A REVIEW: APPROACHES AND CHALLENGESFOR NAMED ENTITY RECOGNITION FOR INDIAN LANGUAGE TEXT

Mrs. Rupa A.Fadnavis,
Assistant Professor

Department of Information Technology,

Yeshwantrao Chavan college of Engineering, Nagpur, India.

Abstract : Named Entity Recognition (NER) is a task in Information Extraction consisting in identifying and classifying information elements, called Named Entities (NE). It serves as the basis for many other crucial areas like Semantic Annotation, Question Answering, Ontology Population, Opinion Mining, Text summarization.Most of the NER research has been done in English and other European languages. Indian languages have their own constraints for identifying named entities which is a great challenge. This paper discusses about various approaches used for named entity recognition used for identifying entities of various Indian languages.

Index Terms: **-Named Entity Recognition (NER) , Information Extraction, Natural language text.**

I. INTRODUCTION

Named Entity Recognition (NER) is a task in Information Extraction consisting in identifying and classifying information elements, called Named Entities (NE). It serves as the basis for many other crucial areas like Semantic Annotation, Question Answering, Ontology Population, Opinion Mining, Text summarization. It is a process where an algorithm takes a string of text (sentence or paragraph) as an input and identifies relevant nouns (people, places, and organizations) that are mentioned in that string. It is a sub task of information extraction that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. It is also known as entity identification, entity chunking and entity extraction.

The term Named Entity was first used at the 6th Message Understanding Conference (MUC), where it was clear the importance of the semantic identification of people, organizations and localization, as well as numerical expressions such as time and quantities.Most of the research on NER systems has been structured as taking an unannotated block of text,and producing an annotated block of text that highlights the names of entities. The names may consist of one or more tokens, temporal expression ,etc. NER uses a standard set of three metrics to describe the performance of NER system, each for different aspect of the task. These metrics are called precision, recall and F-measure (also F-score or F1 score).State-of-the-art NER systems for English produce near-human performance. For example, the best system entering MUC-7 scored 93.39% of F-measure while human annotators scored 97.60% and 96.95%.

Named Entity Recognition and classification encompasses two main tasks:
1) The identification of entity in a given unstructured text.
2) The classification of these names into predefined entity types, such as Location, Organization, Person, product, time, diseases, sports leagues etc.

II. APPLICATIONS OF NER

The following list mentions few of Named Entity Recognition applications which mostly focus on Natural Language Processing

1) Mostly useful for Search engines in Optimizing Search Engine Algorithms :

To design a search engine algorithm, instead of searching for an entered query across the millions of articles and websites online, a more efficient approach would be to run an NER model on the articles once and store the entities associated with them permanently. The key tags in the search query can then be compared with the tags associated with the website articles for a quick and efficient search.

2) Powering Recommender Systems: NER can be used in developing algorithms for recommender systems which automatically filter relevant content user might be interested in and accordingly assist to discover related and unvisited relevant contents based on our previous behaviour. This may be achieved by extracting the entities associated with the content in users history or previous activity and comparing them with label assigned to other unseen content to filter relevant ones.

3) In the context of Cross-Lingual Information Access Retrieval (CLIR), given a query word, it is very important to find if it is a named entity or not.

4) An amount of information can be examined using named entities, like plotting the popularity of entities over time
5) Mainly used in machine translation. Usually, entities identified as Named Entities and are transliterated as disparate to getting translated.
6) Most of the words indexed in the back index of a book are Named Entities.
7)Automatically Summarizing Text data ,for instance resumes.

III. APPROACHES USED FOR NAMED ENTITY RECOGNITION

Various approaches used for named entity recognition are broadly classified into following types :

1. Manually created rule-based systems
2. Fully automatic learning-based systems
3. Semi supervised learning based systems

1. Manually created rule-based systems :

In this kind of system, developers initially elaborate a set of patterns that will be applied on the text to accurately recognize and tag named entities. The rule-based systems work on the basis of rules created by a domain expert.Early systems made use of hand-crafted rules and pattern matching based on these annotations and formation patterns.The rule-based systems work on the basis of rules created by a domain expert. Nearly all classical MUC systems were using this approach.

The drawback with this approach is that developing hand-crafted rules is laborious and time-consuming. And here the domain plays an important part in creating rules.Rule-based techniques require huge experience and grammatical knowledge of the particular language or domain and these systems are not transferable to other languages or domains.

2. **Fully automatic learning-based systems:** These systems are using Machine Learning (ML) techniques to learn a model in order to accurately tag the texts. The result of the learning task can be a set of rules, a decision tree or a set of numeric data. They can be further divided based on the type of data they use. If the system needs corpus with already labeled entities, then the system uses supervised learning. The system uses unsupervised learning, if it does not use any examples of desired output.

Supervised learning methods make use of training datasets which include classification labels for each data point.A model is trained by feeding it numerous positive and negative classification instances.It then establishes learned rules and leverages them to make predictions on new data. These techniques require the use of a large annotated corpus as a source of training and testing data. Creation of such corpora requires extensive manual effort, and as such, they are not easily available for free.Domain customization for a supervised learning system involves acquiring labeled data for the new domain and learning a new model from scratch. Some of the common machine learning algorithms used for NER are: Hidden Markov models, Support vector machines ,Maximum entropy classifier, Maximum entropy Markov models, Conditional random fields.

Unsupervised Learning
Unsupervised learning techniques deal with the NERC problem by resolving it into a clustering one. They do not require initial training data to be fed to them which makes them a popular approach for resource-starved languages and domains. Clustering is carried out by aggregating named entities into contextual groups. There are no labels provided at the beginning of an unsupervised algorithm. When the algorithm finishes, it outputs groups of entities that share similar features.

3. **Semi supervised learning based systems :** This approach is Relatively recent and are essentially a hybrid of supervised and unsupervised approaches.semi supervised learning algorithms use both labeled and unlabeled corpus to create their own hypothesis. They usually involve a small set of example names referred to as “seeds” being provided to the system. Algorithms typically start with small amount of seed data set and create more hypotheses' using large amount of unlabeled corpus. Sentences containing seed names are then searched, and the system identifies contextual clues that may be common to these names.The system then searches for other names that appear in similar contexts.Once a few such names are found, the learning process is reapplied to the new extended set of names and with each iteration, more names are identified.

IV.EVALUATION METRICS

The Named Entity Recognition performance is always measured in terms of Accuracy (A), Precision (P), Recall(R) and harmonic mean of precision and recall F-Measure (F).In information retrieval , precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of the total amount of relevant instances that were actually retrieved. Both precision and recall are therefore based on an understanding and measure of relevance.In an information retrieval (IR) scenario, the instances are documents and the task is to return a set of relevant documents given a search term; or equivalently, to assign each document to one of two categories, "relevant" and "not relevant". In this case, the "relevant" documents are simply those that belong to the "relevant" category. Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents, while precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search.F1 score is the harmonic mean of these two i.e precision and recall.

V. NER PLATFORMS

Different NER platforms are:
1. **GATE** supports NER across many languages and domains out of the box, usable vigraphical interface and also Java API
2. **OpenNLP** includes rule-based and statistical named-entity recognition

3. Stanford University also has the Stanford Named Entity Recognizer-**NLTK**
4. **SpaCy** features fast statistical NER as well as an open-source named entity visualiz.er
5. **Cogcomp-NER** a state of the art NER tagger that tags plain text with18-label type set (based on the OntoNotes corpus). It uses gazetteers extracted from Wikipedia, word class models derived from unlabeled text, and expressive non-local features.
6. **ParallelDots** is a deep learning powered named entity extraction API.
7. **ChatbotNER** - an open source NER tool tailored for use with chat-bots.
8. **Dataturks** - An online annotation tool to build datasets for NER. Visual UI to easily build domain-specific NER datasets in a format compatible with OpenNLP, CoreNLP etc libraries.

VI. CHALLENGES FOR NAMED ENTITY RECOGNITION IN INDIAN LANGUAGES

For English Language lots of NER system has been built. But such NER system cannot be used for Indian Language because of various issues which are addressed in existing systems . Some of these issues are discussed as following:

1. Ambiguities in named entity classes : -Indian names are ambiguous. Ambiguities in names where the words have multiple interpretations makes NER process difficult while analyzing the text containing words with Named Entities. Word Sense Disambiguation (WSD) need to be used to resolve the ambiguities in the text to determine the classification of a named entity.

2. Abbreviations and non local dependencies:Multiple tokens can be written in different ways such as abbreviations or long form, usually first instance with descriptive long formulation followed by instances with short forms or aliases. Such tokens sometimes require same label assignments or require cross referencing. This ability of a system refers to non-local dependency. External knowledge is required to deal with non-local dependencies. Construction of external knowledge including names lists and expansive lexicons is not easy since domain lexicons and names are continuously expanding.

3. Foreign words: Foreign words in some instances of person, organization, location and miscellaneous names for eg: English words appear in Marathi texts which are spelled in Devanagari script. So recognition of such foreign words is a big challenge. Difficult to create gazetteers that include such names because they are not limited. So Non- availability of large Gazetteer is a challenge.

4. Dialects -Indian language is spoken using many dialects for eg: Marathi is spoken using many dialects such as standard Marathi, Warhadi, Samavedi, Khandeshi, and Malwani in various regions of India. There are specific words used in each dialect to express the text. Words from different dialects are difficult to identify.

5. Encoding related issues: Indian language text is encoded using various fonts and systems. If a document is opened on computer that does not support the font or system using which the document is written, then text in document is displayed with unreadable characters and becomes unusable. Sometimes document created using one computer with a particular operating system cannot fully display the text on computer with same configuration but with another operating system. In such situation some characters are readable but some are displayed using hollow circles or wrongly spelled. This is called as partly corrupted text. Such issues are difficult to handle.

6. Agglutinative and inflectional nature of languages: Indian languages are inflectional and morphologically rich.for eg: Marathi is agglutinative language. Unlike English prefixes and suffixes are added to root words in Marathi to form meaningful contexts[7]. Sometimes in Marathi inclusion of suffixes or prefix to root word leads to change in semantic.So a difficult and critical situation is raised to use gazetteers,dictionaries, similarity measurement and pattern matching techniques to recognize names. Dictionaries or gazetteers contain entities without any suffix added. In Marathi suffixes are added to words in order to create the meaningful context. A well written stemmer is required for morphologically rich language Marathi to separate the root from the suffix in order to compare the word forms with gazetteer or dictionary entries. Next, it cannot be claimed that stemming will solve the problem completely because adding suffixes to roots may change the grammatical category of the root word, which may result in wrong entity recognition.

7. Detection of NEs in raw information as Indian languages do not have capitalization.

8. Named entities can be composed of single or multiple words chunk of text. Parsing prediction or name chunking model is required to predict whether consecutive multiple words belong to same entity.

VI. CONCLUSION

Most of the Named Entity research has been done in English and other European languages. Indian languages have their own constraints for identifying named entities like non- availability of annotated corpora, agglutinative nature , demanding morphology and no concept of capitalization and many more which is a great challenge. So A system need to be developed which will be able to classify various named entities of Indian languages into different categories which will aid in various Natural Language Processing applications like question answering , text summarization etc.

REFERENCES

[1] Monica Marrero ,Julián Urbano Sonia Sánchez,Cuadrado JorgeMorato Juan Miguel Gómez-Berbís ,September 2013,Named Entity Recognition: Fallacies, challenges and opportunities, Computer Standards & Interfaces,ELSEVIER, Volume 35, Issue 5, Pages 482-489

[2] H.Ji and R.Grishman, 2011, p"Knowledge Base Population: Successful Approaches and Challenges," in Annual Meeting of the Association for Computational Linguistics, p. 1148–1158.

[3]Asif Ekbal and Sriparna Saha. Classifier ensemble selection using genetic algorithm for named entity recognition. Research on Language & Computation, 8:73{99, 2010. 10.1007/s11168-010-9071-0.

[4]M. Marrero, S. Sanchez-Cuadrado, J. Urbano, J. Morato, and J. A. Moreiro, "Information Retrieval Systems Adapted to the Biomedical Domain," 2012. ACM Computing Research Repository,

[5]M. Marrero, S. Sánchez-Cuadrado, J. Morato Lara, and Y. Andreadakis, 2009."Evaluation of Named Entity Extraction Systems," in 10th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'09),

[6] Karthik Gali, Harshit Surana, Ashwini Vaidya, Praneeth Shishtla and Dipti Misra Sharma,2008Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition, Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, pages 25–32,Hyderabad, India, January 2008. Asian Federation of Natural Language Processing

[7] Survey of named entity recognition systems with respect to Indian and foreign languages,January 2016

[8] N Patil, AS Patil, BV Pawar - International Journal of Computer Applications (0975 – 8887) Volume 134 – No.16,

[8] Padmaja Sharma1, Utpal Sharma1 and Jugal Kalita,September 2010, The first Steps towards Assamese Named Entity Recognition ,22 Brisbane Convention Center Brisbane Australia

[9]Satoshi Sekine , Chikashi Nobata May,2004 "Definition, dictionaries and tagger for Extended Named Entity Hierarchy" Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04),,Lisbon, Portugal

[9] Rakhi Joon and Archana Singhal, February 2017.ANALYSIS OF MWES IN HINDI TEXT USING NLTK, International Journal on Natural Language Computing (IJNLC) Vol. 6, No.1,

[11] Chenliang Li, Aixin Sun, Jianshu Weng, and Qi He ,Tweet Segmentation and its Application to

Named Entity Recognition DOI 10.1109/TKDE.2014.2327042, IEEE Transactions on Knowledge and Data Engineering

[12] . Leon Derczynski a,*. , Diana Maynard, Giuseppe Rizzo b,d, Marieke van Erp c,. Genevieve Gorrell .27 Oct 2014Analysis of named entity recognition and linking for tweets. arXiv:1410.7182v1 [cs.CL]