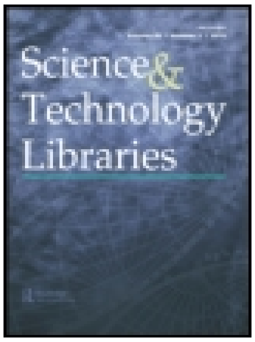

Science & Technology Libraries

ISSN: 0194-262X (Print) 1541-1109 (Online) Journal homepage: https://www.tandfonline.com/loi/wstl20

Named Entity Recognition Approaches and Their Comparison for Custom NER Model

Hemlata Shelar, Gagandeep Kaur, Neha Heda & Poorva Agrawal

To cite this article: Hemlata Shelar, Gagandeep Kaur, Neha Heda & Poorva Agrawal (2020): Named Entity Recognition Approaches and Their Comparison for Custom NER Model, Science & Technology Libraries, DOI: 10.1080/0194262X.2020.1759479

To link to this article: https://doi.org/10.1080/0194262X.2020.1759479

Published online: 19 May 2020.

Routledge
Taylor & Francis Group

Check for updates

Named Entity Recognition Approaches and Their Comparison for Custom NER Model

Hemlata Shelar, Gagandeep Kaur, Neha Heda, and Poorva Agrawal

Department of Computer Science, Symbiosis Institute of Technology, Pune, India

ABSTRACT

Named entity recognition (NER) is a natural language processing tool for information extraction from unstructured text data such as e-mails, newspapers, blogs, etc. NER is the process of identifying nouns like people, place, organization, etc., that are mentioned in the string of the text, sentence, or paragraph. For building the NER system, many different libraries and natural language processing tools using Java, Python, and Cython languages are available. All these tools have pretrained NER models that can be imported, used and can be modified or customized according to requirements. This paper explains different NLP libraries including Python's SpaCy, Apache OpenNLP, and TensorFlow. Some of these libraries provide a pre-build NER model that can be customized. The comparison of these libraries is done based on training accuracy, F-score, prediction time, model size, and ease of training. The training and testing data are the same for all the models. When considering the overall performance of all the models, Python's Spacy gives a higher accuracy and the best result.

KEYWORDS

Named entity recognition (NER); custom named entity recognition (CNER); natural Language Processing (NLP); bio Tagging; spacy; OpenNLP; TensorFlow

Introduction

Today, an enormous amount of unstructured data from different sources like social media, e-mails, news articles, conversation data, etc. is available. This data can be processed to obtain some useful information, though, the information is hard to fetch, and this is one of the greatest challenges of Natural Language Processing. Natural Language Processing is used to understand human language as it is spoken by a computer program. The final objective of NLP is to read, understand, and make sense of the human language in a valuable manner so that machine can use that information (Shah and Bhadka 2017). NLP focuses on figuring out how to resolve the rule of natural language for the machine. Using NLP many processes such as translation, information extraction from unstructured data (like e-mails, newspapers, blogs, etc.), question answering, and text summarization are performed automatically by machines (Sazali, Rahman, and Bakar 2016). Named entity recognition (NER) is the core part of Natural

CONTACT Hemlata Shelar ✉ hemlata.shelar.mtech2018@sitpune.edu.in Department of Computer Science, Symbiosis Institute of Technology, Pune, India

Language Processing. NER is very popular in the field of Information Extraction (IE), Information Retrieval (IR) and Machine Translation (Sazali, Rahman, and Bakar 2016). NER is the process of data extraction that is liable for finding, sorting unstructured text data into categories like names of people, organizations, geographical locations, quantities, expressions of times, percentages, and monetary (Shah and Bhadka 2017). NER plays an important role in the semantic part of NLP, as NER mextracts the keywords or entities from the input statement and obtains the meaning of those entities, sentences, and their relationships. Using previously published information, the NER system identifies keywords like person, institution, geographical location, and time. Basic NER algorithm takes structured and unstructured text as input, processes it and identifies or locates entities (Gürkan et al. 2017). For example, the NER model will locate "Steve Jobs" as a person and it will locate it as a single entity instead of identifying "Steve" and "Jobs" as two different entities.

Named entity recognition

NER is a process in which anything that is denoted by a proper name or tag, for example, a location, an organization, or a person is identified as an entity. Named entities include things like geographical location, date, time, or money, and customization of the NER model for user-defined named entities is also possible. In the following text, named entities are marked:

The Symbiosis Institute of Technology [**ORGANIZATION**] was established in 2008 [**TIME**] by SB Mujumdar [**PERSON**], founding director of the Symbiosis Society Pune [**LOCATION**].

This sentence contains four name entities: one word labeled as **ORGANIZATION**, one word labeled as **TIME**, one word labeled as **PERSON** and one word labeled as **LOCATION**.

Figure 1 shows the basic structure of NER, which is a sequence-tagging task, where fetching of the contextual meaning of words, by using word embedding is done.

More developed NER models can recognize, identify, and classify entities as well. There are many scenarios and use cases of NER technology, like classifying content for news providers, powering content recommendations, customer support, efficient search algorithm, etc. These use cases are domain specific, and the need for identifying words is also different. For example, in classifying content for news providers, the words required to be identified are attack, crime, politics, etc. For the fashion website database search engine, the words required to be identified are clothes, shoes, color, size, etc. For every different requirement, a new NER model can be created or the existing one can be customized to identify some specific words.

There are many different Python and Java-based libraries and tools available for NER. In this paper, we have explained different NLP libraries and

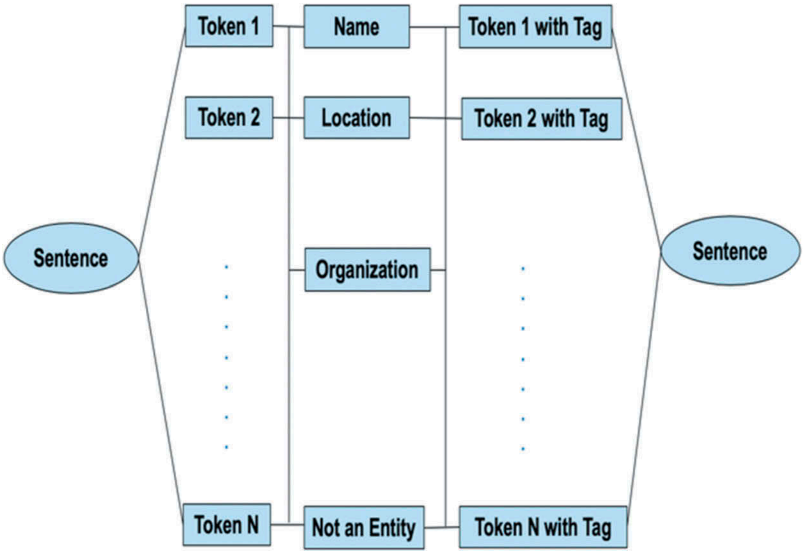

Figure 1. Named entity recognition structure.

tools for NER models including Python's SpaCy, Apache OpenNLP, and TensorFlow and compared their performance. The various parameters used for the comparison of NER tools are time, accuracy, and quality of performance. Different learning methods like supervised learning, semi-supervised learning, and unsupervised learning can be used for NER (Kanya and Ravi 2012). All these techniques use different approaches. For example, supervised learning use support vector machines (SVM), maximum entropy models (ME), hidden Markov models (HMM), decision trees, conditional random fields (CRF), etc. The semi-supervised learning use technique called "bootstrapping," and the unsupervised learning use clustering technique (Patil, Patil, and Pawar 2016).

Various NLP libraries are available for NER. Using these tools, NER models can be built. These libraries are developed by considering some specific languages as a domain for recognition of name entities. The Stanford named entity is built using JAVA, The SpaCy and TensorFlow is a Python-based library (Kanya and Ravi 2012). All these libraries provide inbuilt NER models for some specific entities like person, organization, time, location, etc.

For real-world applications and problem statements, the requirements are different. For different applications, different types of entities are required to be identified by model and that's why there is a need to implement or customize the NER models. When comparing different NER models, not only accuracy is the concern but prediction time, size of model and ease of training are also important factors.

Related work

Atdağ and Labatut 2013 presented a comprehensive comparative study of different NER tools. For this study, a whole new annotated corpus of 247 Wikipedia articles was used. The authors compared four NER tools: Alias-I LingPipe, Stanford NER, OpenCalais NER WS and Illinois NET. They processed precision and recall independently for each type. The precision and recall score of Stanford NER (SNER) was higher than Alias-I LingPipe, OpenCalais NER WS, and Illinois NET. In this study, it was found that the NER model's performance is not uniform over entity types and article categories. The authors compared the overall performance of all four models and their accuracies and found that Stanford NER gave better results.

Dawar, Samuel, and Alvarado 2019 compared the training system, accuracies, and design of three different NER tools, which are Stanford's Named Entity Recognizer, Python's spaCy and IBM Bluemix's Natural Language Understanding tool. The main problem with Bluemix was reading special characters. It organized nouns into organizations and corporations. The problem with Stanford's NER model was the separation of terms. The model tokenized the input data which leads to text deconstructing into single word tokens because of this, a single name like organization name or location name was divided into two different entries that leads to the false prediction of the NER model. This also increased the count of entities and made it more difficult to use the output provided by the model. The spacy gave better results as compared to Bluemix and Stanford NLP. It classifies the entities and also adds layers to the classification. For example, places are contextually separated into three entities like countries, cities, and states.

Reshmi and Balakrishnan 2018 proposed the system architecture for chatbot using Stanford NLP's NER. Stanford NLP's NER model was used to recognize entities like location, person, time, money, etc. All these entities were stored and used to generate response from the chatbot. Besides, they did not customize their Stanford NLP's NER model. The NER model was able to recognize the entities which Stanford's NER model recognizes by default.

Rodriquez et al. 2012 analyzed the comparison of different NER tools. They used the NER models to identify entities directly from the output of the optical character recognition workflow. They used OCR as the input for NER models and compared the recall, precision, and F1 score for tools like OpenNLP, Stanford NER, AlchemyAPI, and OpenCalais. The authors found the F1 score of Stanford NER model more superior followed by AlchemyAPI NER, OpenCalais NER, OpenNLP NER models. From their observation, they also concluded that if they manually correct the text, it will not help to improve the NER's performance.

Dlugolinsky, Ciglan, and Laclav́ ık 2013 selected eight information extraction tools for NER, six of them were NLP tools and two were Wikipedia concept extractors for the evaluation. The evaluation dataset contained four entity types: Location (LOC), Organization (ORG), Person (PER), and Miscellaneous (MISC). The authors found every model gave a different performance for different entity types. For Apache OpenNLP, the authors recorded the worst recall value for LOC and ORG. The MISC was not estimated as Apache OpenNLP cannot classify the entity types subsumed under MISC. The Overall F1 score of Illinois NER for PER, LOC, ORG was better. The model gave poor classification for the MISC entity type. The LingPipe's NER model gave the best recall for ORG but it gave the worst precision as compare to all the evaluated tools. The Open Calais's NER model gave the best precision score for all the three types. The Stanford's NER model gave the second-best F1 score in PER Classification. Among all these NER tools the Open Calais gave the overall best performance for classification. The experiment also showed that each tool is good at discovering different types of entities.

Hemati and Mehler 2019 introduced a new approach for sequence labeling that is LSTM Voter for chemical NER. LSTM Voter is a bidirectional long short-term memory (LSTM) tagger which utilizes a CRF layer in conjunction with attention-based feature modeling. The authors compared the performance of LSTM Voter with other NER models like Stanford NER, MarMot, CRF++, MITIE, and Glample. The authors found LSTM Voter more superior followed by Stanford NER, MarMot, CRF++, MITIE, and Glample.

Different tools and algorithms for named entitiy recognition model

The different techniques and algorithms for creating NER models are:

TensorFlow

The first model employed in the proposed work is TensorFlow with Keras. TensorFlow is an open-source library for dataflow programming. It is a symbolic math library, which is also used for machine learning applications such as neural networks (Fok et al. 2018). NER using neural network is the main area of research today, but here we tried to build a small NER model using TensorFlow and Keras for the research purpose. TensorFlow is a deep learning model and it accepts only numerical data as input rather than text. The process of converting input text data into numerical form is called hot encoding.

SpaCy

SpaCy is an open-source library that is specially built to do many Natural Language Processing (NLP) tasks. NER is one of them, along with dependence parser, lemmatization, entity linker, phrase matcher, sentencization, text classification, part-of-speech tagging, etc. SpaCy comes with pre-build models for a lot of languages. The spaCy tool is offered by Explosion AI, which [redacted]. HMM are generative models, which are used to allocate joint probabilities to paired label sequences and observations. Maximum entropy models (MEMMs) are conditional probabilistic sequence models that can handle long-term dependency and can represent multiple features of a word (Dawar, Samuel, and Alvarado 2019).

Apache OpenNLP

Apache OpenNLP is a Natural Language Processing library. It is a machine learning-based toolkit, which processes the natural language text. It supports the various tasks of Natural Language Processing such as tokenization, chunking, sentence segmentation, part-of-speech tagging, parsing, named entity extraction, and coreference resolution. OpenNLP also includes perceptron-based machine learning and maximum entropy (Dawar, Samuel, and Alvarado 2019). The OpenNLP provides a number of pretrained models for different languages and annotated text resources from which those models are derived.

The Name Finder of Apache OpenNLP can detect named entities in the text. The Name Finder needs a model to detect entities in the sentence. The OpenNLP model is a language-dependent and entity type-dependent model for the entity it is trained for. The OpenNLP projects offer many pretrained name finder models that are trained on different corpora, to find entities from the input text data. The training text data and input text data must be segmented into tokens and sentences. The tokenization for the training data and input text must be the same.

When a model is created, 70% of data are given as training data to model at design time and 30% data is testing data which is given at runtime as shown in Figure 2.

Methodology

In this section, all the techniques used to create the custom NER model for an intelligent search engine for the database are discussed and compared.

The dataset used for this project is the IBM's data set of advance search panel. These datasets contain different possible search queries to predict the search statement. Every NER model takes input in different format therefore three different copies of the same dataset are created with the required format.

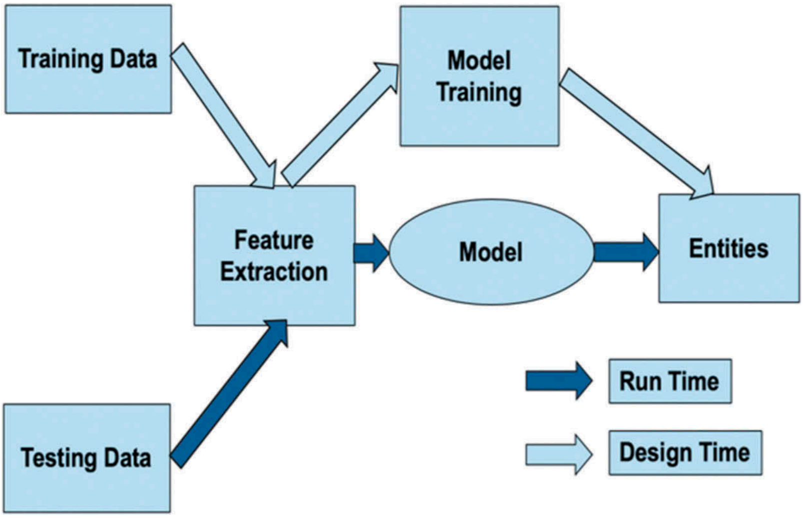

Figure 2. Named entity recognition model.

For intelligent search engine, the NER model classifies tokens as attribute, criteria, and value. The different tools used to build custom NER model are given below:

Custom named entity recognition with TensorFlow

The first step to create the NER model is training data. For training data, we used the IOB tagging format. The IOB format (short for the inside of the text, outside of the text, beginning of the text) is a **common method for assigning names to tokens in** a chunking task in computational linguistics.

Consider the sample text "The Symbiosis Institute of Technology was established in 2008 by S B Mujumdar president and founding director of the Symbiosis Society Pune."

The IOB tagging for this text is shown below:

The -> O
Symbiosis -> B-Organization
Institute -> I- Organization
of -> I- Organization
Technology -> I- Organization
was -> O
established -> O
in -> O
2008 -> B-Time
by -> O
S -> B-Person
B -> I-Person

Mujumdar -> I-Person
president -> O
and -> O
founding -> O
director -> O
of -> O
the -> O
Symbiosis -> B-Organization
Society -> I-Organization
Pune -> B-Location

To create the NER model, we used Keras and Long-Short term memory (LSTM). LSTM algorithm is one of the types of Recurrent Neural Networks which is used to store more contextual information than a simple Recurrent Neural Network. It is intended to recognize patterns in sequences of data, such as handwriting, text, the spoken word, stock markets, and government agencies or numerical time-series data emanating from sensors.

Custom named entity recognition with SpaCy

SpaCy has many in-built models for different languages which can be customized according to different requirements. To create a custom NER model for the intelligent search engine, we have used the SpaCy's 'en' model for the English language and customized the model by adding the labels like attribute, criteria, and value. For adding new labels, we used the add label() method of SpaCy's pipeline.

For training data set, we created the set of tuples as following:

("The Symbiosis Institute of Technology was established in 2008 under the patronage of S B Majumdar president and founding director of the Symbiosis Society Pune", {'entities':[(04,37, 'Organization'),(57, 61, 'Time'),(85, 97, 'Person'),(155, 159, 'Location')]})

Named Entity Recognition with Apache OpenNLP

Apache OpenNLP provides many builds in NER models. For creating the Custom NER model in OpenNLP, we have used Token Name Finder Model. To create a custom NER model, there is no need to add labels like it is done in SpaCy. The labels are only added in the training data using <START:><END> tags.

For the training data set, the following set of tuples were created:

The <START:Organization> Symbiosis Institute of Technology <END>was established in <START:Time> 2008 < END> by <START: Person> S B Majumdar <END> founding director of the Symbiosis Society <START:Location> Pune <END>

Result

TensorFlow gives good accuracy while training the data and prediction accuracy is also good with TensorFlow as shown in Table 1. We discovered that the separation of terms is the major problem with TensorFlow's NER model. It tokenized the text data for input that divides a single entity into two tokens, and it takes that single entity as multiple entries. For instance, when input text contains South Africa as an entry, South and Africa both would be classified as two different entities by the TensorFlow model, which leads to

Table 1. Comparison of NER model's accuracy.

	TensorFlow and Keras	SpaCy	OpenNLP
Training accuracy	0.9955	1.0000	1.0000
Training loss	0.0219	0.00000001427	OpenNLP NER model does not show loss
F1-score	0.9700	1.0000	1.0000
Prediction probability	0.9800	1.0000	0.9969

Table 2. Comparison of NER models.

	TensorFlow and Keras	OpenNLP	SpaCy
Time for prediction	TensorFlow model takes approximately 0.2 µs for prediction	OpenNLP model takes 2 to 3 ms for prediction	SpaCy Model takes approximately 0. 2 µs for prediction
Ease of training	Conversion of text data to numerical data is required for training of the model	Directly works on text data	Directly works on text data
Size of model for X rows	The model size is 2,808 bytes for 15,000 data rows The model size is 2,807 bytes for 30 data rows	The model size is 10,239 bytes for 15,000 data rows The model size is 1,870 bytes for 30 data rows	The model size is 40,10,784 bytes for 15,000 data rows The model size is 41,51,896 bytes for 30 data rows

Table 3. Advantages of TensorFlow with Keras, OpenNLP and SpaCy.

TensorFlow and Keras	OpenNLP	SpaCy
(1) TensorFlow NER model training accuracy is almost 90% most of the time and losses are very low, it means that the model is predicting the entities correct 90% of the time. (2) Prediction time is very less (in ms). (3) Model gives F-score accuracy for every individual tag. (4) Model size is very less and the size of training data does not affect the model size.	(1) OpenNLP NER model gives low accuracy for the wrong prediction. (2) NER model ignores the unknown tokens.	(1) SpaCy's NER Model's prediction time is very less (in ms). (2) After training the model it gives false positive and false negative for each wrong prediction of the training data, which helps to improve the model's performance. (3) NER model ignores the unknown tokens. (4) Losses decrease with every training iteration of training. (5) It gives F-score accuracy for every individual tag.

Table 4. Disadvantages of TensorFlow with Keras, OpenNLP and SpaCy.

TensorFlow and Keras	OpenNLP	SpaCy
(1) Model remembers the sequence of the tags and does not understand the relation between entities and tags. (2) It gives high accuracy for the wrong prediction also.	(1) OpenNLP model does not give F-score accuracy for every individual tag.	(1) The model size is bigger as compared to other models.

increment in the count of entities and makes it significantly more difficult to use the output in a real-world application.

Another problem with TensorFlow is that the model tries to remember the sequence of the labels without considering the relationship between labels and tokens. TensorFlow does not support multi-class annotation. If we label

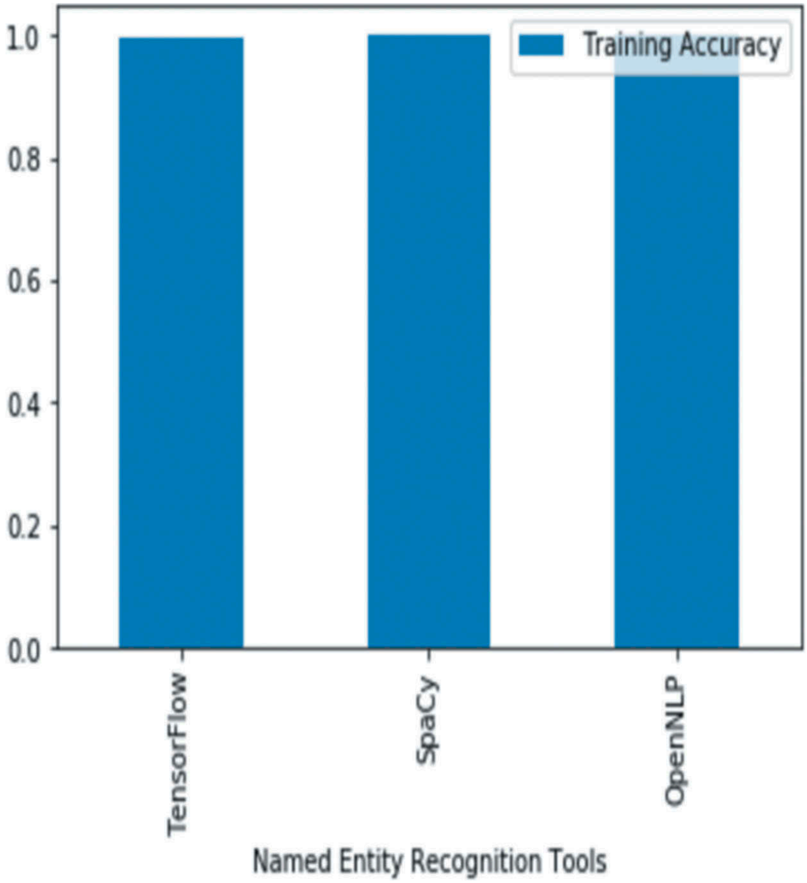

Figure 3. Training accuracy of NER models.

the same token in two different labels, the model will not be able to differentiate and will always predict the wrong label with high accuracy.

OpenNLP gives better accuracy as compared to TensorFlow. It directly works on text data, so there is no extra burden of converting training data to numeric. It gives low accuracy for the wrong prediction which was the issue in TensorFlow. The main problem discovered with OpenNLP NER model is that it also addresses the unknown tokens at the time of prediction which creates the problem at the time of SQL query generation. Also, the model size increases drastically with increasing the size of training data.

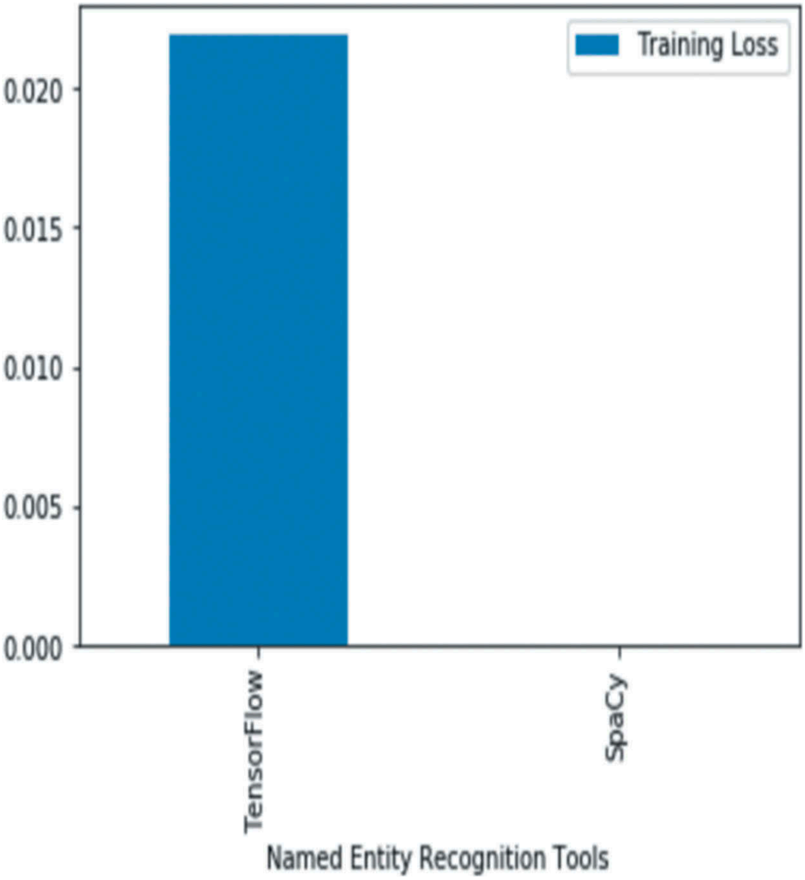

Figure 4. Training loss of NER models.

SpaCy Custom NER model proved to be a better and accurate recognition system with respect to our dataset [redacted] ta as OpenNLP. [redacted] For instance, it gave false positive, false negative, true positive and true negative which helped us to improve our NER model.

[redacted]

Table 2 shows the comparison of these three NER models with respect to different aspects such as how much time they took for prediction, how easy it is to train the model using text data etc. All the advantages and disadvantages of TensorFlow, Apache OpenNLP and SpaCy's NER model is explained in Table 3 and Table 4.

The SpaCy's NER model gives 100% training accuracy. The TensorFlow and OpenNLP approximately give 99% accuracy as shown in Figure 3.

The TensorFlow NER model's loss while training is near 0.0219, and SpaCy NER model's loss is 0.00000001427 which is very less while the OpenNLP model does not show losses as shown in Figure 4.

The OpenNLP and SpaCy NER model give 100% F1-score while TensorFlow gives 97% F1-score as shown in Figure 5.

The SpaCy's NER model gave the best result in classification. The prediction probability of SpaCy's NER model is 100% as shown in Figure 6.

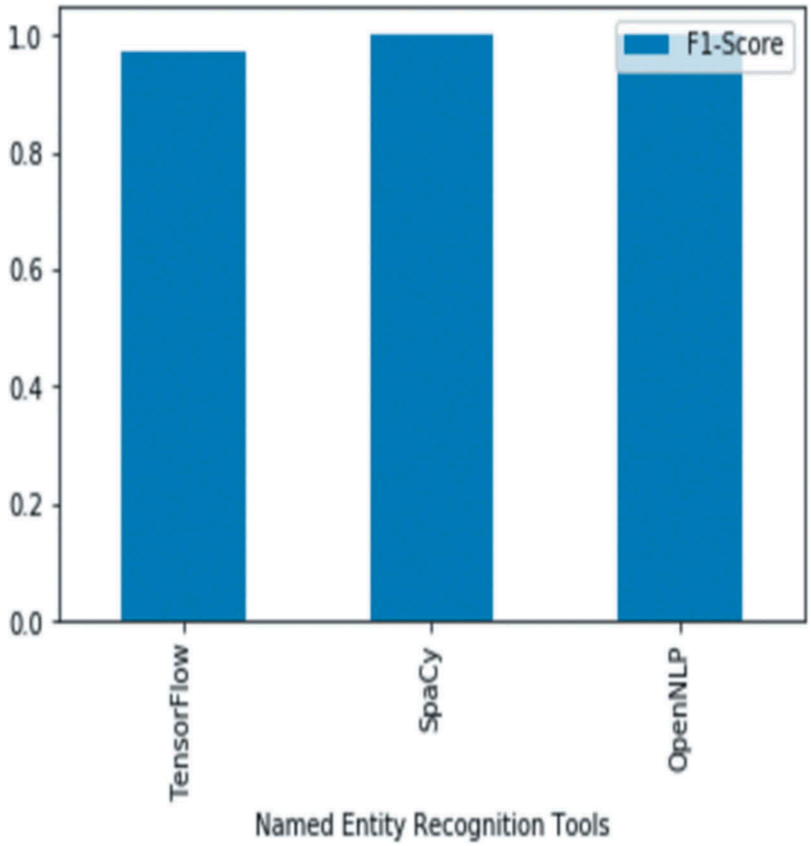

Figure 5. F1-score of NER models.

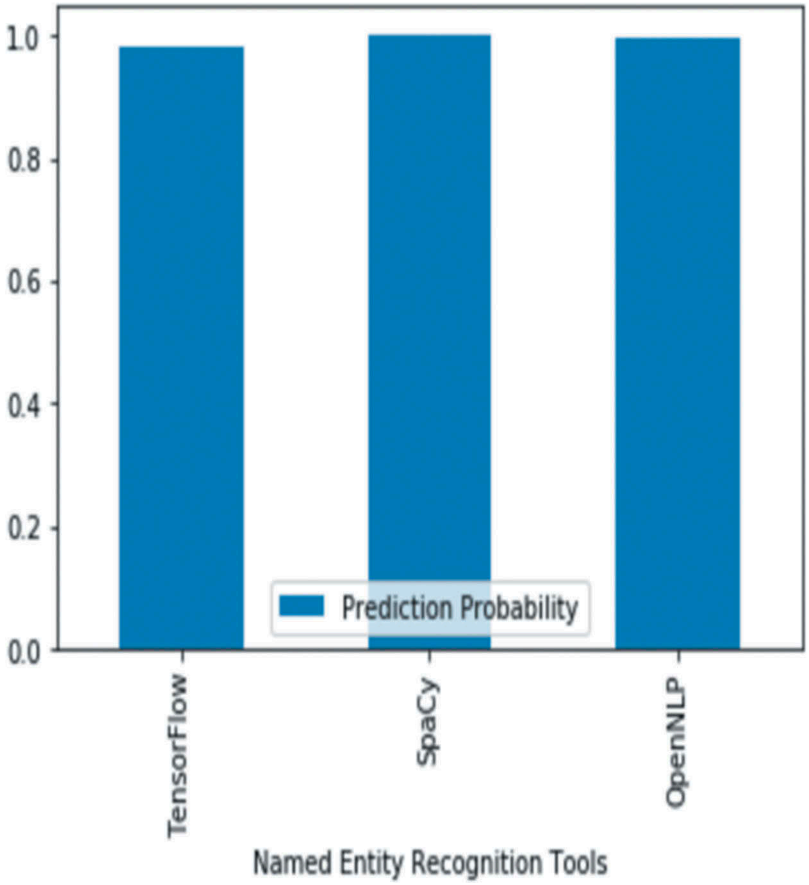

Figure 6. Prediction probability of NER models.

Conclusive summary

In this paper, we have compared the different techniques to create the custom NER model, which are TensorFlow, Apache OpenNLP and Spacy. From our observation of all the models, we found that the [redacted]e. Our work proves that Spacy is better than Apache OpenNLP and TensorFlow for the Custom NER model. As future work, we intend to continue exploring more techniques and libraries for NER.

References

Atdağ, S., and V. Labatut. 2013. A comparison of named entity recognition tools applied to biographical texts. 2nd International Conference on Systems and Computer Science (ICSCS) 228–33. doi:10.1109/IcConSCS.2013.6632052.

Dawar, K., A. J. Samuel, and R. Alvarado. 2019. Comparing topic modeling and named entity recognition techniques for the semantic indexing of a landscape architecture textbook. Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, IEEE.

Dlugolinsky, S., M. Ciglan, and M. Laclav´ık. 2013. Evaluation of named entity recognition tools on microposts. IEEE 17th International Conference on Intelligent Engineering Systems 197–202. doi: 10.1109/INES.2013.6632810

Fok, W. W. T., Y. S. He, H. H. A. Yeung, K. Y. Law, K. H. Cheung, Y. Y. Ai, and P. Ho. 2018. Prediction model for students' future development by deep learning and tensorflow artificial intelligence engine. 4th IEEE International Conference on Information Management 103–06. doi: 10.1109/INFOMAN.2018.8392818

Gürkan, A. T., B. Özenç, I. Çam, B. Avar, G. Ercan, and O. T. Yıldız. 2017. A new approach for named entity recognition. 2nd international conference on computer science and engineering 474–79. doi: 10.1109/UBMK.2017.8093439

Hemati, W., and A. Mehler. 2019. LSTMVoter: Chemical named entity recognition using a conglomerate of sequence labeling tools. *Journal of Cheminformatics* 1–7. doi:10.1186/s13321-018-0327-2.

Kanya, N., and T. Ravi. 2012. Modelings and techniques in named entity recognition-an information extraction task. Third International Conference on Sustainable Energy and Intelligent System VCTW. doi: 10.1049/cp.2012.2199

Patil, N., A. S. Patil, and B. V. Pawar. 2016. Survey of named entity recognition systems with respect to Indian and foreign languages. *International Journal of Computer Applications* 134 (16):21–26. doi:10.5120/ijca2016908197.

Reshmi, S., and K. Balakrishnan. 2018. Enhancing Inquisitiveness of chatbots through NER integration. International Conference on Data Science and Engineering (ICDSE). doi: 10.1109/ICDSE.2018.8527788

Rodriquez, K. J., M. Bryant, T. Blanke, and M. Luszczynska. 2012. Comparison of named entity recognition tools for raw OCR text. Proceedings of KONVENS 410–14. doi: 10.13140/2.1.2850.3045

Sazali, S. S., N. A. Rahman, and Z. A. Bakar. 2016. Information extraction: evaluating named entity recognition from classical malay documents. Third International Conference on Information Retrieval and Knowledge Management 48–53. doi:10.1109/INFRKM.2016.7806333

Shah, D. N., and H. Bhadka. 2017. A survey on various approaches used in named entity recognition for Indian languages. *International Journal of Computer Application* 167 (1):11–18. doi:10.5120/ijca2017913878.