International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
IRJET Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Named Entity Recognition on legal text for secondary dataset

Vikas Mastud[1], Roshan Jaiswal[2]

[1]*Student, Dept. of Institute of Computer Science, MET College, Maharashtra, India*
[2]*Assistant Professor, Dept. of Institute of Computer Science, MET College, Maharashtra, India*

---***---

Abstract - The legal judgment documents contain valuable information which can be used for discovering unknown patterns from it. but Legal texts are full of challenges; they contain many typos which include space between a single word, two words merged. In India, each geographic region has its style and language to write judgment documents. Judgment documents are free text and real-life data, and these data sets are very hard to clean and process.

This paper describes an approach to creating a secondary dataset using Named Entity Recognition (NER) in English language documents from the legal domain. The aim is the extraction of entity classes: person name, judge name, lawyer name, country, city, street, landscape, company, organization, court, brand, institution, law, ordinance, court decision, and legal literature. For performing Named Entity Recognition transfer learning is used. Frameworks and libraries used and tested for better accuracy are FlairNLP, Bert, AllenNLP, Spacy, NLTK.

***Key Words*: NER, NLP, legal, Named Entity Recognition.**

1. INTRODUCTION

For Indian languages, NER is a difficult task. This can be attributed to various reasons like Some noun words are missing with capitalized letters which make them difficult to recognize. Let's take an example like “Hi Mr. Ram, what are you doing”, in this sentence “Mr. Ram” is a person named entity, if we remove Mr. from the sentence and remove capitalization of Ram “Hi ram, what are you doing” it will become more complex to detect. And now we get a new kind of challenging task for our NER model. The judgment document also contains named entity ambiguities for example S.T. Bus is used to represent a vehicle but it is detected as a Person Name. It is difficult to identify an entity depending on the subject's context. and it is difficult to create a general solution for NER because in India each geographic region has its style and regional language to write judgment documents. If we switch regions we have to write new rules and we need to use different language models.

The latest research in Deep Learning and Natural language processing achieves good accuracy on NER tasks. Models like FlairNLP, AllenNLP, Bert-NER, and Spacy can perform very well in the Indian context. and regular expression for an entity that contains a fixed pattern.

2. NER

Named-entity recognition (NER) is the Extraction or identification of words as a predefined category like person names, currency, location, organizations, in the text.

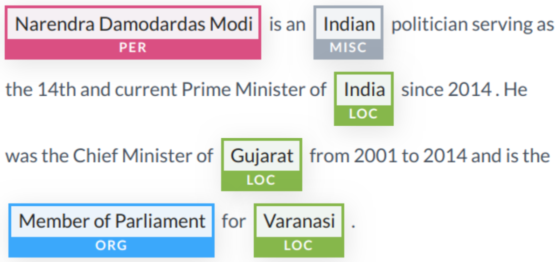

Fig-1: Named Entity Recognition

Fig-1 is showing the highlighted Named entities in paragraph.

2.1 NER dataset

Entity Recognition Datasets: A Structured Dataset for named entity recognition tasks. These annotated datasets cover the range of languages, domain, and entity types.
Here is a demo from CoNLL 2003

U.N. NNP I-NP I-ORG
official NN I-NP O
Ram NNP I-NP I-PER
heads VBZ I-VP O
for IN I-PP O
Baghdad NNP I-NP I-LOC

In the above example, the first word on every line is a word from the corpus, second is part-of-speech (POS) tag, the third is syntactic chunk tag and fourth is the name entity tag for that word. Every line in the dataset follows this pattern [word] [POS tag] [chunk tag] [NER tag]. Entities are annotated with **ORG** (organization), **LOC** (location), **PER** (person), and **MISC** (miscellaneous), O (other). The chunk tags and the named entity tags have the format I-TYPE which means that the word is inside a phrase of type TYPE. Only if two phrases of the same type immediately follow each other, the first word of the second phrase will have tag B-TYPE to show that it starts a new phrase. A word with tag O is not part of a phrase. Here is an example Similarly, there are other types of datasets available for ner for different regional languages in India.

2.2 Named Entity Recognition (NER) Frameworks:

Off-the-shelf Open Source NER Frameworks and tools offered by academia and industry projects.

Table-1: Frameworks and tools for NER

NER tools	URL
transformer	https://huggingface.co/models
flairNLP	https://github.com/flairNLP/flair
Bert	https://github.com/google-research/bert
AllenNLP	https://demo.allennlp.org/
deeppavlov	http://deeppavlov.ai/
Gluon	https://gluon-nlp.mxnet.io/model_zoo/ner/index.html
spaCy	https://spacy.io/api/entityrecognizer
NLTK	https://www.nltk.org
Polyglot	https://polyglot.readthedocs.io/en/latest/
OpenNLP	https://opennlp.apache.org/

3. Secondary Dataset from legal documents:

Legal domain is very big and divided into sub domains. Here we are taking examples on motor vehicle act cases for creating dataset and the same can be applied on other domains as well. Here we are considering only the judgment document containing all the necessary information about the case. Following are the Name Entities which can be extracted from the judgment document.

Table-2: Named Entity List

Named Entity List
claimant_annual_income
claimant_age
claimant sex
claimant education level
claimant occupation
claimant relationship
injury_type
claim_amount
judge_decision
Awarded amount
Driving license
Medical report available
Medical help from
Disability percentage
Claimed amount
Rewarded amount
Capital gains
Capital loss
Incident date
Incident type
Collision type
Incident severity
Authorities contacted
Incident state
Incident city
Incident location
Incident hour of the day
Number of vehicles involved
Property damage
Bodily injuries witnesses
Police report available
Total claim amount
Property claim

Vehicle claim
Vehicle Brand
Vehicle model
Vehicle year

4. Methodology and approach:

It's always better to take a sample from one particular district and language of your choice. Here we are taking a sample dataset from one particular district with few documents in the English language.

Table-3: Entities for Extraction

claimant_annual_income
claimant_age
injury_type
claim_amount
judge_decision
Awarded amount
Driving license
Medical report available
Medical help from
Disability percentage

Entities from **Table-3** are selected for extraction from judgment documents. For extraction complex entity transfer learning is used. and the entity which has any pattern like date, currency can be extracted by a regular expression.

4.1 algorithms:

Input: legal document X

Output : {category: Name Entity}

```
0: result = list()
1: data = Preprocess(X)
2.for class in list_of_entity_class:
3:        data = ReduceSearchSpace(data, class)
4:        result.add( NER(data))
```

Step-1: Preprocess:

In this process we remove some special characters and stop words. there will be some spelling mistakes that need to be fixed and for that spelling correction step.

Step-2: Search space reduction:

Legal documents are more or less in pages, we don't need to pass the entire document in Step3 NER function. Instead of that, we can pass essential parts of the document for a particular type of entity. So we are iterating this list list_of_entity_class and each class goes as a second parameter in the ReduceSearchSpace() function . The implementation of this function depends on the geographic region of court and rule base system or regular expression. So when regions change structure and writing style of judgment documents also change according to that we need to write rules for extracting essential parts of the judgment document.

Step-3: NER and Performance evaluation:

In this function, we pass a paragraph for extracting entities from it. This function uses the FlairNLP library for NER tasks. The performance of this function depends on its implementation, Hardware of system, and size of the paragraph. If the size of the paragraph is big it will take more time and if the size of the paragraph is small it will take less time. In the case of a GPU, it will run faster as compared to CPU. And most important the model which we are using.

5. Result:

After running the algorithm over a few documents with FlairNLP, AllenNLP, Deeppalvo, Bert, Spacy models we can say that FlairNLP gives more accurate results than other deep learning models. Because FlairNLP is trained on Multilingual text. AllenNLP is the second-best model for NER, there is Bert with a multilingual model it is also good but after accuracy comparison fairNLP is best.

Table-4: sample secondary dataset

annual_income	age	injury	claim_amount	judge_decision	awarded_amount	driving_license	medical_report	medical_help	disability
50000	61	permanently_disabled	48787	allowed	180000	Yes	Yes	hospital	20
30000	50	permanently_disabled	75000	rejected	0	Yes	No	clinic	0

62760	36	partial_disabled	200000	allowed	72000	Yes	Yes	hospital	13
60000	25	partial_disabled	200000	allowed	127167	Yes	Yes	clinic	8
24000	18	partial_disabled	400000	allowed	96080	No	Yes	hospital	19
15000	62	permanently_disabled	25000	allowed	25000	No	Yes	hospital	15

Table-4 shows the first 5 records from a secondary dataset created from a judgment document. There are some missing, and wrong values present in the secondary dataset. But that can be solved by retraining the bottom layer of the used transfer learning model with our own dataset.

6. CONCLUSION

Extracting Name entity from a judgment document is a difficult task. The dataset generated from this method is real data and can be used for discovering unknown patterns in fraud and crime. Insights from these datasets will be helpful for decision making and system improvements. The same method with little change can be applied to other legal acts like cyber, property, family, tax, etc.

REFERENCES

[1] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," Lingvist. Investig., vol. 30, no. 1, pp. 3–26, 2007.

[2] P. Cheng and K. Erk, "Attending to entities for better text understanding," arXiv preprint arXiv:1911.04361, 2019.

[3] D. M. Aliod, M. van Zaanen, and D. Smith, "Named entity recognition for question answering," in ALTA, 2006, pp. 51–58.

[4] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," Trans. Assoc. Comput. Linguist., pp. 357–370, 2016

[5] X. Dai, "Recognizing complex entity mentions: A review and future directions," in ACL, 2018, pp. 37–44

[6] R. Sharnagat, "Named entity recognition: A literature survey," Center For Indian Language Technology, 2014.

[7] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in COLING, 2018, pp. 2145–2158

[8] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," arXiv preprint arXiv:1508.01991, 2015.