Address entities extraction using named entity recognition

Kanita Krdžalić-Korić
International University of Sarajevo
Sarajevo, Bosnia and Herzegovina
kkrdzalic@ius.edu.ba

Emine Yaman
International University of Sarajevo
Sarajevo, Bosnia and Herzegovina
eyaman@ius.edu.ba

***Abstract*—**Due to presence of large amounts of digital data, many tools for information extraction were developed in order to provide meaningful information and knowledge that could be used in text analysis and interpretation. Machine learning, artificial intelligence and data mining can help there a lot. In this paper, program for extracting address entities is presented as task of named entity recognition. The dataset for named entity recognition are USA addresses that are labeled as one of 8 labels. The model is trained in Python with Tensorflow using pretrained word vectors taken from GloVe-Global vector word embedding. The algorithm that is used is long short-term memory (LSTM) which is special type of recurrent neural network. It was very useful for this application since it takes care of context of the input data. By using this algorithm, model was able to learn how later entities are related to previous ones and thus resolve some complex examples such as differentiating between city and state with the same name.

Keywords—named entity recognition, address extraction, natural language processing, geoparsing

I. INTRODUCTION

Today, there is a lot of digital information present around us. It is impossible to analyze them manually and extract some knowledge from all of them. It is unstructured text that is coming from different websites, articles, e-mails, blogs and news. They usually contain meaningful and useful contents which are hidden to the computers [8]. Social networks had provided people with abilities to express their opinions, share contents and ideas in time and cost efficient ways with many other people in the world. This large amount of text cannot be directly processed by computers [12]. The aim of information extraction is to construct structured data from unstructured text [9]. Natural language processing and information extractors play an important role in extraction useful information from unstructured and semi-structured text sources [8].

A. Natural language processing

Natural language processing (NLP) started in 1950s as combination of artificial intelligence and linguistics [11]. It is a theory-motivated span of computational techniques designed for automatic representation and analysis of human language. For the computers, it is pretty easy to compute number of words in the text or to check the spelling but when it comes to interpreting sentences and extracting meaningful knowledge from them, it becomes very hard for the computer and its abilities are limited. Natural language processing needs high-level symbolic capabilities in order to successfully analyze text. It requires following capabilities:

- Manipulation of recursive structures
- Acquisition of semantic memory
- Control of multiple learning modules
- Identifying the basic objects and actions
- Representation of abstract concepts [12].

Most of the work in natural language processing is done by computer scientists but this field attracted other scientists as well. There are linguists, psychologists and philosophers among those who are interested in this field of artificial intelligence [13]. Computational models in NLP are trying to connect cognitive gap by imitating the human way of processing the language and looking for semantic features that are not explicitly expressed in text [12]. The systems that include NLP processes are used for practical purposes such as enabling machine-human communication. It is present in applications where sentiment analysis has to be done, when meaning of numerous comments in social networks or websites need to be understood. NLP is used in language translation, articles sorting, articles searching, automated assistants for reservations and other similar applications.

B. Named entity recognition

Named entity recognition is part of natural language processing and is used as basis for many applications in Information management such as semantic annotation or question answering. It is a task of extracting and identifying only some types of information elements that are called named entities [6]. Market survey performed by IDC in 2010 [7] showed that amounts of digital information will increase by factor 44 by 2020 and that investments in staff that would manage those information will increase only by 1.4 which represents problem of maintenance of such data. That is why there was a need for tools that would search and discover those amounts of information and give meaning and structure to unstructured data. One of such tools is named entity recognition. Named entity recognition is one of the four tasks of natural language processing. It labels each word in the sentences into categories such as “organization”, “person” or “location”. Some indicator within an entity makes tag that is assigned to the entity. What makes named entity recognition challenging are word/phrase variation of order, derivation of other words using suffixes, change of form of the word (smaller/smallest), synonyms and so on [11]. In order to make computer understand words and their relations and contexts, word are converted into vectors and represented in multidimensional spaces which is meaningful for the computer [10]. Name entity recognition is used in content recommendation, customer support, text classification and many other applications.

II. RELATED WORKS

In [1], spatiotemporal and semantic information were extracted from web news using CNN articles. The content that is extracted was related to natural hazards and helped to provide information about addresses, locations and times of the events that are related to natural disasters. The STS precision that is calculated in this work is the number of correctly resolved spatiotemporal semantic references divided by the number of spatiotemporal semantic references that the system or users attempt to resolve. This precision is higher than ones obtained in previous works related to information extraction. The geographical locations that are collected from the news are sorted into two different groups: explicit locations such as cities and generalized locations such as East Coast. The results from the extraction showed more generalized geographical locations because the source of text were CNN articles and results would show more explicit addresses and locations if local news were used since they provide more detailed geographical information. After those entities are extracted, they were mapped using geographic information systems to represent patterns of natural disasters. The results helped understand the event dynamics, whether certain event is caused by humans or it is an environmental phenomena.

Each day, there is more and more data at World Wide Web and it is increasing with a large speed. The results of it are various and numerous resources for businesses and researchers. The authors of [2] proposed method for extracting geographical locations of commercial companies and services that are available from their websites. They are labeling information that is related to geographical entities. Natural language processing for text analysis is used in annotation process. They relied on pattern matching and clustering for geographical entities. This work is considered useful due to its importance for retrieving information related to geographical entities and locations since there is an increasing attention for such fields over past years. This paper focuses on labeling geographic information from unstructured text from different websites and documents. The emphasis is on commercial entities since understanding names of places can provide great benefit for data mining and searching. One study showed that 15% of questions which are asked on search engines were related to geographic names [3]. The main objective of work from [2] was to present the system that will extract administrative information which includes geographical coordinates and addresses as well. The data that was used is related to all human activities including commercial and research organizations and public administration and it was found on their websites. The areas of artificial intelligence that are used in this work are named entity recognition, natural language processing, text mining and annotation and part-of-speech.

Researchers from [4] recognized the importance of spatial language in text documents and many other applications because combination of unstructured text with structured Geographic information systems provides connection between the two. They say that web pages, blogs, stories, tweets and articles can all take benefit from recognition of geographic terms in texts. In this work, processes of so called geoparsing (recognizing spatial items in text) are discussed as well as challenges related to geoparsing methods and data collection. In this paper, researchers state there are many ambiguities present in natural language, some of them related to toponyms. It can often be seen that some location name is confused with non-location name. The given example for that is location name Paris that can be thought of as Paris in France and Paris Hilton as person. Authors found that another challenge with geoparsing is dealing with misspellings and errors in text documents. The methods for geoparsing that are presented in this paper are Gazetter Lookup Based, Rule based and machine learning based. In the first one, the text is traversed word by word or character by character and searched for toponyms which are previously defined. Those words are then stored in gazetteer which represents database of place names. In the second method, set of predefined rules in certain language decides whether some word is toponym or not. In machine learning approach, text is scanned and set of features are computed. Those features can contain particular strings that appear in place items, length computations, capitalizations and other. Based on training corpus, words that are most highly correlated with toponyms are extracted. Later on, model is run on unannotated text and it decides whether word in the text is toponym or not. This paper gave an overview of tools and techniques used in geographic references extraction.

The research about efficient location extraction algorithm [5] discusses two challenges that appear in location extraction approach. It proposes detection ranking framework which would solve those problems and also introduce the set of new features in mining the contextual information from websites which is usually done using natural language processing. The two problems or challenges that are mentioned are about effect of contextual evidence with aim to improve performance of location extraction and improvement of relation between disambiguation step and named entity recognition step. The location name detection step means solving ambiguity by identifying the meaning that is related to geography and it is done by looking into special words that are common. In the second step, ambiguity is solved by providing the most preferable geographic location to every name that is extracted as location. The researchers showed through experiments that their solution performed much better than the best previous solutions. They designed their solutions in order to effectively and correctly find geographical locations from web sources. Since humans are able to better recognize location entities in context, the solution for computers is designed in this research. It is related to context location prior meaning that they refer to place names that define nearby locations and the other one is related to context word prior meaning that terms that appeared in the context tend to be relevant to the location. They have implemented those extractors on different sets using different solutions and noticed that location context significantly improves accuracy of the extraction. At the end, they have compared their results with famous industrial approaches GeoTagger and Yahoo Placemaker.

III. DATA PREPARATION

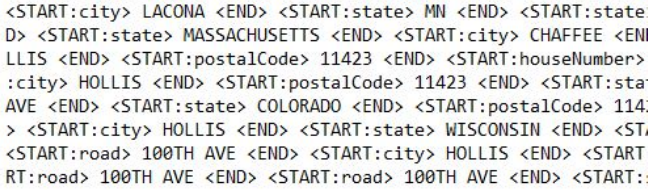
The dataset that is used in this project is set of addresses from USA which came in specific format that had to be preprocessed so that it can be used in named entity recognition algorithm. There were 999765 different addresses in raw data. Some of them included only state, some of them city and postal code. Sometimes address entities were expressed as full name like state name “Minnesota” and sometimes as abbreviations like “MN”. There were also numerous examples of full addresses that were consisted of house number, road, city, postal code and state. In order to use them with named entity recognition, we needed to separate those addresses to words and label them with an appropriate label. In figure 1, there is a snippet from raw data .txt file. It can be seen that each address entity is written between two parentheses with START and END keywords. Additionally, there is a label for each word after keyword START. The following address from raw data will be analyzed in order to present what is done in data preparation.

<START:houseNumber> 193 <END> <START:road> 100TH AVE <END> <START:state> VIRGINIA <END> <START:city> HOLLIS <END> <START:postalCode> 24012 <END>

Each number that is between <START:houseNumber> and <END> is separated and put in new .txt file with label HOUSE_NO representing the house number. The procedure is similar for state, city, road and postal code. All words from addresses are converted to lower case. When this preprocessing is done, the address from the example is stored in new .txt file in the following form that is ready for named entity recognition algorithm:

193 HOUSE_NO
100th ROAD
ave I_ROAD
virginia STATE
hollins CITY
24012 POSTAL

In this example, we can see that road is separated into two parts since it is consisted of two words: “100th” and “ave”. In cases when address entities contain more than only one word, such entity is separated in more words. First word is always labeled as STATE, ROAD or CITY while all rest parts are labeled with the same labels but including “I” in the label such as I_ROAD, I_CITY or I_STATE. Whenever such label is found, it means the labeled word is the extension of the previous address entity. Later on, this is specified in algorithm which learns the context of words related in this way.



```
<START:city> LACONA <END> <START:state> MN <END> <START:state
D> <START:state> MASSACHUSETTS <END> <START:city> CHAFFEE <EN
LLIS <END> <START:postalCode> 11423 <END> <START:houseNumber>
:city> HOLLIS <END> <START:postalCode> 11423 <END> <START:sta
AVE <END> <START:state> COLORADO <END> <START:postalCode> 114
> <START:city> HOLLIS <END> <START:state> WISCONSIN <END> <ST
<START:road> 100TH AVE <END> <START:city> HOLLIS <END> <START
RT:road> 100TH AVE <END> <START:road> 100TH AVE <END> <START:
```

Figure 1 Snippet of raw data

After raw data is preprocessed, it resulted in 8 tags/labels/class attributes. The algorithm in some way classifies each word as one of the labels. The output of the program is always one of the following 8 classes or labels:

- CITY
- I_CITY
- HOUSE_NO
- POSTAL
- ROAD
- I-ROAD
- STATE
- I_STATE

IV. METHODOLOGY

In order to train data for named entity recognition, we need words from dataset to be transferred into format that is readable and understandable by the computer. Due to this reason, our words that represent parts of the addresses are converted to vectors using GloVe, the unsupervised learning algorithm developed by professors from Stanford University. That procedure is called “word to vector” model and is part of word embeddings. After word to vector conversion, our data became 300-dimensional vector where context is also captured which is very important for address entities since it has to discover what entity is city and what is state, for example. Using Tensorflow, the machine learning library in Python, these vectors are trained through recurrent neural network. The type of neural network that is used is LSTM-long short-term memory because it is suitable for capturing the context within the text and is frequently used in language translation as well. The result of this project is program that allows user to enter certain address and the program labels the entities within that address according to 8 class attributes or labels.

A. *Word embeddings and GloVe*

Word embedding which is also known as word representation plays a vital role in producing continuous word vectors that take into serious considerations its context in a large corpus. Word embeddings catch both syntactic and semantic information of words and are useful in measurement of similarities between words which is very important and useful for natural language processing. In reality, several senses of certain word may be correlated and there is not clear boundary between them. [14] Word to vector (Word2vec) is set of related models that are used to make word embeddings. They are consisted of neural networks that are trained on corpus of words in order to capture linguistic contexts of words. The input for such model is large sets of words and the output is vectors which represent each word in multidimensional space and preserve their context. Word2vec uses continuous bag-of-words algorithm which includes representation of context. [15] Besides Word2vec that is developed in Google, there is GloVe model that is developed at Stanford University. This model is global regression model for unsupervised learning of word representations that is better than other models related to word analogy, similarity and named entity recognition tasks. [16] The example of words represented as vectors in multidimensional space is shown in figure 2. It can be seen that model placed cities and their zip codes near each other in the space. It is done because GloVe found the similarity between cities and their codes and captured the context.

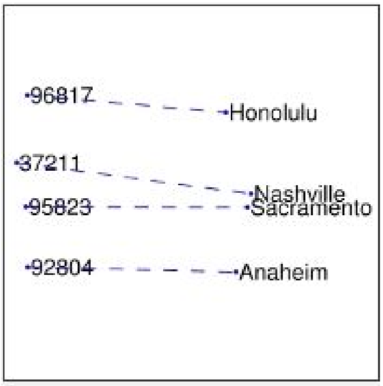


Figure 2 Word representation by GloVe

The pretrained vector of words that is used in this project is 300D GloVe vector. This could be done using this ready-made vector since addresses are coming from English language speaking area and words that are present in address entities are already represented in this pretrained vector.

B. Recurrent neural network and LSTM

A recurrent neural network (RNN) is an extension of feedforward neural network. One of its main characteristics is the ability to handle variable-length sequence inputs. It handles it by recurrent hidden state whose activation is dependent on previous one at each stage. [17] RNN can be represented as multiple copies of the same network where each of the networks is sending the information to the following one. The relation between multiple stages of recurrent neural network can be seen in figure 3. The recurrent neural networks are used in speech recognition, language modeling and translation. The main difference between RNN and feedforward networks is feedback loops that produce the recurrent connection in the network. With the recurrent structure, RNN can model the contextual dependence of text input. [18] This is very useful in text classification since context is important for correct interpretation. It is generally harder to train RNNs due to vanishing gradient and certain errors. These problems are solved with long short-term memory (LSTM) architecture which represents special type of RNN. LSTM memory contains more units which can store and find long range information related to context in a temporal space. [18] LSTMs are capable of learning long-term dependencies. They remember information for long periods and this advantage is used in this project for remembering contexts and other related entities from previous inputs. LSTM overcomes some restrictions and weaknesses of RNNS and is definitely attractive to sequence labeling tasks. [19]

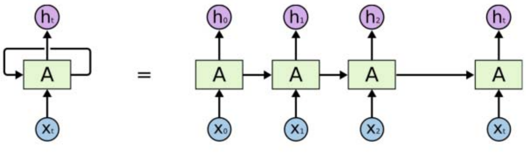


Figure 3 Recurrent neural network schema

The settings that are done in configuration of neural network and model which trained our dataset, number of epochs is set to 1 in order to speed up the process since more epochs for learning require more time for training. Batch size is 20 meaning the model took 20 training examples in each iteration.

V. RESULTS AND DISCUSSION

The model was run on different examples in order to check its performance with various input data. The example of address that was checked is “25 71st St Arizona 11370 East Elmhurst”. The model labeled those entities as follows: HOUSE_NO, ROAD, I_ROAD, STATE, POSTAL, CITY, I_CITY which is actually correct labeling. The model recognized that “St” in address means street and that it is related to road entity so it labeled it as extension of road name.

The interesting part was when address from other country was checked. The address in Bosnia and Herzegovina was used as an example of non-USA address entities. When “Hrasnicka cesta 15, 71210 Ilidza” was entered, the model labeled correctly only some parts of the address, probably guessed them by chance. Only “Hrasnicka” was labeled as road as it really is but all other labels were incorrect. Other address entities from the example were labeled wrongly. The reason for that is most probably the fact that model is trained on USA addresses which are expressed in English language. The model could not recognize the words used in Bosnian address because it was trained using pretrained vector of English words.

The other interesting result is when model labeled “New York, New York” as CITY, I_CITY, STATE, I_STATE which is correct because New York is city in state New York. This is a great result because it is hard for the machine to recognize that state or city is not written twice but that it refers to city within state. This example was correctly labeled in the model which is trained without house number but when house number was added to training data, the performance was not great on this particular example. The same happened with example of “Three Way, Tennessee” where Three Way is confusing city name which would confuse humans as well. The model classified this as ROAD I_ROAD STATE where only state was correct label. This happens only with model that is trained with house number.

The accuracy that is achieved is 99.68 and we can say the model is overfit because it learnt from well structured data and was tested on dataset which is very similar to training dataset structure-wise. When house number is included in training dataset, it does not work that great as accuracy shows. That is why we believe the model is overfit.

VI. CONCLUSION AND FUTURE WORK

Since there are many unstructured data around us, it is precious to provide certain tools which can extract useful information that could be used in text interpretation, searching, sorting, related decision making and so on. This project showed that LSTM is a good choice for this kind of problem since it remembers relationships between current and previously seen entities; which enables taking care of very important thing that is context. This algorithm managed to relate different words within similar context. The importance of named entity recognition tasks and applications is large since it facilitates many operations that should be done on various articles, blogs, mails and news.

This particular project can improve querying databases when it comes to geocoding,-conversion of addresses into latitude and longitude. When certain address entities are labeled, geocoders would easily know in which table they should search for particular entity. This avoids the use of brute force approach in analysis of addresses. Address entities extraction can also be used for simple data extraction when all cities or all states from certain text need to be extracted.

In comparison to other related projects, it can be seen that this application can be used for separated classification of specific topological entities. It is intended to be used for analysis of different addresses which do not have to necessarily contain all parts of the address but can include only cities, only states, postal codes etc. which means it is equally accurate on incomplete addresses. The additional contribution of this project is that it can be used for direct geocoding since it provides an opportunity to query appropriate tables using labeled entities from this application.

In future work, we might try to mix data in dataset and add some characters which are present in real addresses but are not present in training dataset. That might make model learn better on imperfect data. This project might also be applied on other language addresses and for that we would need new corpus and new word vectors that could be trained separately. The another interesting thing that could be done in future work is training model with combination of algorithms and approaches by introducing other types of neural networks as well.

REFERENCES

[1] W. Wang and K. Stewart, “Spatiotemporal and semantic information extraction from Web news reports about natural hazards”, Computers, Environment and Urban systems 50, pp 30-40, 2014.

[2] P. Nesi, G. Pantaleo and M. Tenti, “Geographical localization of web domains and organization addresses recognition by employing natural language processing, Pattern Matching and clustering”, Engineering Applications of Artificial Intelligence, 2016.

[3] M. Sanderson and J. Kohler, “Analyzing geographic queries. In: Workshop on Geographic Information Retrieval (SIGIR)”, 2004.

[4] J. L. Leidner and M. D. Lieberman, “Detecting Geographical References in the Form of Place Names and Associated Spatial Natural Language”, SIGSPATIAL Special, Vol. 3, pp 5-11, 2010.

[5] T. Qin, R. Xiao, L. Fang, X. Xie and L. Zhang, “An Efficient Location Extraction Algorithm by Leveraging Web Contextual Information”, GIS ’10 Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp 53-60, California, 2010.

[6] M. Marrero, J. Urbano, S. Sanchez-Cuadrado, J. Morato and J. M. Gomez-Berbiz, “Named Entity Recognition: Fallacies, challenges and opportunities”, Computer standards and interfaces 35, pp 482-489, 2013.

[7] J. Gantz and D. Reinsel, “The Digital Universe Decade-Are you ready?” IDC iView, 2010.

[8] G. Rizzo and R. Troncy, “NERD: A Framework for Evaluating Named Entity Recognition Tools in the Web of Data”, International Semantic Web Conference, Germany, 2011.

[9] J. Elder, G. Miner, B. Nisbet, D. Delen, A. Fast, T. Hill, “Practical text mining and statistical analysis of non-structured text data applications”, Elsevier, 1st edition, USA, 2012.

[10] R. Collobert, J. Weston, L. Bottou, M.Karlen, K. Kuvakcuoglu, P. Kuksa, “Natural language processing almost from scratch”, Journal of Machine Learning Research 12, pp 2493-2537, 2011.

[11] P. M. Nadkarni, L. Ohno-Machado and W.W. Chapman, “Natural language processing: an introduction”, Journal of the American Medical Informatics Association, Vol 18, pp 544-551, 2011.

[12] E. Cambria and B. White, “Jumping NLP Curves: A Review of Natural Language Processing Research (Review article)”, IEEE Computational Intelligence Magazine, pp 48-57, 2014.

[13] W. G Lehnert and M. H. Ringle, “Strategies for natural language processing”, Psychology Press, New York and London, 2014.

[14] Y. Liu, Z. Liu, T. S. Chua and M. Sun, “Topical word embeddings”, Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015.

[15] J. Lilleberg, Y. Zhu and Y. Zhang, “Support Vector Machines and Word2vec for Text Classification with Semantic Features”, IEEE 14th International Conference on Cognitive Informatics&Cognitive Computing, 2015.

[16] J. Pennington, R. Socher and C. D. Manning, “GloVe: Global Vectors for Word Representation”, Association for Computational Linguistics, 2014.

[17] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, presented in NIPS 2014 Deep Learning and Representation Learning Workshop, 2014.

[18] Y. Du, W. Wang and L. Wang, “Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1110-1118, 2015.

[19] H. Sak, A. Senior and F. Beaufays, “Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling”, INTERSPEECH 15th Annual Conference of the International Speech Communication Association, 2014.