Journal of Modern Technology and Engineering
Vol.6, No.2, 2021, pp.104-111

COMPARISON OF SPACY AND STANFORD LIBRARIES' PRE-TRAINED DEEP LEARNING MODELS FOR NAMED ENTITY RECOGNITION

Resmiye Nasiboglu[1*], **Mustafa Gencer**[2]

[1]Department of Computer Science, Dokuz Eylul University, Izmir, Turkey
[2]Graduate School of Natural and Applied Sciences, Dokuz Eylul University, Izmir, Turkey

Abstract. Named Entity Recognition (NER) is method that aims to place or classify the named entity in unstructured texts into predefined categories such as person names, institution or organization names or place names. New and popular algorithms such as basic machine learning algorithms, artificial neural network, convolutional neural network or Long Short Term Memory (LSTM) are generally used for NER problems. In this study, it was tried to determine the names of people, locations, organizations, dates and events in around 47 thousand sentences taken from English news articles. Two different trained models were used to find named entities in the dataset. Models are pre-trained Deep Learning models built for NER in the Stanford and Spacy libraries. In this study, structures of the models, the results obtained, and the learning times of the models were discussed comparatively. It is observed that the Spacy library was better at recognizing person names, while the Stanford library was better at recognizing organization and location names. Also, it was observed that only the Spacy library was successful in the dates, events and geopolitical names that it could predict. Moreover, the Spacy library was found to be more efficient than the Stanford library in terms of training time.

Keywords: Named entity recognition, Deep learning, Pre-trained model.

Corresponding author: Resmiye Nasiboglu, Department of Computer Science, Dokuz Eylul University, Izmir, Turkey, e-mail: *resmiye.nasiboglu@deu.edu.tr*

Received: 19 March 2021; Revised: 2 June 2021; Accepted: 8 August 2021; Published: 31 August 2021.

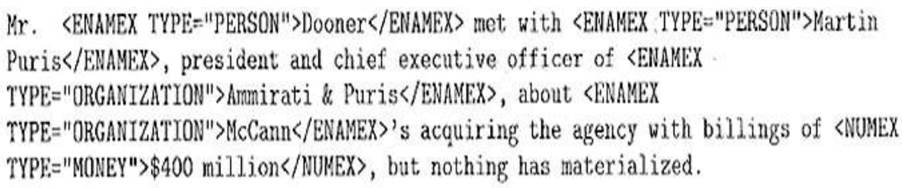
1 Introduction

Named entity recognition (NER) problem is considered as a sub-branch of fields such as data extraction, natural language processing and text mining. Named entity recognition is a tool used to assign classes such as predetermined person name, organization name or place name according to the suitability of entity names in unstructured texts. One of the first studies on the named entity recognition problem was done by Rau (Rau, 1991) in 1991. The author used named entity recognition to find company names in the text. Although different class names are used for naming later, entity name definitions used or put forward in CoNLL 2003 (Conference on Computational Natural Language Learning) and MUC-6 (Message Understanding Conference) conferences have been accepted and used in recent and current studies. In CoNLL, the named entity recognition problem is generally accepted as a classification process for the person's name (Person), place name (Location) and organization name (Organization) mentioned in the text and called ENAMEX (Grishman & Sundheim, 1996). In MUC-6, besides the ENAMEX class, NUMEX (monetary values, numerical values and percentage expressions) and TIMEX (time, date) values are included as new classes in the named entity recognition problem (Krupka, 1995). Apart from these three named entity recognition classes, namely ENAMEX, NUMEX and TIMEX classes, domain-specific entity definitions can also be made for data extraction

Table 1: Entity Examples and Definitions for Named Entity Recognition

TYPE	**DESCRIPTION**
Person	Human, Fictional Character Names
Groups	Nation, Religion, Political Group Names
Organization	Company, Agency, Institute Names
Place	Country, City, State Names
Location	Mountain, Water source, Non-navigation place Names
Product	Automobile, Vehicle, Food Names
Event	Named Hurricane, War, Sports Events Names
Artistic Activity	Book, Song, etc. Names
Legal Document	Names of Documents Named by Law
Language	Any Language Name
Date	Absolute or Relative Dates, Period Names
Time	Names Shorter Than Day
Percentage	Percentages with a percent sign

depending on the scope of the study.



Mr. <ENAMEX TYPE="PERSON">Dooner</ENAMEX> met with <ENAMEX TYPE="PERSON">Martin Puris</ENAMEX>, president and chief executive officer of <ENAMEX TYPE="ORGANIZATION">Ammirati & Puris</ENAMEX>, about <ENAMEX TYPE="ORGANIZATION">McCann</ENAMEX>'s acquiring the agency with billings of <NUMEX TYPE="MONEY">$400 million</NUMEX>, but nothing has materialized.

Figure 1: An example of Named Entity Recognition

Figure 1 provides some example of named entity recognition from the text from the MUC-6 Conference (Krupka, 1995). Here the name "Dooner" after the item "Mr." was named ENAMEX and the word type was determined as PERSON. "Ammirati & Puris" was assigned as ENAMEX again, but ORGANIZATION was determined as the entity type. Finally, "$400 million" was assigned to the NUMEX class and the entity type was determined as MONEY. In the study (Yadav & Bethard, 2019), also the named entity recognition problem is handled, and entities are found in such unstructured texts and assigned to predetermined classes.

In this study, experiments were conducted to identify named entities found in English news articles downloaded from kaggle.com website. In the calculations, the appropriate pre-trained models of the Spacy and the Stanford libraries for solving the NER problem were taken into account and comparative analyzes were made.

In the rest of the article, information about transfer learning, pre-trained model structure, and model evaluation metrics are given in Chapter 2. In Chapter 3, information is given about the used dataset, and also information about the labeling of training set in accordance with the IOB (inside, outside, beginning) format. Experimental results and discussions of these results are given in Chapter 4. The article is completed with the Conclusion section.

2 Method

2.1 Transfer learning

Transfer learning is a popular method of Deep Learning models. It is used most in computer vision because it allows us to build accurate models in a time-saving manner (Pan & Yang, 2010; Rawat & Wang, 2017). With transfer learning, instead of starting the learning process from scratch, you start from patterns learned while solving a different problem. This way you take advantage of previous learning and avoid starting from scratch. Transfer learning is shown schematically in Figure 2.

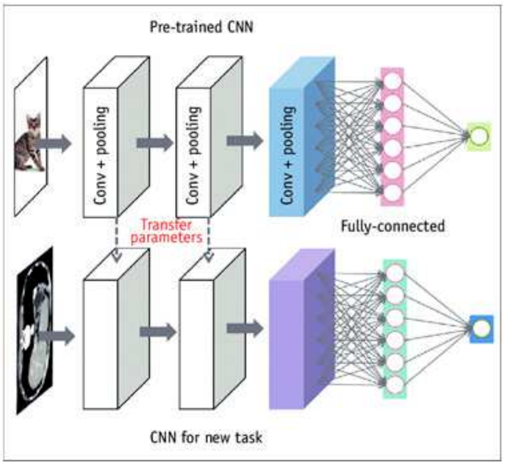


Figure 2: Transfer Learning process

The ready mode used in Transfer Learning is called the pre-trained model. A pre-trained model is a model created by someone else to solve a similar problem. Instead of building a model from scratch to solve a similar problem, you use the model trained on another problem as a starting point. For example, if you want to build a self-learning car, you can spend years building a decent image recognition algorithm from scratch, or you can get the initial model (a pre-trained model) from Google built on ImageNet data to identify images in those images. A pre-trained model may not be 100% accurate in your application, but it saves a lot of effort to reinvent the wheel.

In our study, organization names, personal names, geographical and geopolitical names, date and event names were tried to be determined. In this study, the pre-trained "en_core_web_lg (3.0.0)" model, which is the largest English model, was used as the Deep Learning-based NER model of the Spacy library. Also, the "stanford-ner-4.0.0" model of the Stanford library was used in this study.

Since the labeled words in the dataset, we use labeling according to the IOB (Inside, Outside, Beginning) tag structure, where each word has a definition tag and an IOB tag. But while the Spacy library returns NER model words in bulk, the Stanford library gives each word as a separate tag. As such, the words were tagged with preprocessing from the single IOB structure and inserted into the sentence structure. Then, sentences were predicted using named entity recognition models and the results were compared with labeled data to obtain precision, recall and F1 score values. These values are discussed comparatively in Section 4, where inter-model evaluations were made and models were compared in terms of both time and accuracy.

2.2 NER with the Spacy Library

The Spacy library is a free, open-source library for Natural Language Processing (NLP) developed in Python programming language. When working with too much text, it may be desirable to extract summary information about the text. For example, what is the text about? What does the words mean in context? Who is doing what to whom? Which companies and products are you talking about? Which texts are similar? etc. (Khademi & Fakhredanesh, 2020).

The Spacy library is designed specifically for production use and helps you build applications that process and "understand" large volumes of text. It can be used to build information extraction or natural language comprehension systems, or to preprocess text for deep learning (Yadav & Bethard, 2019). Spacy models generally consist of 3 types. These are; small (sm-small), medium (md-medium), large (lg-large) packages. In this study, the large English package was used for the Spacy library NER model (Room, 2020).

2.3 NER with the Stanford Library

The Stanford NER library is a Java implementation of Named Entity Recognizer. It comes with well-designed feature extractors for Named Entity Recognition and many options for defining feature extractors. Included in the download are well-named entity recognizers in English specifically for the 3 classes (Person, Organization, Location) and also make various other models available for different languages and conditions, including models trained for CoNLL 2003 (Sang & Meulder, 2003).

The library consists of different language packs trained and English is offered as the basic language pack. There are packages in German, French, Spanish, Chinese and Arabic languages other than English. But these packages need to be downloaded separately.

The library consists of 3 different models for English. These are the model trained for 3 classes (Location, Organization, Person), the model trained for 4 classes (Location, Organization, Person, Type) and the model trained for 7 classes (Person, Organization, Location, Percentage, Money, Date, Time). The 4-class model was trained with English training data from the CoNLL 2003 conference. The 7-class model was trained with data from the MUC6 and MUC7 conferences. The 3-class model is trained with these two data sets and additional data.

Stanford NER software is also known as the Conditional Random Fields (CRF) classifier. The software allows designing CRF array models in general (Lafferty et al., 2001). In other words, this code can be used to create sequence models for named entity recognition or any other task by training your own models on labeled data.

2.4 Sequential Classification Model

Our ultimate goal in a sequence classification model is to find the probability of a given set of labels (Y) relative to consecutive input vectors (X). This is denoted as P(Y|X).

1. Training set consists of the input and the target row pairs: $\{(X_i,\, y_i)\}$;
2. The i.th input sequence of vectors is $X_i = \ [x_{i1} \dots\ x_{il}]$;
3. The i.th target sequence of labels is $Y_i = \ [y_{i1} \dots\ y_{il}]$;

For an example (X, Y), we can calculate $P(Y|X)$ in a sequential classification problem by multiplying the probability of each item in the k.th position of the inputs in $k = 1, 2, \dots l$ consecutive times (Song et al., 2018):

$$P(Y|X) = \frac{\exp(\sum_{k=1}^{l} U(x_k, y_k))}{\prod_{k=1}^{l} Z(x_k)}$$

$U(x, y)$ are called emissions or single scores. This is just a score for the y label given to x input vector in the time step k. This can be assumed as the k.th output of an LSTM model. In practice, the x vector is usually a combination of surrounding elements, such as word insertions from a floating window. Each single factor is weighted by a learnable weight in the model. $Z(x)$ is often called the division function. Since we want to take the probabilities, we can think of it as a normalization factor. In this case the sum of scores of each different label must equal to 1. Emissions or single scores (U), represents how likely y_k is given the input x_k. Division function (Z) is the normalization factor to finally get a probability (Song et al., 2018).

2.5 Evaluation metrics

When comparing the models of the libraries, true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values were used. Precision, sensitivity and F1 score values were used to evaluate the models. The definition of these values is given in Figure 3.

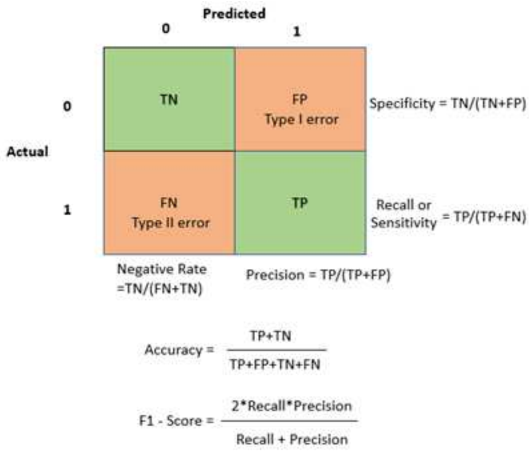


Figure 3: Model evaluation scores

3 Experimental Study

3.1 Dataset

In this study, a dataset containing news articles in English from the "kaggle.com" site was used (www.kaggle.com/abhinavwalia95/entity-annotated-corpus?select= ner_dataset.csv). There are 1354149 words and 47928 sentences in the data set. The data set is labeled in accordance with the GMB (Groningen Meaning Bank) structure, and using the IOB tags. Each word in the data set is given as a sentence in ".csv" format with its label. Labeled words were taken as phrases and sentences, and the named entity recognition were performed with the Spacy and the Stanford libraries. The used dataset includes Geographical Entity, Organization, Person (Person names), Geopolitical Entity (GeoPolitical Entity), Time Indicator, Artifact (Structure names), Event (Important event names), Natural Phenomenon (There are entity names such as Natural Events).

3.2 Groningen Meaning Bank

The Groningen Meaning Bank (GMB) corpus structure, developed at the University of Groningen, consists of thousands of texts in raw and symbolized format, tags for a part of speech, named entities and word categories, and discourse representation structures compatible with first-order logic (Bos, et al., 2017).

3.3 IOB Labeling

As mentioned in the sections above, the IOB format is a common tagging format in computational linguistics for tagging inner, outer, and starting tokens in a stacking task. This method was first presented by Ramshaw and Marcus in their article (Ramshaw & Marcus, 1995). An "I" in front of a label indicates that the label is in a stack. An "O" tag indicates that a token does not belong to any part. The "B" prefix before a tag indicates that the tag is the start of a stack that immediately follows another piece with no "O" tags in between. When a stack comes after the label "O", the first symbol of the stack is prefixed with "B". An example of IOB labeling format is given in Table 2. Another commonly used similar format is the IOB2 format, which is the same as the IOB format except that the B tag is used at the beginning of each stack (i.e. all tracks start with the B tag) (Souza, et al. 2019).

Table 2: IOB labeling structure

Word	**Tag**
John	B-Person
is	O
from	O
United	B-Location
Kingdom	I-Location
.	O

4 Experimental Results and Discussion

The largest versions of the library models were used when performing the named entity recognition application with the dataset. In other words, "en_core_web_lg – 3.0.0" model was used for the Spacy and "stanford-ner-4.0.0" was used for the Stanford libraries. The precision, sensitivity and F1 score values obtained to measure the success of the models are given in the Tables 3-5. Using the Spacy model, the names of entities such as geographical entities, organizations, person names, geopolitical entities, and important event names in the data set were estimated. Using the Stanford model, the entity names such as organization, person and location names were predicted only.

In terms of organization and location names, the Stanford library performed around 3%-4% better on an F1 score basis. Also, the Stanford model performed better in terms of both precision and sensitivity (or recall) in location names. This is because in the Spacy model, when controlling for the "geo" (geo-entity) field for location, there is sometimes confusion between the semantically close geographical entity and the geopolitical entity (Ex: UK and British). Therefore, the Spacy library performed less well in location name prediction than the Stanford library. Also, on the basis of precision score for organization name prediction, the Stanford library provided a better margin of around 6%-7%. Based on this result, it can be said that the Stanford model is more successful in distinguishing organization names from other entity names in this data set.

In recognizing person names, the Spacy model performed better in precision, while the Stanford model performed slightly better in sensitivity. In the F1 score, the Spacy model gave a more successful result around 1%-2%. As a result, for this data set, the Spacy model was found to be more preferable for person name identification. Entity names estimated only by the model of the Spacy library are date, important event names, and geopolitical entity names. For historical data, the F1 score success is around 0.97, while for geopolitical asset names it is around 0.9. However, in this dataset, the Spacy model did not show much success in recognizing important event names, and the F1 score success remained around 0.3. In addition, 47928 sentences were handled in this study. The total prediction time of the Spacy model was around 12 minutes, while the total prediction time of the Stanford library was around 28.9 hours.

The precision, sensitivity and F1 score values of the Stanford and the Spacy models are given separately in Table 3 and Table 4, respectively. Finally, the F1 score values of the two models are shown comparatively in Table 5.

Table 3: Stanford Library Pre-Trained Model Results

	Precision	**Sensitivity**	**F1-score**
Location	0.8933	0.8937	0.8935
Organization	0.8686	0.6884	0.7680
Person	0.7618	0.9405	0.8417
Accuracy	-	-	0.8547
Weighted avg	0.8563	0.8547	0.8512

Table 4: Spacy Library Pre-Trained Model Results

	Precision	**Sensitivity**	**F1-score**
Date	0.9920	0.9498	0.9704
Event	0.2825	0.5206	0.3663
Geo	0.8684	0.8468	0.8575
Gpe	0.9090	0.8913	0.9001
Organization	0.8015	0.6823	0.7371
Person	0.8355	0.8785	0.8564
Accuracy	-	-	0.8446
Weighted avg	0.8762	0.8446	0.8593

Table 5: Comparative F1-Scores for two models

	Spacy NER	**Stanford NER**
Organization (ORG)	0.7371	0.7680
Person (PER)	0.8564	0.8417
Geographical (GEO)	0.8575	0.8935
Geo-Political Entity (GPE)	0.9001	-
Date	0.9704	-
Event	0.3663	-

5 Conclusion

In this study, entity names were recognized by using a data set containing English news articles via the pre-trained deep learning based named entity recognition models of the Spacy and the Stanford libraries. Organization, person, geopolitical, geographical, date and event names were tried to be recognized as entity names.

Models and evaluation results were explained comparatively. The Spacy library was better at recognizing persons' names, while the Stanford library was better at recognizing organization and location names. Success scores were also analyzed for dates, events, and geopolitical names that only the Spacy library could predict.

In general, both libraries showed comparable success in predicting entity names on the dataset we used. However, the total training time of the Spacy model is around 12 minutes, while the total training time of the Stanford library is around 28.9 hours. Therefore, it was observed that there was a significant time difference between the two models.

In further studies, we intend to examine the most effective named entity recognition models on the Turkish texts by creating learning and test datasets with suitable tags.

References

Bos, J., Basile, V., Evang, K., Venhuizen, N., Bjerva, J. (2017). *The Groningen Meaning Bank. In Book: Handbook Of Linguistic Annotation.* Springer, Dordrecht, 463-496.

Grishman, R., Sundheim, B. (1996). *Design of the Muc-6 Evaluation.* New York Univ NY, Dept of Computer Science.

Khademi, M.E., Fakhredanesh, M. (2020). Persian Automatic Text Summarization Based on Named Entity Recognition. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 1-12.

Krupka, G. (1995). SRA: Description of the SRA System as Used For MUC-6. In Sixth Message Understanding Conference (MUC-6). *Proceedings of a Conference Held in Columbia*, Maryland, November 6-8.

Lafferty, J., Mccallum, A., Pereira, F.C. (2001). *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.*

Pan, S.J., Yang, Q. (2010). A Survey on Transfer Learning. IEEE *Transactions on Knowledge and Data Engineering, 22*(10), Pp.1345–1359.

Ramshaw, L., Marcus, M.P. (1995). Text Chunking Using Transformation-Based Learn. *ACL Third Workshop on Very Large Corpora*, June 1995, pp. 82-94.

Rau, L. F. (1991). Extracting Company Names From Text. In Proceedings The Seventh IEEE Conference on Artificial Intelligence Application, pp. 29-30. IEEE Computer Society.

Rawat, W., Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. N*eural Computation, 29*(9), Pp.2352–2449.

Room, C. (2020). Named Entity Recognition. *Algorithms*, *8*(3), 48.

Sang, K,T., Meulder, F. (2003). Introduction To The CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, *Proceedings Of The Seventh Conference on Natural Language Learning At HLT-NAACL* 2003, pp.142-147.

Song, H.J., Jo, B.C., Park, C.Y., Kim, J.D., Kim, Y.S. (2018). Comparison of Named Entity Recognition Methodologies in Biomedical Documents. *Biomedical Engineering Online*, *17*(2), 1-14.

Souza, F., Nogueira, R., Lotufo, R. (2019). Portuguese Named Entity Recognition using BERT-CRF. arXiv preprint arXiv:1909.10649.

Yadav, V., Bethard, S. (2019). A Survey On Recent Advances in Named Entity Recognition From Deep Learning Models. Arxiv Preprint Arxiv:1910.11470