized Embeddings with Flair Architecture

(Printed from Akbik, A., Bergmann, T., & Vollgraf, R. (2019). Pooled Contextualized Embeddings for Named Entity Recognition. Proceedings of the 2019 Conference of the North. doi:10.18653/v1/n19-1078)

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

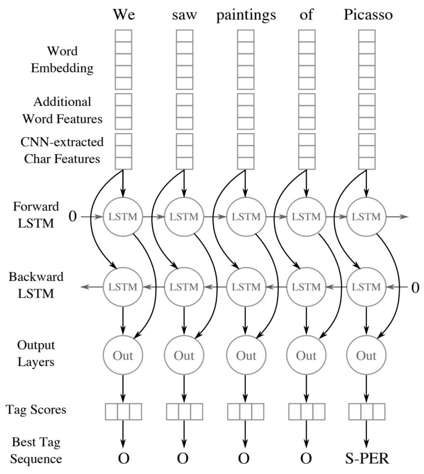


Figure 5: *An Overview of Bidirectional LSTM Architecture*

(Printed from Chiu, Jason P.c., and Eric Nichols. "Named Entity Recognition with Bidirectional LSTM-CNNs." Transactions of the Association for Computational Linguistics, vol. 4, 2016, pp. 357–370., doi:10.1162/tacl_a_00104.)

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

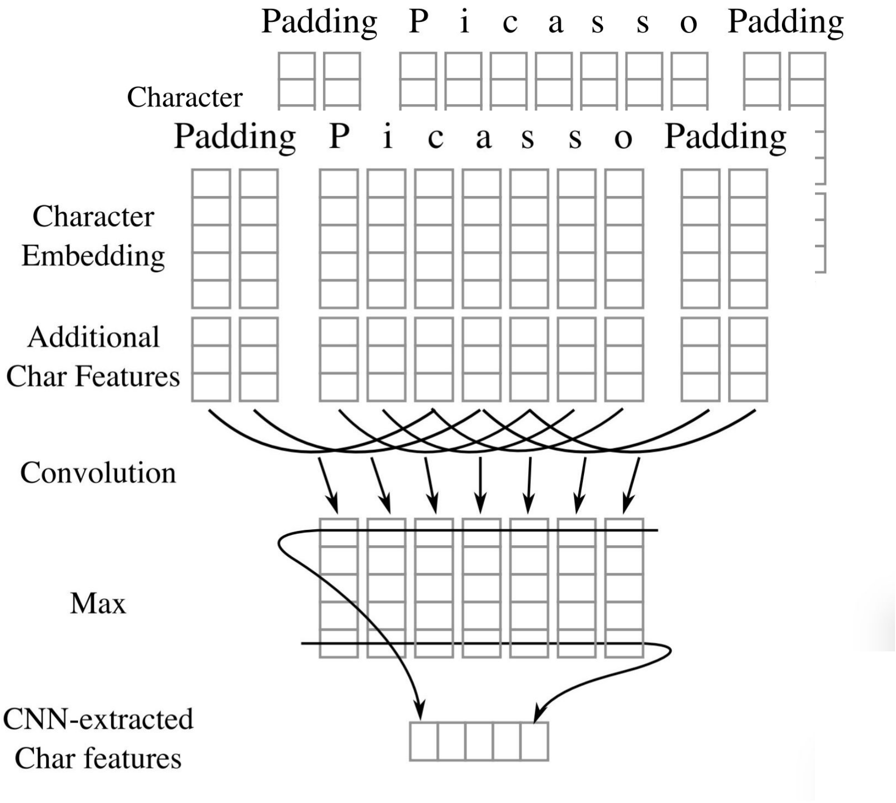

Padding P i c a s s o Padding
Character
Padding P i c a s s o Padding
Character Embedding
Additional Char Features
Convolution
Max
CNN-extracted Char features

Figure 6: *An Overview of CNN Architecture*

(Printed from Chiu, Jason P.c., and Eric Nichols. "Named Entity Recognition with Bidirectional LSTM-CNNs." Transactions of the Association for Computational Linguistics, vol. 4, 2016, pp. 357–370., doi:10.1162/tacl_a_00104.)

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.

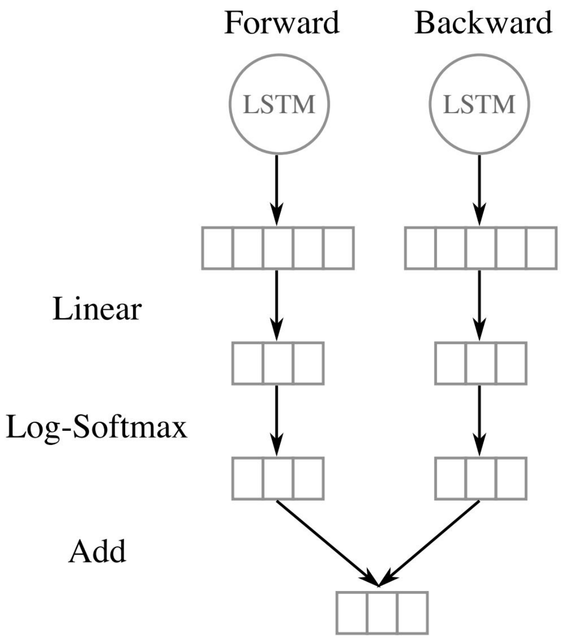


***Figure 7:** An Overview of Sequence Labelling in BiLSTM Architecture*

(Printed from Chiu, Jason P.c., and Eric Nichols. "Named Entity Recognition with Bidirectional LSTM-CNNs." Transactions of the Association for Computational Linguistics, vol. 4, 2016, pp. 357–370., doi:10.1162/tacl_a_00104.)

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

To understand Adapting Transformer Encoder for Named Entity Recognition, the Transformer should be understood. 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 $O(l^2d)$ 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.

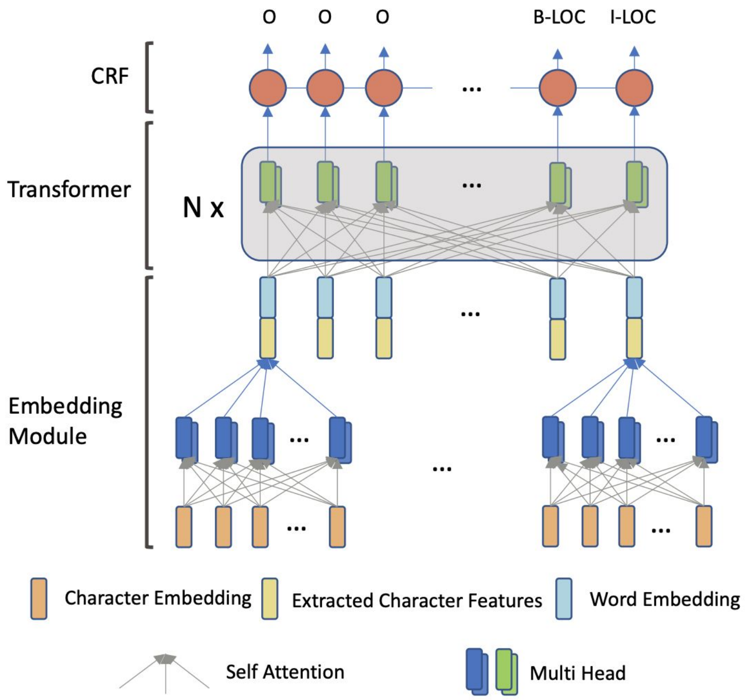


Figure 8: *An Overview Adaptive Transformer Encoder for Named Entity Recognition Architecture*

(Printed from Yan, H., Deng, B., Li, X., & Qiu, X. (n.d.). TENER: Adapting Transformer Encoder for Named Entity Recognition. Retrieved November 10, 2019, from https://arxiv.org/abs/1911.04474.)

2.2.2 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 four, 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.

2.2.3 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:

3.1 Conll 2003:

All of the algorithms used in this article is trained on the same 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 from Reuters News corpus. The English training data includes 203,621 Tokes, as shown in 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 2: *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 four 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 Table 3.

English Data	Location	Miscellaneous	Organisation	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	Organisation	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 three tags 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*

3.2 Fictional and Fantasy Text:

To evaluate the four state of the art named entity recognition algorithms and architectures, a new testing corpus has been developed. This corpus involves over 1800 instances from 4 classes;

Conll03, Fictional Text, Wikia Text and Fantasy Text respectively; from different resources including books, websites etc.

3.2.1 Fictional Corpus:

This part of the dataset contains the piece of text that was coming from famous plays of Shakespere (not in old English) and From 2 Sherlock Holmes Books, The Valey of Fear and The Hound of Baskervilles, written by Sir Arthur Conan Doyle. This corpus was then labelled by hand to compare with the named entity recognition architecture output and get an F1 score, precision, accuracy and recall for analysis. This part of the dataset contains over 440 tokens.

Tokens	Named Entity Recognition Tag
The	O
Hound	B-MISC
of	I-MISC
the	I-MISC
Baskervilles	E-MISC
even	O
in	O
daylight	O
were	O
not	O
pleasant	O
to	O
hear	O
.	O

Table 6: *The format of the Fictional Corpus*

(Tokens from Doyle, S. A. (2011). Hound of the baskervilles: Another adventure of sherlock holmes. Place of publication not identified: Martino Fine Books.)

3.2.2 Wikia 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, Lex Luthor, Tatooine and Naboo respectively. This corpus was then labelled by hand to compare with the machine learning output and get an F1 score, Recall, Accuracy and Precision for analysis. This part of the data includes over 450 tokens.

Tokens	Named Entity Recognition Tag

Tatooine	S-LOC
was	O
a	O
sparsely	O
inhabited	O
circumbinary	O
desert	O
planet	O
located	O
in	O
the	O
galaxy	O
s	O
Outer	B-LOC
Rim	I-LOC
Territories	E-LOC
.	O

Table 7: *The format of the Wikia Corpus*

(Tokens from https://starwars.fandom.com/wiki/Tatooine)

3.2.3 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, as well as Harry Potter, quotes from both the movies and books. This corpus was then labelled by hand to compare with the named entity recognition architecture output and get an F1 score, precision, accuracy and recall for analysis. This part of the dataset contains over 420 tokens.

Tokens	**Named Entity Recognition Tag**
There	O
is	O

only	O
one	O
Lord	B-PER
of	I-PER
the	I-PER
Ring	E-PER
,	O
only	O
one	O
who	O
can	O
bend	O
it	O
to	O
his	O
will	O
.	O

Table 8: *The format of the Fantasy Corpus*

(Tokens from Tolkien, J. R. R. The Lord of the Rings. the Fellowship of the Ring. 2007.)

4. Experiments:

4.1 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	Number of Workers
GCDT	30000	1024	N/A
BiLSTM+CNN	50	64	0
TENER	100	8	N/A

Table 9: *The Training Features of the Algorithms trained on Conll 2003 Data*

4.2 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.

	F1 Score	Precision	Recall
GCDT	94.64	94.39%	94.88%
TENER	91.0832	91.752%	90.4241%
BiLSTM+CNN	92.2000	91.900%	92.600%

Table 10: *The Evaluation Results of the Algorithms trained on Conll 2003 Data*

4.3 Testing:

To test the 3 different state of the art models, four different corpus were collected from different sources, including novels and internet fan websites and labelled by hand. The first corpus, Conll 2003 Corpus, is the same corpus that the algorithms were trained with. This corpus is the control case in the experiment. The second Corpus is the texts collected from Shakespeare plays and two of the Sherlock Holmes books by Sir Arthur Conan Doyle. The second corpus is the texts collected from Wikia. This corpus does not involve as many fantastic elements as the Fantastic Corpus involves. But it involves more fantastic and fictional 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 two fantasy corpora 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 corpora. For the comparison and scoring between the models; F1 score, precision, accuracy and recall will be used. Because TENER by FastNLP does not provide the testing algorithm with the open-source code in Github, the Demo for testing is requested from them. On the contrary, the instructions were not compatible with the infrastructure that the model was trained on. The provided codes to test had some bugs with the libraries used: Torch and FastNLP. For these reasons only 3 state of art models were tested and analyzed and this was the only unsolvable limitation through the research. If the bugs are solved in the Libraries FastNLP and Torch, then for the further development of the fourth and the last model, TENER will be tested successfully for deeper comparison between the state of art architectures.

The outputs from the three state of art architectures then were evaluated using a python script version of conlleval.

5. Learning Approach:

Learning Approach used in this article is Domain Adaptation. In this paper 4 different data stated earlier are tested on 3 state of art Architectures that are trained on the same dataset, Conll 2003. A limitation for such a task was the fact that Bidirectional LSTM+CNN Architecture was using a different tagging method, IOB. Instead of using the BIOES like the other algorithms, it was just using B(Beginning), I (Inside) and O (No Chunks) tags. This was overcome using a python script that was converting IOB output into BIOES to test using conlleval.

6. Results:

6.1 GCDT Results

6.1.1 GCDT Conll03 Corpus Test

processed 432 tokens with 48 phrases; found: 49 phrases; correct: 47.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
99.31%	98.46%	95.92%	97.92%	96.91

Table 11: *The Test Results of the GCDT Architecture trained on Conll 2003 Data & Tested on Conll 2003 Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	95.45%	100.00%	97.67	22
MISC	92.31%	100.00%	96.00	13
ORG	100.00%	100.00%	100.00	2
PER	100.00%	92.31%	96.00	12

Table 12: *The Detailed Test Results of the GCDT Architecture trained on Conll 2003 Data & Tested on Conll 2003 Corpus*

6.1.2 GCDT Fictional Corpus Test

processed 445 tokens with 28 phrases; found: 31 phrases; correct: 15.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score

94.16%	48.89%	48.39%	53.57%	50.85

Table 13: *The Test Results of the GCDT Architecture trained on Conll 2003 Data & Tested on Fictional Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	85.71%	75.00%	80.00	7
MISC	0.00%	0.00%	0.00	0
ORG	50.00%	100.00%	66.67	4
PER	35.00%	43.75%	38.89	20

Table 14: *The Detailed Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Conll 2003 Corpus*

6.1.3 GCDT Wikia Corpus Test

processed 457 tokens with 60 phrases; found: 58 phrases; correct: 35.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
53.47%	84.90%	60.34%	58.33%	59.32

Table 15: *The Test Results of the GCDT Architecture trained on Conll 2003 Data & Tested on Wikia Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	33.33%	60.00%	42.86	9

MISC	29.41%	62.50%	40.00	17
ORG	75.00%	33.33%	46.15	4
PER	85.71%	63.16%	72.73	28

Table 16: *The Detailed Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Conll 2003 Corpus*

6.1.4 GCDT Fantasy Corpus Test

processed 427 tokens with 26 phrases; found: 30 phrases; correct: 10.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
92.51%	28.21%	33.33%	38.46%	35.71

Table 17: *The Test Results of the GCDT Architecture trained on Conll 2003 Data & Tested on Fantasy Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	33.33%	20.00%	25.00	3
MISC	0.00%	0.00%	0.00	1
ORG	25.00%	50.00%	33.33	4
PER	36.36%	53.33%	43.24	22

Table 18: *The Detailed Test Results of the Four Algorithms trained on Conll 2003 Data Tested on Conll 2003 Corpus*

6.2 Flair Results

6.2.1 Flair Conll 2003 Corpus Test

processed 432 tokens with 48 phrases; found: 50 phrases; correct: 47.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
98.84%	98.46%	94.00%	97.92%	95.92

Table 19: *The Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Conll 2003 Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	95.45%	100.00%	97.67	22
MISC	85.71%	100.00%	92.31	14
ORG	100.00%	100.00%	100.00	2
PER	100.00%	92.31%	96.00	12

Table 20: *The Detailed Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Conll 2003 Corpus*

6.2.2 Flair Fictional Corpus Test

processed 445 tokens with 28 phrases; found: 26 phrases; correct: 18.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
96.18%	64.44%	69.23%	64.29%	66.67

Table 21: *The Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Fictional Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	87.50%	87.50%	87.50	8
MISC	100.00%	100.00%	100.00	2
ORG	100.00%	100.00%	100.00	2
PER	50.00%	43.75%	46.67	14

Table 22: *The Detailed Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Fictional Corpus*

6.2.3 Flair Wikia Corpus Test

processed 457 tokens with 60 phrases; found: 55 phrases; correct: 35.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
84.03%	50.69%	63.64%	58.33%	60.87

Table 23: *The Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Wikia Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	71.43%	100.00%	83.33	7
MISC	35.71%	62.50%	45.45	14
ORG	33.33%	44.44%	38.10	12
PER	95.45%	55.26%	70.00	22

Table 24: *The Detailed Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Wikia Corpus*

6.2.4 Flair Fantasy Corpus Test

processed 427 tokens with 26 phrases; found: 25 phrases; correct: 15.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
94.61%	41.03%	60.00%	57.69%	58.82

Table 25: *The Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Fantasy Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	80.00%	20.00%	25.00	5
MISC	50.00%	50.00%	50.00	4
ORG	0.00%	0.00%	0.00	1
PER	60.00%	53.33%	43.24	15

Table 26: *The Detailed Test Results of the Flair Architecture trained on Conll 2003 Data & Tested on Fantasy Corpus*

6.3 BiLSTM+CNN Results

6.3.1 BiLSTM+CNN Conll 2003 Corpus Test

processed 432 tokens with 48 phrases; found: 51 phrases; correct: 45.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score

97.69%	93.85%	88.24%	93.75%	90.91

Table 27: *The Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Conll 2003 Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	90.91%	95.24%	93.02	22
MISC	85.71%	100.00%	92.31	14
ORG	66.67%	100.00%	80.00	3
PER	91.67%	84.62%	88.00	12

Table 28: *The Detailed Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Conll 2003 Corpus*

6.3.2 BiLSTM+CNN Fictional Corpus Test

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
95.51%	64.44%	54.55%	64.29%	59.02

Table 29: *The Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Fictional Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	63.64%	87.50%	73.68	11
MISC	0.00%	0.00%	0.00	0

ORG	50.00%	100.00%	66.67	4
PER	50.00%	56.25%	52.94	18

Table 30: *The Detailed Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Fictional Corpus*

6.3.3 BiLSTM+CNN Wikia Corpus Test

processed 457 tokens with 60 phrases; found: 64 phrases; correct: 35.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
50.69%	84.46%	54.69%	58.33%	56.45

Table 31: *The Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Wikia Corpus*

	Precision	Recall	F1 Score	Instance Count
LOC	23.08%	60.00%	33.33	13
MISC	11.11%	12.50%	11.76	9
ORG	46.15%	66.67%	54.55	13
PER	86.21%	65.79%	74.63	29

Table 32: *The Detailed Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Wikia Corpus*

6.3.4 BiLSTM+CNN Fantasy Corpus Test

processed 427 tokens with 26 phrases; found: 32 phrases; correct: 12.

Accuracy	Accuracy non-O	Precision	Recall	F1 Score
92.51%	38.46%	37.50%	46.15%	41.38

Table 33: *The Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Fantasy Corpus*

	Precision	Recall	F1 Score	Instances Found
LOC	40.00%	40.00%	40.00	5
MISC	0.00%	0.00%	0.00	1
ORG	0.00%	0.00%	0.00	2
PER	41.67%	66.67%	51.28	24

Table 34: *The Detailed Test Results of the Bidirectional LSTM+CNN Architecture trained on Conll 2003 Data & Tested on Fantasy Corpus*

7. Analysis:

Before starting to anaanalyselyze the results, it is required to understand the following terms:

- Accuracy: Accuracy is the ratio of true predictions over sum of all the predictions. It can be written as follows:

$$\frac{TruePositive+TrueNegative}{TruePositive+TrueNegative+FalseNegative+FalsePositive}$$

- Precision: Precision is the ratio of the true positive predictions over all the positive predictions. It can be written as follows:

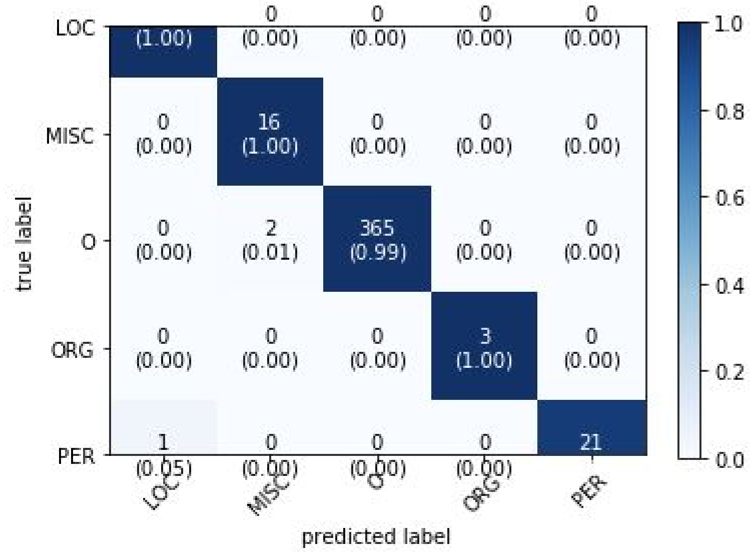
$$\frac{TruePositive}{TruePositive+FalsePositive}$$

- Recall: Recall is the ratio of the true positive predictions over the sum of the false-negative and true positives, sometimes called as the class. Recall can also be mentioned as Sensitivity
- F1 Score: F1 Score can be defined as the weighted average of Precision and Recall. It can be written as:

$$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

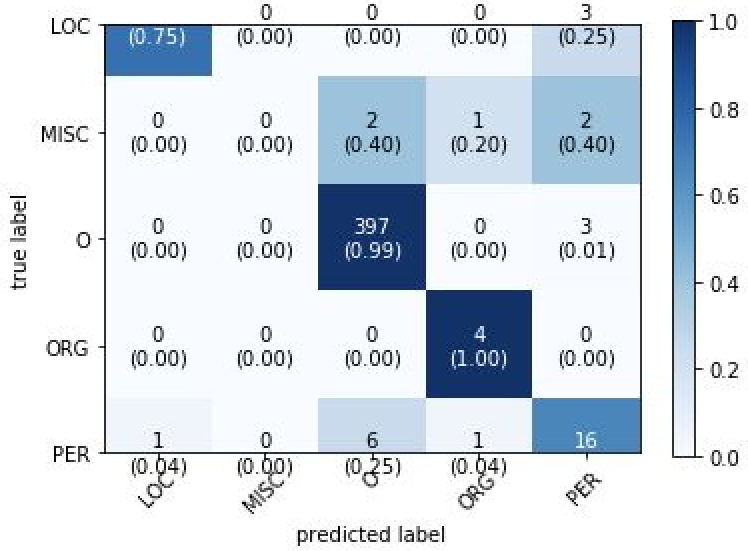
To analyse the results, visualising them using confusion matrices in order to compare the architectures and purpose of a better and more powerful synthesised architecture for named entity recognition for all types of text would be a more effective approach. Moreover, because inequality in the numbers of the instances containing the different labels, especially the "O" label, the confusion matrices are normalised to purpose a more understandable representation.

6.2 GCDT Architecture Analysis

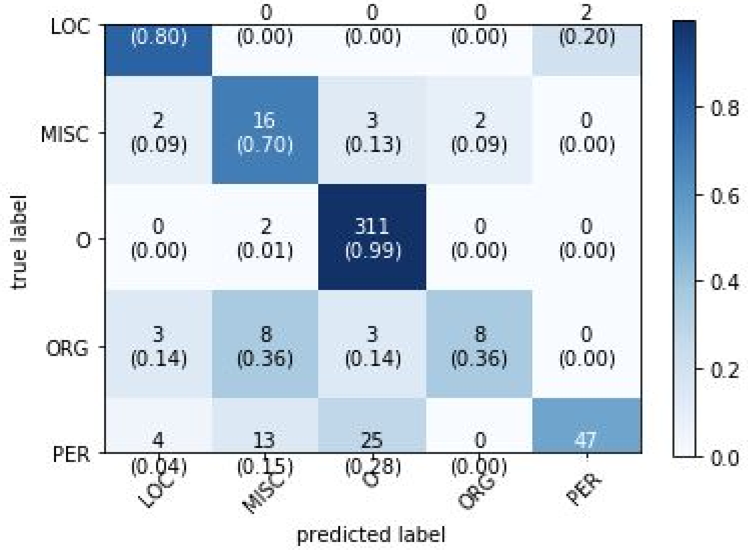


	LOC	MISC	O	ORG	PER
LOC	(1.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)
MISC	0 (0.00)	16 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)
O	0 (0.00)	2 (0.01)	365 (0.99)	0 (0.00)	0 (0.00)
ORG	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)	0 (0.00)
PER	1 (0.05)	0 (0.00)	0 (0.00)	0 (0.00)	21

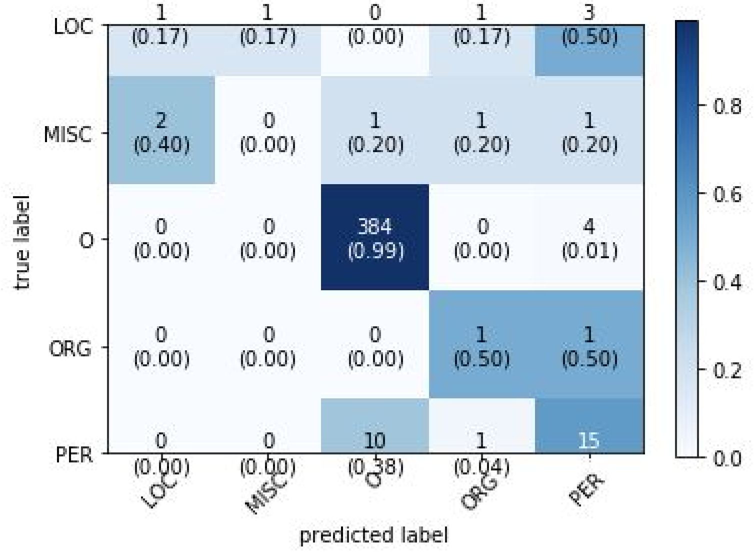
Confusion Matrix 1: *Normalized Confusion Matrix of GCDT Architecture on Conll03 Corpus Test*



Confusion Matrix 2: *Normalized Confusion Matrix of GCDT Architecture on Fictional Corpus Test*



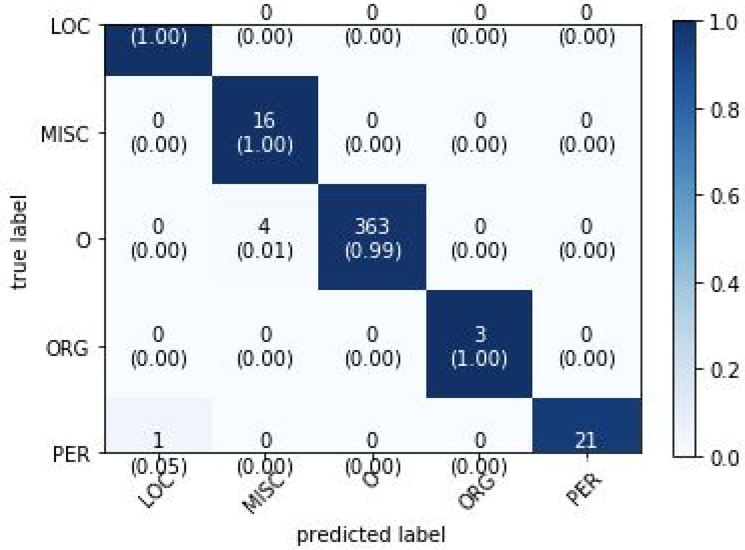
Confusion Matrix 3: *Normalized Confusion Matrix of GCDT Architecture on Wikia Corpus Test*



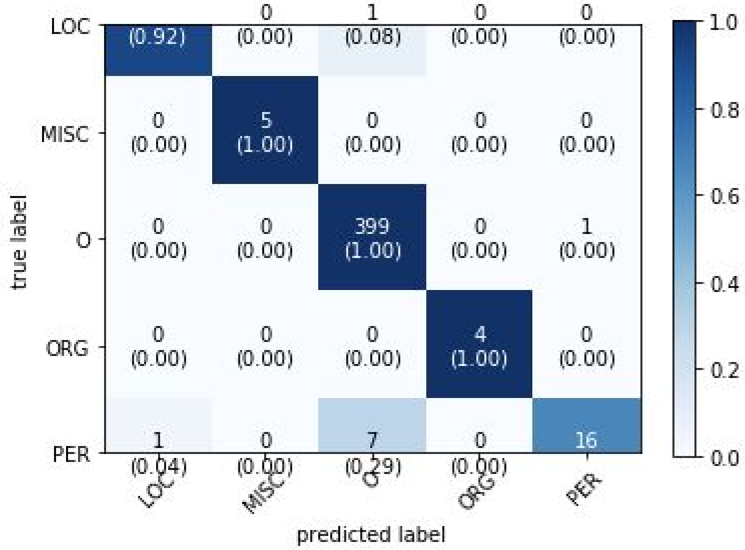
Confusion Matrix 4: *Normalized Confusion Matrix of GCDT Architecture on Fantasy Corpus Test*

As seen in the Confusion Matrix 1, there is a clear diagonal line crossing from the top left corner to the bottom right corner. The Confusion Matrix 1 with the Table 11 suggests that the GCDT Architecture predicts and tags the tokens with an average accuracy of 99.31% including the label “O” and 98.46% with excluding the label “O”. On the contrary as the phrases and words used approaches a more fantasy and fiction level, the accuracy decreases significantly. As seen in Table 13 and Table 11 even though the non-O accuracy for the Wikia Corpus Result is much higher than the Fictional Corpus Result, the normal accuracy is significantly lower than the Fictional Corpus Result. This fact can be seen in the Confusion Matrix 2 and Confusion Matrix 3. In Confusion Matrix 2 even though there are some misclassifications such as the “MISC” tag classification, there is a diagonal passing from the corners. On the other hand, in the Confusion Matrix 3, there is some more noise, but there is a complete diagonal passing from the corners. This fact also can be analyzed using the F1 score of Table 13 and Table 11. Even though the fictional and abstraction level of Wikia corpus is much higher than the Fictional Corpus, the F1 score is higher by nearly 9 points. Finally it can be seen from the Confusion Matrix 4, there is not a clear diagonal line passing from the corners, and there are more misclassifications than the other three corpora (Conll 2003, Fictional, Wikia) analysed. This fact can be deduced from Table 13.

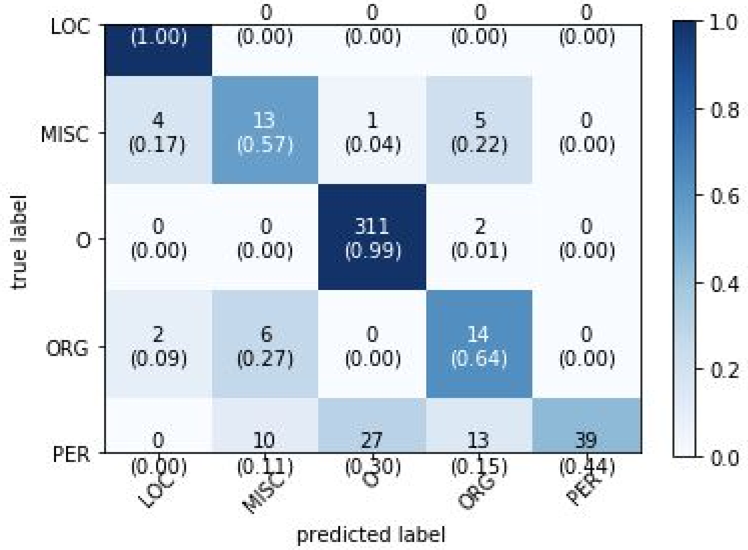
6.2 Flair Architecture Analysis



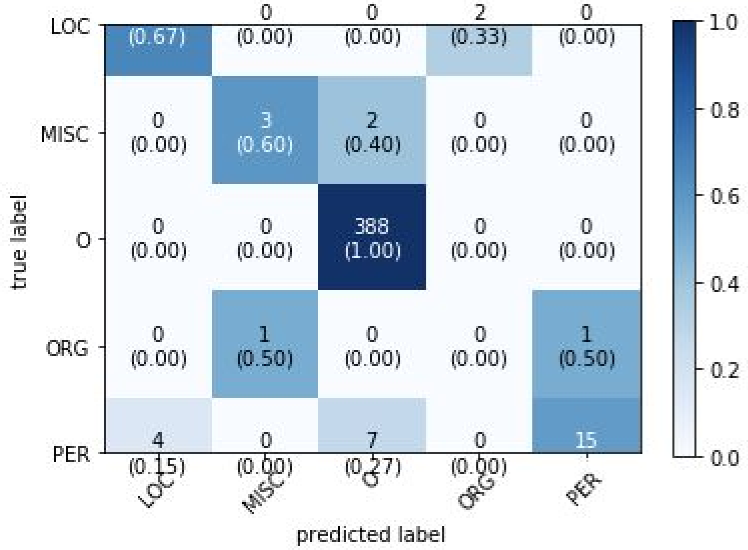
Confusion Matrix 5: *Normalized Confusion Matrix of Flair Architecture on Conll 2003 Corpus Test*



Confusion Matrix 6: *Normalized Confusion Matrix of Flair Architecture on Fictional Corpus Test*



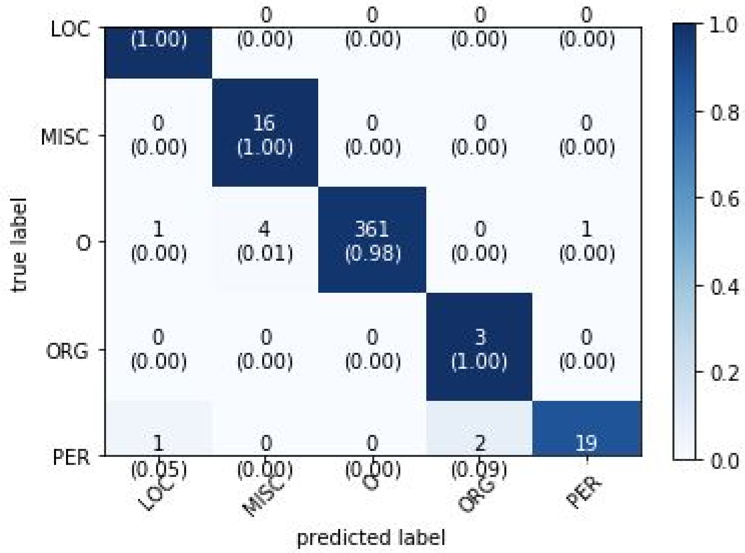
Confusion Matrix 7: *Normalized Confusion Matrix of Flair Architecture on Wikia Corpus Test*



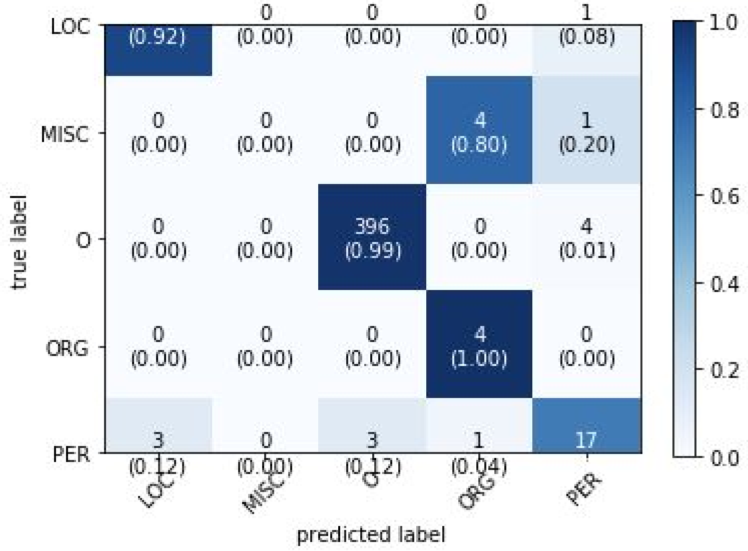
Confusion Matrix 8: *Normalized Confusion Matrix of Flair Architecture on Fantasy Corpus Test*

There is a clear diagonal line crossing from the top left corner to the bottom right corner in the Confusion Matrix 5. Unlike the GCDT Architecture, Flair Architecture involves a much stable and visible diagonal in the Fictional Corpus as presented in Confusion Matrix 6. Both non-O and standard accuracy and Recall, Precision and F1 Score in Table 21 is higher than Table 13 which supports this argument, which suggests that Flair Architecture is a better algorithm to tag and classify fictional text than the GCDT Architecture. On the contrary, as seen from the difference between Table 19 and Table 11. Flair Architecture is slightly less accurate in Conll 2003 control task. This situation can be understood from the difference between the Confusion Matrix 1 and Confusion Matrix 5. Similar to the GCCDT Architecture, the accuracy and the clarity of the diagonal in the confusion matrices decreases as the type of text approaches fantasy and fiction. This fact can be supported by Table 21, Table 23 and Table 25. Another support can be the noise and unclarity in the Confusion Matrix 7 and Confusion Matrix 8.

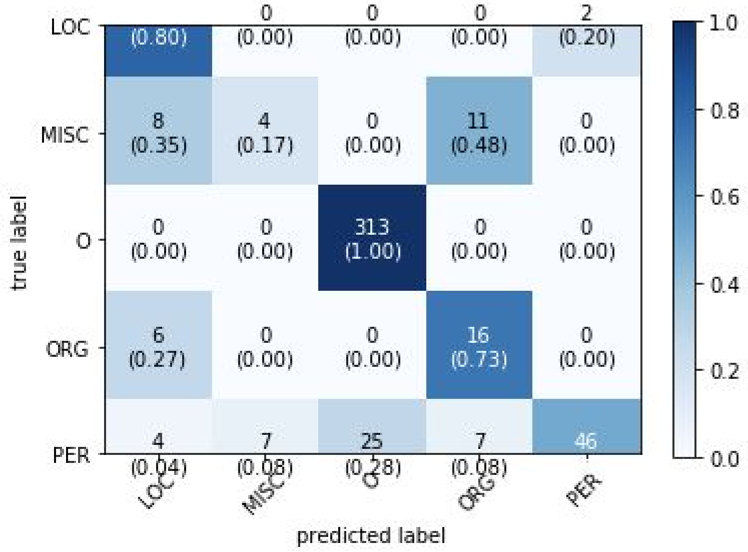
6.3 Bidirectional LSTM+CNN Architecture Analysis



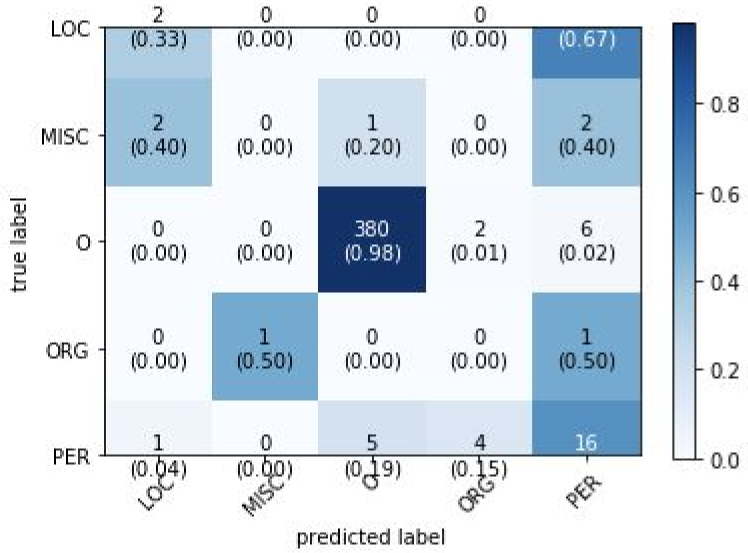
Confusion Matrix 9: *Normalized Confusion Matrix of BiDirectional LSTM+CNN Architecture on Conll 2003 Corpus Test*



Confusion Matrix 10: *Normalized Confusion Matrix of BiDirectional LSTM+CNN Architecture on Fictional Corpus Test*



Confusion Matrix 11: *Normalized Confusion Matrix of BiDirectional LSTM+CNN Architecture on Wikia Corpus Test*



Confusion Matrix 12: *Normalized Confusion Matrix of BiDirectional LSTM+CNN Architecture on Fantasy Corpus Test*

It can be seen that there is a clear diagonal line crossing from the top left corner to the bottom right corner in the Confusion Matrix 9. This suggests a high accuracy and high F1 score in the Conll 2003 Corpus for BiDirectional LSTM+CNN. This statement can be supported by Table 27. However, these results are lower than the results in Table 19(Flair Architecture on Conll 2003 Corpus) and Table 11(GCDT Architecture on Conll 2003 Corpus). This can be deduced from the Confusion Matrices as well because the Confusion Matrix because The diagonal in Confusion Matrix 1 and Confusion Matrix 2 is less spread and more concentrated on the ideal diagonal. On the contrary, the Confusion Matrix 9 involves 2 more misclassifications in the “PER” Tag. Similar to the other architectures tested in this paper, the accuracy and F1 scores decrease as the type of the language and phrases approaches a more fictional and fantasy level. However, in every algorithm, the label “O” is consistent and is classified with %100-%97 accuracy in every task. It can be observed that as the type of the language gets more fictional, the F1 score decreases however the normal Accuracy indeed increases at some points such as 50.69% to 92.51% as shown in Table 31 and Table 33.

6.4 Synthesized Model

Using the detailed results for all of the tested algorithms, it can be said that a combination, synthesis of the three algorithms; GCDT, Bidirectional LSTM+CNN and Flair Architecture; will be a better and more powerful model to identify all types of text accurately. For instance as stated before, GCDT Architecture does slightly better than Flair Architecture in Conll 2003 Corpus however, GCDT Architecture has a lower F1 Score than Flair Architecture in Fictional corpus which suggests that if a new and hybrid model is made using these two algorithms, it can perform better in both of the tasks. Similar to this process, four of the state of art models TENER, GCDT, Flair, BiLSTM+CNN respectively, can be synthesized into one model. Because as the detailed results show, these algorithms can complement each other and this new architecture could potentially tag every type, fictional and nonfictional, of text more accurately.

8. Conclusion:

To conclude, Named Entity Recognition is still one of the most important aspects and building blocks of modern natural language processing and machine learning. Named Entity Recognition aims to label and identify the elements of a sentence; Enamex, Numex, and Timex. Enamex is the term for organisations, names, and places; Numex is the term for numbers, and Timex is the term for time and dates. In this paper 4 state of art machine learning models are trained and tested with different domain texts. Even Though these algorithms are accurate in formal text such as Conll 2003, they are not as good in 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 synthesised model should be trained and tested using the current state of the art models and corpus.

9. Acknowledgements:

I would like to thank,

My research professor Prof. Dr. Sameer Pradhan for guiding me to the right sources and ideas which allowed me to deeply understand my research topic. His continuous feedback on my ideas and my progress on my paper helped me to accomplish and successfully complete this paper. The course that he gave on the first 5 weeks opened my eyes in this field of research. He supported me academically and helped me in my personal growth and thinking process. I am honored and gladful to be his student. He is an amazing leader, teacher, instructor.

Dominick Bruso for his endless support and effort on my completion of this paper. He helped me to understand the system of Pioneer Academics. He answered my questions regarding the process of writing a paper. His continuous feedback and reports helped me to understand what I should improve. I am very thankful to meet such an amazing person and a mentor.

Pioneer Academics for accepting me to this program and giving me an opportunity to explore new areas of study. I am grateful for successfully becoming a Pioneer Academics Alumni. The resources that they shared allowed me to fully understand and explore my research question. The given Seminars taught me the ways to write a great paper. I would like to thank them for their amazing contributions to my research. I am very honored to be a Pioneer Academics Scholar.

10.References:

- Yan, H., Deng, B., Li, X., & Qiu, X. (n.d.). TENER: Adapting Transformer Encoder for Named Entity Recognition. Retrieved November 10, 2019, from https://arxiv.org/abs/1911.04474.
- Akbik, Alan, et al. "Pooled Contextualized Embeddings for Named Entity Recognition." *Proceedings of the 2019 Conference of the North*, 2019, doi:10.18653/v1/n19-1078.
- Chiu, Jason P.c., and Eric Nichols. "Named Entity Recognition with Bidirectional LSTM-CNNs." *Transactions of the Association for Computational Linguistics*, vol. 4, 2016, pp. 357–370., doi:10.1162/tacl_a_00104.

- Liu, Yijin, et al. “GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling.” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, doi:10.18653/v1/p19-1233.
- Chu, Cuong Xuan, et al. “Entyfi.” *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, doi:10.1145/3336191.3371808.
- Sang, Erik F. Tjong Kim, and Fien De Meulder. “Introduction to the CoNLL-2003 Shared Task.” *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* -, 2003, doi:10.3115/1119176.1119195.
- Tolkien, J. R. R. *The Lord of the Rings. the Fellowship of the Ring*. 2007.
- *Doyle, S. A. (2011). Hound of the baskervilles: Another adventure of sherlock holmes. Place of publication not identified: Martino Fine Books.*
- *Ratinov, Lev, and Dan Roth. “Design Challenges and Misconceptions in Named Entity Recognition.” Proceedings of the Thirteenth Conference on Computational Natural Language Learning - CoNLL '09, 2009, doi:10.3115/1596374.1596399.*
- *Nadkarni, Prakash M, et al. “Natural Language Processing: an Introduction.” Journal of the American Medical Informatics Association, vol. 18, no. 5, 2011, pp. 544–551., doi:10.1136/amiajnl-2011-000464.*
- *Straková, Jana, et al. “Neural Architectures for Nested NER through Linearization.” Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, doi:10.18653/v1/p19-1527.*
- *https://github.com/sighsmile/conlleval*
- *https://starwars.fandom.com/wiki/Tatooine*
- *https://github.com/fastnlp/TENER*
- *https://github.com/Adaxry/GCDT*
- *https://github.com/flairNLP/flair*
- *https://github.com/kamalkraj/Named-Entity-Recognition-with-Bidirectional-LSTM-CNNs*

11. Appendix:

The data for this paper can be found in this link.

The code for this paper is in this Github repository.