# Resolving Ambiguities in Named Entity Recognition Using Machine Learning

Nitin Bhandari
Department of CS
University of Missouri
Kansas City
nbgm3@mail.umkc.edu

Ritika Chowdri
Department of CS
University of Missouri
Kansas City
rcc92@mail.umkc.edu

Harmeet Singh
Department of CS
University of Missouri
Kansas City
hsp79@mail.umkc.edu

Salim Raza Qureshi
Department of CSE
MIET, Jammu
salim.cse@mietjammu.in

***Abstract*-In this paper, a named entity recognition model is proposed using data from Wikipedia. In every natural language, noun plays an important role. Named entity recognition is the process of identifying and tagging the proper noun in a text and then categorizing them on basis of names, location, product, and others. It has been performed in various languages using different approaches like rule-based, supervised or unsupervised learning. This paper presents a supervised learning algorithm which is used to train the classifier. Different combination rules are applied to the data to increase the performance of the model. Naive Bayes algorithm is also used to calculate the probability of different classes. The aim of this paper is to put forward a distinct approach and using these features analyze the performance measure of the system.**

***Keywords – NER, Natural language processing, Supervised Learning, Naïve Bayes, features.***

## I. INTRODUCTION

Named entity recognition (NER) is one of the essential problems in natural language processing (NLP) related researches, such as information extraction (IE), information retrieval (IR), machine translation (MT), as well as general domain text-to-speech (TTS) synthesis. It aims to classify every word in a document into some predefined categories or “not-a-named-entity”. A ‘named entity’ refers to a word that is not registered in commonly used dictionaries or is a proper noun such as a person’s name, name of a location, or an institution’s name. Named entity recognition refers to the recognition of such named entities within the natural language and categorizes them according to their meaning. Studies on the named entity recognition mainly began with the recognition process of people, location, and institution names. Recently, various named entity categorization system are being developed using various technologies and studied for various applications.

Named entity recognition (NER) classifies the parts of the text to a number of defined classes under different domains[1]. In general, information retrieval system uses words to represent document contents and query. There are two problems to this: ambiguity and different words which represent the same concept [1]. While NER is relatively simple and it is fairly easy to build a system with reasonable performance, there are still a large number of ambiguous cases that make it difficult to attain human performance. There has been a considerable amount of work on NER problem, which aims to address many of these ambiguity, robustness and portability issues. As defined in [McDonald96], there are two kinds of evidences that can be used in NER to solve the ambiguity, robustness and portability problems described above. The first is the internal evidence found within the word and/or word string itself while the second is the external evidence gathered from its context. In order to effectively apply and integrate internal and external evidences, we present a NER system using a HMM [2].

Ambiguity in information retrieval can cause the retrieval of irrelevant documents, while different words which represent the same concept can cause the retrieval system to not find all of the relevant documents. These problems can decrease the information retrieval performance system. In information retrieval, these problems can be addressed with query expansion using senses of the query terms [3].For example in Chinese language, a plain Chinese text does not contain any explicit delimiters such as white spaces in English text to mark word boundaries. Therefore, most NER systems for Chinese must have a pre-processing module for word segmentation. Second, the Chinese language has weak morphology. A plain text in Chinese presents very few exterior morphological hints such as inflexion and capitalization for correct NER. Finally, the Chinese language seems to allow very free word formation. As a consequence, many Chinese named entities may contain out-of-vocabulary (OOV) words, incurring another challenge to Chinese NER [4]. A paper by Guohong Fu on Chinese named Entity Recognition uses a Morpheme-based chunking tagger where morpheme are preferred to character or lexicon words as the basic tokens for entity chunking since morpheme based framework has proven to be more effective than character or lexicon word ones [5].There is always challenge in underlying the process of detection of named entities. This issue becomes more complex when languages other than English are used since every language has its own particularities. Another issue with named entities is that they also belong to the open class of words meaning that new named entities keep getting added to the language with the time [1]. Most of the times named entity recognition uses machine learning approaches. Machine learning approaches have become very popular as now there is more as they are less expensive in maintenance and are easily portable to new languages and domains [2]. The most important task in named entity recognition is emphasize the problem of ambiguity, which often degrades the performance of the system [6]. Evidences found within the word and the context where the word occurs can help to solve these problems. Research in the field of named entity

978-1-5386-4205-4/17 $31.00 © 2017 IEEE
DOI 10.1109/ICNGCIS.2017.24

recognition (NER) in various domains which are selected above has been pointed to major concerns. The computational work over these domains is very limited while in several others domains it has wide range of work available [7].This paper shows a different approach to use supervised learning. Here, firstly the large range of dataset which is edited manually has been trained in both positive and negative sets. Then we used natural language tool kit to convert the sentences into tokens. Then, one of the most common Naive Bayes algorithm is used to calculate the probability of each word occurrences using the likelihood principle. It is discussed in detail in the next two sections. Section II discusses about the related work done in the past on named entity recognition. Methodology used and its approach is discussed in section III. Our results and discussion for the model is described in section IV. The paper is concluded with future work in section V. And, the references are listed in section VI.

## II. Related Work

Generally, there are two most adapted approaches for named entity recognition:

-linguistic approach and

-machine learning approach

Linguistic approach is a classical approach to NER which uses rules manually provided by the linguists. Though it may require a lot of work by domain experts, a named entity recognition system on many rules may provide very high accuracy provided there are no errors in the rule provided. There are several rule-based NER systems, containing mainly lexicalized grammar, gazetteer lists, and list of trigger words, which are capable of providing F-value of 88-92 for English [2], [14], [16]. The main drawback of the linguistic named entity recognition model is they require huge experience and knowledge on the particular language or domain. Also, these systems are not generalized, in fact they are specific to a particular domain and cannot be used for other languages.

The recent Machine Learning (ML) techniques make use of a large amount of annotated data to acquire high-level language knowledge. ML based techniques facilitate the development of recognizers in a very short time. Several ML techniques have been successfully used for the NER task. Here we mention a few NER systems that have used ML techniques. ‘Identifier’ is one of the first generation ML based NER systems which used Hidden Markov Model (HMM) [8]. By using mainly capital letter and digit information, this system achieved F-value of 87.6 on English. Borthwick used MaxEnt in his NER system with lexical information, section information and dictionary features [18]. He had also shown that ML approaches can be combined with hand-coded systems to achieve better performance. He was able to develop a 92% accurate English NER system. Mikheev et al. has also developed a hybrid system containing statistical and hand coded system that achieved F-value of 93.39 [9]. In the recent past, a number of machine learning approaches have been used for named entity recognition such as Hidden Markov Model(HMM) [17], the maximum entropy model (MEM), the support vector machines (SVM), the decision tree machine(DTM).

Much work has been done on Named Entity Recognition in general. Machine learning approach has been a prevailing technique to address this problem. Hai Leong Chieu and Hwee Tou Ng presented a maximum entropy-based name entity recognizer that uses information from the whole document to classify word, with one classifier that uses local context within the sentence as well as uses the context of that word occurring again in the same document to extract useful features to enhance the performance of the named entity recognition. The maximum entropy framework estimates probabilities based on the principle of making as few assumptions as possible, other than the constraints imposed. Such constraints are derived from training data, expressing some relationship between features and outcome. The probability distribution that satisfies the above property is the one with the highest entropy. It is unique, agrees with the maximum-likelihood distribution [10]. Another paper, Morpheme-based Chinese Nested named entity recognition by Guohong Fu and Chunyuan Fu presented a morpheme-based scheme for Chinese Nested named entity recognition. To achieve this task, they employed the logistics regression model to extract multi-level entity morphemes from an entity-tagged corpus and then, explored the variety of lexical features under the framework of conditional random fields to perform Chinese nested named entity recognition [13].

Other work includes a two-phase named entity recognizer based on Support Vector Machines (SVM) which consists of a boundary identification phase and a semantic classification phase of named entities. It mitigates the unbalanced class distribution problem as well as reduces the SVM training cost. This helps in improving the performance of the model.

## III. Method Used and Implementation

Information retrieval is a way of finding useful information from an unstructured data. Named entity recognition is a subtask of information retrieval that aims to find the named entities. It means extracting what is a real-world entity from the text. It has various applications rather than information extraction such as referencing, question and answer system, automatic forwarding, and document and news searching. One way of recognizing named entities in a given text document is Natural Language Processing Tool Kit. It has its own way of categorizing and tagging the parts of speech in any sentence. Parts of speech is often referred to as lexical categories. It characterises different parts of speech very well. But, there can be certain ambiguities while tagging data which can result in errors during performance. These errors are mostly within class errors.

Another way of finding the named entities is to learn the grammar. This can be done by analysing that which parts of speech come before or after noun. Since we have to tag nouns, it can be subject of the verb and appears usually after adjectives or determiners. This might require a lot of time and effort and help of domain experts. So, we propose an algorithm that uses machine learning to do entity recognition.

The system is provided with labelled dataset which can help to make a model that will be able to resolve these ambiguities by classifying them into either of the two classes (positive and negative). The first step of this work is to accumulate the training data. To achieve a reasonable performance, a large amount of annotated corpus is required for training the model. As annotated corpus is not available openly, we have developed a dataset containing approximately 3,000 sentences on name, location and product. Creating the corpus is done by collecting some random text articles in various domains from Wikipedia and other sources. The data is taken manually having three different classes of names, location and product names. We have classified each training data into positive and negative samples, positive referring to the ones that are used in a right context ("Steve Jobs was the co-founder of Apple Inc.") and negative for the ones that are used in some other context ("They were all at their boring 9-5 jobs now"). We have used the natural language processing toolkit of python for working with text processing. It is the part of computer science that deals with human and computer interactions .It is used to process and analyse the text, so that computer can understand human language better. Detecting patterns in a text is an important role of Natural Language Processing. It finds patterns in a data and classifies them into different classes based on their different patterns. Since, we have a lot of data and all the data is in string format so, the first task to do is to break the string into different words i.e. tokens. For recognizing named entities, the text is broken into different tokens using the NLTK function word_tokenize(). To find the probability of the entities with any word we use the probabilistic language models on the training data. To calculate this joint probability, we use the chain rule of probability, but instead of calculating the conditional probability of all the words together, we just calculate the probability of the entity with the word before and after that word. This rule is given by Andrei Markov, called Markov Assumption. It states that the probability of each subsequent state depends on its previous state. The formula is given by:

$$P(Sik|\ Si1, Si2, \ldots, Sik - 1) = P(Sik|\ Sik - 1),$$

where,Si1, Si2, …, Sikare the states when a process moves from one state to another generating a sequence of states. And P(Sik| Sik - 1) is the transition probability of moving from one state to another. So, using this formula in our approach, we can find the words which come before and after a proper noun. Calculating the frequency distribution of those words, we can predict whether a new word is proper noun or not.

Naive Bayes algorithm is used to make predictions for the test data. It is one of the simplest algorithms that is used for text classification. It uses a conditional probability to calculate the probability of an upcoming event using its prior knowledge, given that the likelihood of the all the classes should be equal. Using frequency distribution, we can find which words are likely to occur with which words.

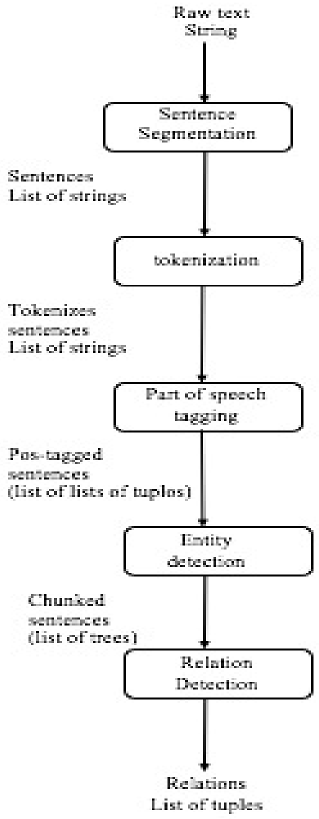

Figure. 1. Simple Pipeline Architecture for Information Extraction.

The amount of natural language text is increasing day by day. So, it becomes difficult to access useful information among all of the text. To find a structure out of all the data given to us, we try to find the entity-relations in our data. Our aim here is to access the structured data from unstructured text, where we can easily find the connection between entities and their relationships.

The natural language tool kit takes sentences and tokenizes, and categorise them into different parts of speech. Then the ones that have been tagged as 'NNP' (Proper nouns) are extracted from it but the problem with NLTK is that it may have many ambiguities. Most of the time it works fine but it might classify some things as a proper noun which are not. Providing a gazetteer could lead to problems such as ambiguities in the tagging. The gazetteer needs to be constantly updated. So, we need to come up with some other way of tagging entities.

One way to reduce the ambiguities is machine learning. Labelled data can be used to train the model for named entity recognition. We have two classes of data, positive and negative, for the correct categorization and for ambiguous data respectively. The model is trained using the training dataset which includes both positive and negative examples and then is asked to classify the testing data into one of the two classes. The training data is broken into tokens and then to find the probability of a word with other words, probabilistic language models are used. In this model we used the Markov model of conditional probability. The data for positive and negative samples should be equal so that their likelihood principles be same. The simplified version of chain rule is used to calculate the joint probability of one word with another word. It is called the N-gram model. It calculates the frequency distribution of any text as a probability of tokens occurring alone or in a sequence. It has unigram, bigram and trigram models that are used in this approach. It is explained below (in table 1). Now once the data is paired into the bigrams and trigrams, the frequency distribution of bigrams and trigrams are calculated.

TABLE 1. Templates and definition of different language models

| Types | Template | Definition |
|---|---|---|
| Unigram | $X_{-1}$, X, $X_1$ | The previous, current and next character |
| Bigrams | $X_{-1}$ X, X $X_1$ | The previous and current character, current and next character |
| Trigrams | $X_{-1}$ X $X_1$ | The previous, current and next character together |

For each pair of words including a proper noun, there is a counter. Naïve Bayes is a simple yet effective method used for the classification of data. It calculates the probability of each group of words belonging to a particular class using the conditional probability to make a prediction. It assumes that the likelihood of a group to belong in either of the classes is same. But still it is a fast and effective algorithm. It has low storage requirements and it is robust to irrelevant features. If some training data has missing label, it ignores the data during training. It is also a good model for text classification and works well in different domains. We calculate the bigram and trigram probability separately and then average the results to calculate the frequency distribution of any particular group. The reason for using both the models is that one can overcome the limitations of other model and can improve the accuracy.

## IV. RESULT AND DISCUSSION

Named entity recognition is an important task. Machine leaning based approach requires annotated data to build the system and requires a huge dataset for better training. The training data is generated manually for each class. 1,000 different examples are taken for name, location and product. We have calculated the precision and recall for every set. Following are the results of different domains separately and then together.

TABLE 2: Precision and recall on development and test set for bigrams

| English Language | Precision | Recall | $F_\beta = 1$ |
|---|---|---|---|
| NAME | 82.67% | 79.23% | 80.95 |
| LOC | 77.59% | 75.24% | 76.41 |
| PRODUCT | 79.57% | 73.29% | 76.43 |
| Overall | 79.94 | 75.92 | 77.93 |

Here we can see that the name class performs well with an accuracy of 80.95%. Location and product have comparatively lower accuracy due to the fact that they are more prone to ambiguities. They can be misinterpreted by the model. This can be overcome by using a variety of training examples on location and product names for each positive and negative class.

TABLE 3 Precision and recall on development and test set for trigrams

| English Language | Precision | Recall | $F_\beta = 1$ |
|---|---|---|---|
| NAME | 80.21% | 76.31% | 78.26 |
| LOC | 79.35% | 77.26% | 78.31 |
| PRODUCT | 78.29% | 76.92% | 77.61 |
| Overall | 79.28 | 76.83 | 78.06 |

We have then averaged the scores from two different models (one using bigrams and another using trigrams), the performance of the system gets better. We achieve an F – score of 85.59%. This shows that averaging the scores improves the performance by a certain percentage as the errors of one model could be reduced by the other model. However this does not work well every time, but most of the time it improves the model.

## V. CONCLUSION AND FUTURE WORK

The machine learning approach requires annotated data and other resources to build named entity recognition model. There are various application of Named Entity Recognition such as automatic summarization, hyperlinking, metadata entity relationship, question answering model, speech recognition and others. Supervised learning is a prominent technique for addressing the NER problem.SL techniques include Hidden Markov Models (HMM), Decision Trees, Maximum Entropy Models (ME), Support Vector Machines SVM) (and Conditional Random Fields (CRF)[21].

The main advantages of the proposed approach are:

- Hidden Markov Model is used to model sequential data, which is a good probabilistic framework[21].
- Naive Bayes approach is simple and fast, works well in domain classification.
- Performance can be improved if the training dataset is increased and a variety of dataset is used to reduce the ambiguities.
- N – Gram models work well or classification problems.

But there are few limitations of our approach:

- The disadvantage of using a supervised learning approach is that it requires a large amount of dataset. So, the model is totally dependent on the training data.
- It still might have some ambiguities within the classes (name vs. product or name vs. location).

## REFERENCES

[1] Talukdar, Gitimoni, Pranjal Protim Borah, and Arup Baruah. "Supervised named entity recognition in Assamese language", 2014 International Conference on Contemporary Computing and Informatics (IC3I), 2014.

[2] McDonald D. 1996. Internal and external evidence in the identification and semantic categorization of proper names. In: B.Boguraev and J. Pustejovsky (eds), Corpus Processing for Lexical Acquisition, pp. 21-39.

[3]R. Krovetz, "Homonym and Polysemy in Information Retrieval,"

in Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistic, 1997.

[4] Guohong Fu , “Chinese Named Entity Recognition using a Morpheme-based Chunking Tagger ”2009I International Conference on Asian Languages Processing.

[5] G. Fu, C. Kit, and J. J. Webster, “Chinese word segmentation as morpheme-based lexical chunking,” Information Sciences, vol. 178, 2008, pp. 2282-2296.

[6]Asif Ekbal, Rajewanul, Haque, Amitava Das, Venkateswarlu Poka and Sivaji Bandyopadhyay,“Language Independent Named Entity Recognition in Indian Languages”, Proceedings of the IJNLP-08 Workshop on NER for South and South East Asian Languages Hyderabad, India, 2008.

[7] Padmaja Sharma, Utpal Sharma and Jugal Kalita, “Suffix Stripping Based NER in Assamese for Location Names”, Computational Intelligence and Signal Processing (CISP), 2012.

[8] Bikel D. M., Miller S, Schwartz R and Weischedel R. 1997. Nymble: A high performance learning name-finder. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 194-201.

[9]Mikheev A, Grover C. and Moens M. 1998. Description of the LTG system used for MUC-7. In Proceedings of the Seventh Message Understanding Conference.

[10] Ki-Joong Lee, Young-Sook Hwang, Seonho Kim, Hae-Chang Rim, “Biomedical named entity recognition using two-phase model based on SVMs”, Journal of Biomedical Informatics 37 (2004) 436–447, 22 July 2004.

[11] David Nadeau and Satoshi Sekine, “A survey of named entity recognition and classification”, Lingvisticae Investigations, Vol. 30, pp. 3-26, 2007.

[12] B. Sasidhar, P. M. Yohan, Dr. A. Vinaya Babu, Dr. A. Govardhan, “A Survey on Named Entity Recognition in Indian Languages with particular reference to Telugu”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[13] ChunyuanFu,and Guohong Fu,” morpheme-based Chinese nested named entity recognition ”, 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)

[14] Grishman R. 1995. Where's the syntax? The New York University MUC-6 System. In: Proceedings of the Sixth Message Understanding Conference.

[15] Sujan Kumar Saha, Sanjay C hatterji, Sandipan Dantapat, Sudeshna Sarkar and Pabitra Mitra , “A Hybrid Approach for Named Entity Recognition in Indian Languages”, Proceedings of the IJNLP-08 Workshop on NER for South and South East Asian Languages Hyderabad, India, 2008.

[16] Wakao T., Gaizauskas R. and Wilks Y. 1996. “Evaluation of an algorithm for the recognition and classification of proper names.” In: Proceedings of COLING-96

[17] Bikel D. M., Miller S, Schwartz R and Weischedel R. 1997. Nymble: A high performance learning name-finder. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 194-201.

[18] Borthwick A. 1999. A Maximum Entropy Approach to Named Entity Recognition. Ph.D. thesis, Computer Science Department, New York University.

[19] Thoudam Doren Singh, , Asif Ekbal and Sivaji Bandyopadhyay, “Named Entity Recognition for Manipuri Using Support Vector Machine”, 23rd Pacific Asia Conference on Language, Information and Computation, pp. 811–818, 2009.

[20] Asif Ekbal and Sivaji Bandyopadhyay, “Bengali Named Entity Recognition using Support Vector Machine”, Proceedings of the IJNLP-08 Workshop on NE R for South and South East Asian Languages Hyderabad, India, 2008.

[21] N. Kanya, Dr. T. Ravi, “Modeling and Techniques in Named Entity Recognition – An Information Extraction Task”, Third International Conference on Sustainable Energy and Intelligent System (SEISCON 2012), Tamil Nadu, India, 27-29 December 2012.