Nafisa Ali Amir

David Yarowsky

[Course Number]

April 18, 2019

Legal Information Retrieval: Precedence Retrieval using Catchphrase extraction

Introduction:

- Legal domain - lot of text and less worked

- Annotated Dataset from Indian Supreme Court cases (Open to other legal datasets if available as annotated)

- Tasks many, but selected catch phrase extraction and Precedence retrieval

- Models implementation based on machine learning/ neural networks

- Compare the performance with standard vector-based model and with prior works on the same Dataset

- Use pre-trained word Embeddings

- Use average of word embeddings to create phrase embeddings

- Empirical Evaluation using precision, recall and f1 measure

Dataset: FIRE 2017 Legal information retrieval

Task1: Catchphrase extraction (Train on 100 cases and test on 300 cases)

Step1- Noun phrase extraction using POS tagger

Create Training set as follows:

Option1:

Let X be the training data of shape N by M, where N is the number of noun phrases in all the training documents and M is length of phrase feature vector for noun phrase.

Terms = Bag of noun phrases extracted

For each term in Terms:

Compute phrase feature vector = average of word embeddings (words in the phrase)

Option2:

Slight deviation from option1,

Terms = bag of noun-phrases along with frequency with which they appear in a document.

Eg, Terms = [land-1, land-2, high-court-2, proceedings-4, proceedings-6]

For each term in Terms:

Compute phrase feature vector = [average of word embeddings] + [freq]

Step2: Train the model

Use a machine learning method to output the probability of a phrase/term to be a Catchphrase candidate

Step3: Validation/Test method

For each document d in the validation set:

Extract noun-phrases (with frequency for option2) from d;

For each term in noun phrase/withfreq:

pred = model.predict(term)

if the term is predicted as a candidate then append to the list of Catchphrases[d]

Step 4: Evaluation:

Let G be the set of true catchphrases for document i,

Let O be the set of predicted catchphrases for the document i.

Precision[i] = #{O intersection G}/#{O}

Recall[i] = #{O intersection G}/#{G}

F1[i] = 2 \* Precision[i] \* Recall[i] / (Precision[i]+Recall[i])

Overall Precision = mean(Precision)

Overall Recall = mean(Recall)

Overall F1 = mean(F1)

Task2: Preceedence Retrieval (200 cases and 2000 prior cases to look up)

Step 1: Catchphrase extraction of the input cases (queries) and prior cases (documents)

Step 2: Compute vectors for all documents and queries

Step 3: For each query:

compute similarity of vectors

Rank based on similarity

Extract 5 best ranks

Use machine learning to improve the ranking

Reference-

Overview of the FIRE 2017 IRLeD Track: Information Retrieval from Legal Documents

Improved Automatic Keyword Extraction Given More Linguistic Knowledge

A Simple but Tough-to-Beat Baseline for Sentence Embeddings

Works Cited

AuthorLastName, FirstName. *Title of the Book Being Referenced*. City Name: Name of Publisher, Year. Type of Medium (e.g. Print).

LastName, First, Middle. "Article Title." *Journal Title* (Year): Pages From - To. Print.