Information Extraction

Motivation

NEED

The amount of text generated nowadays is increasing at an unprecedented rate. Extracting knowledge from unstructured natural language text automatically in a structured way, such that it can be processed, is crucial more than ever. Addressing this problem is key to improving several essential tasks such as question answering, text entailment, document similarity, knowledge-base population Etc.

OBJECTIVE

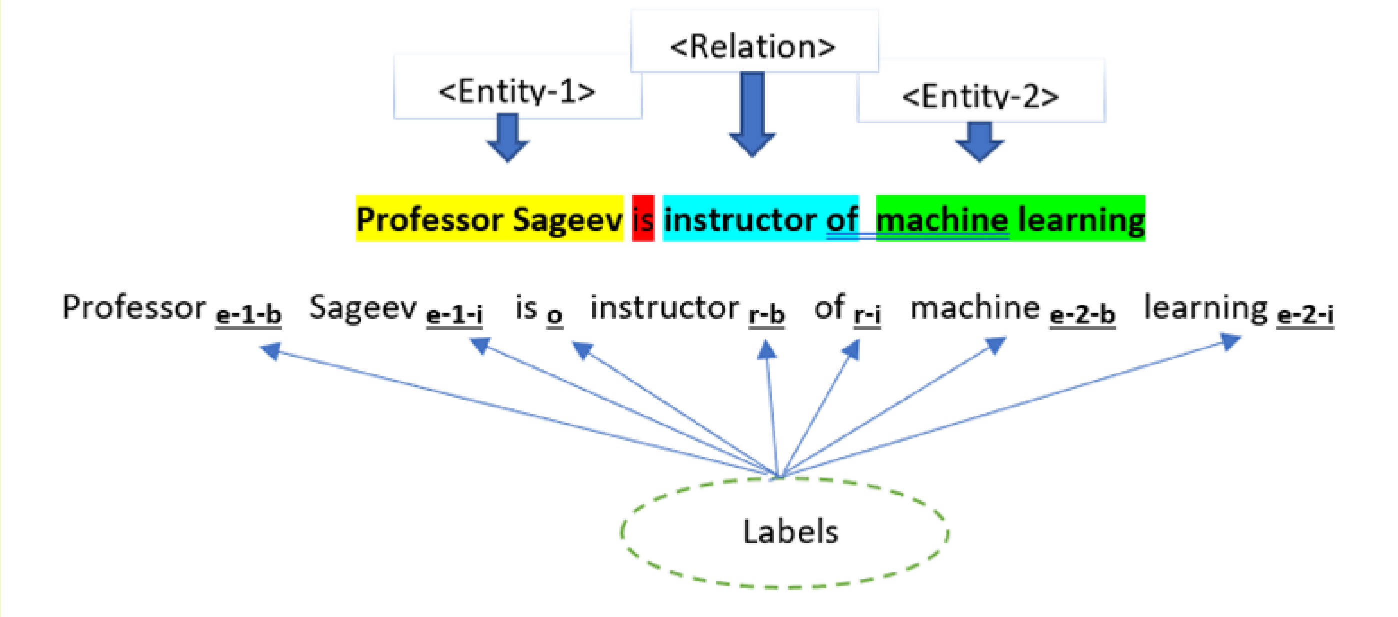
Attempt to extract propositions from a sentence in the form

<<Entity-1> <Relation> <Entity-2>>

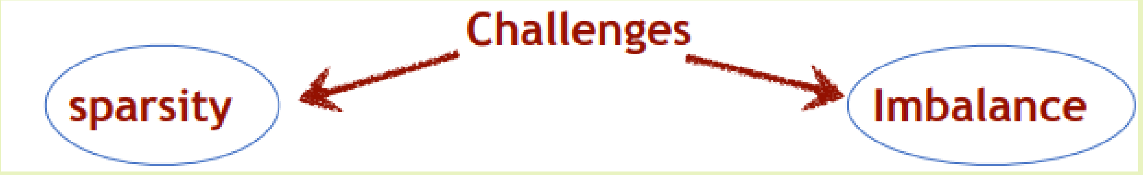
by modelling it as a sequence labelling problem using several approaches and comparing them.

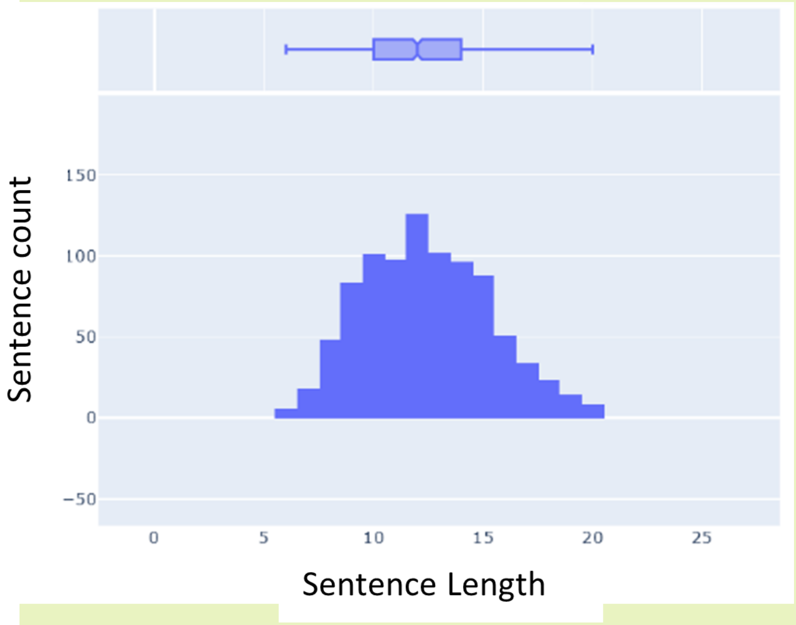
Dataset

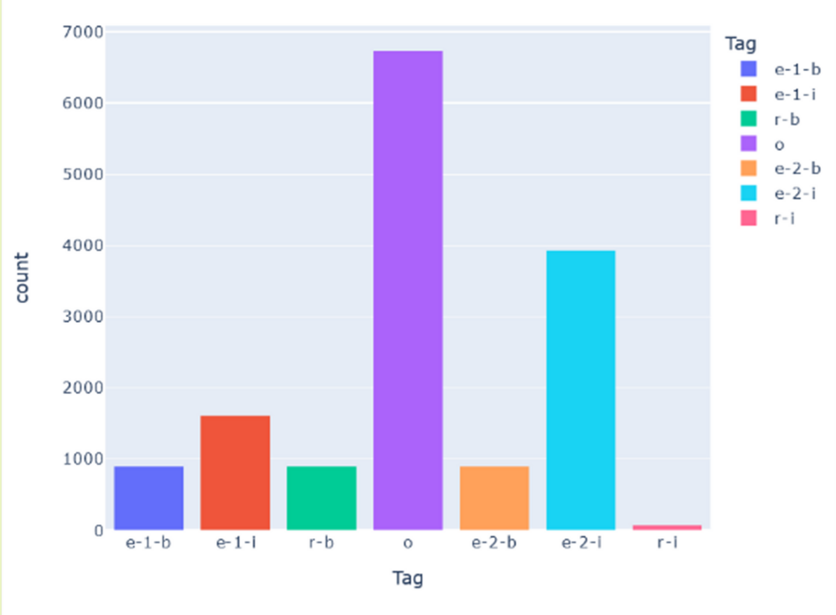
The input is a sentence, and the output is one of the 7 labels corresponding to each word in the sample indicate by the subscripts.



<Relation>
<Entity-1>
<Entity-2>
Professor Sageev is instructor of machine learning
Professor e-1-b Sageev e-1-i is o instructor r-b of r-i machine e-2-b learning e-2-i
Labels

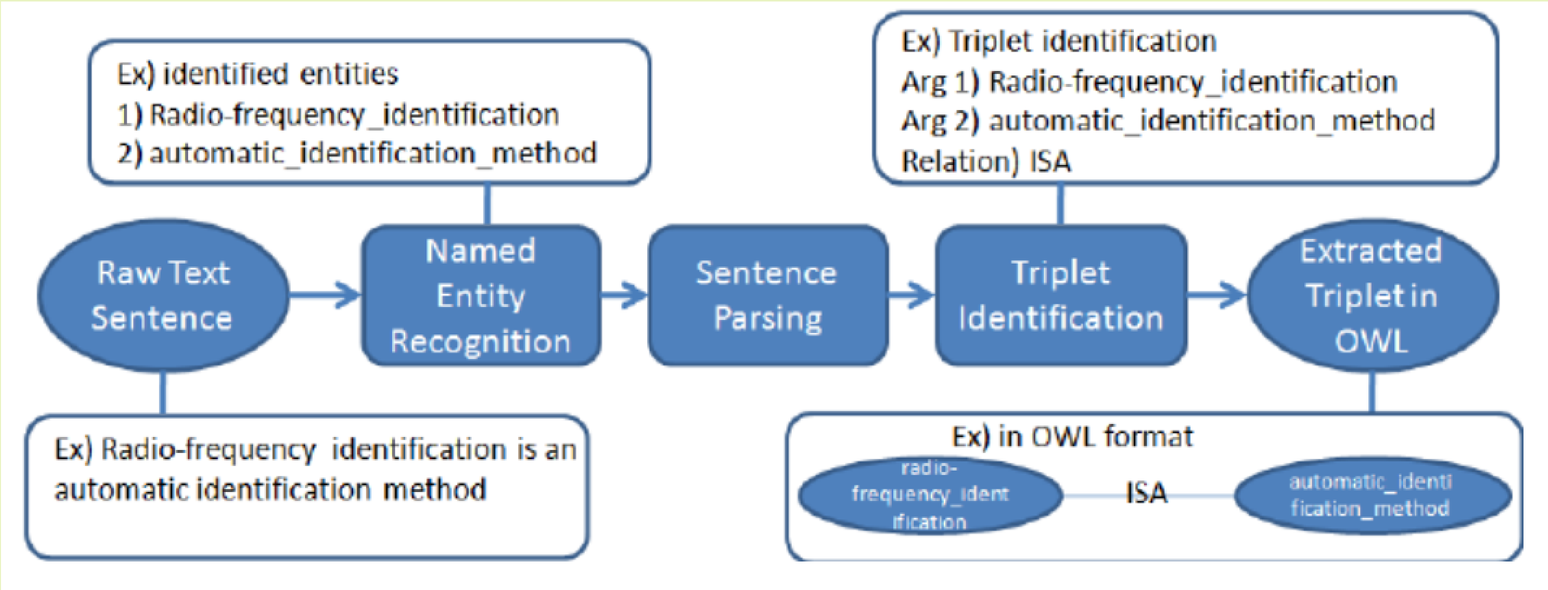






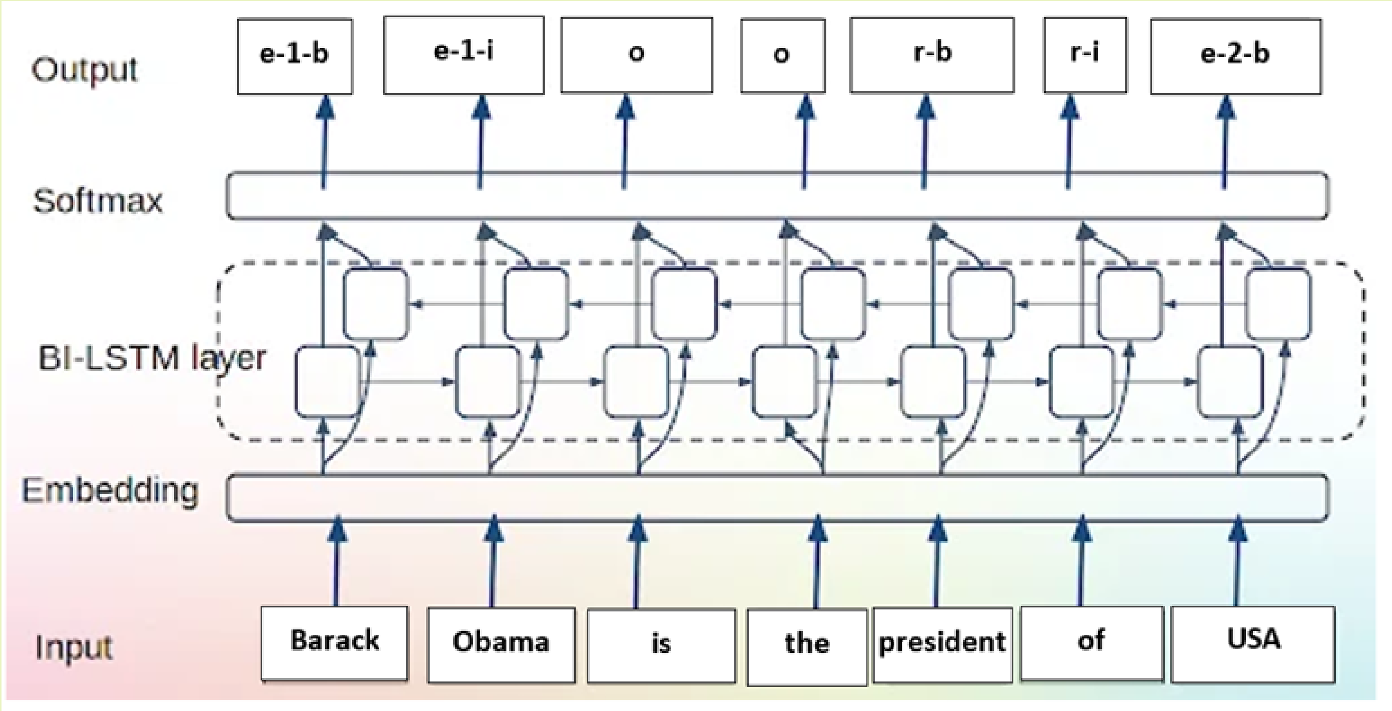
Models

- Baseline Model:
  Instead of using a machine learning model, we use sentence segmentation, dependency parsing, parts of speech tagging, and entity recognition for information extraction.



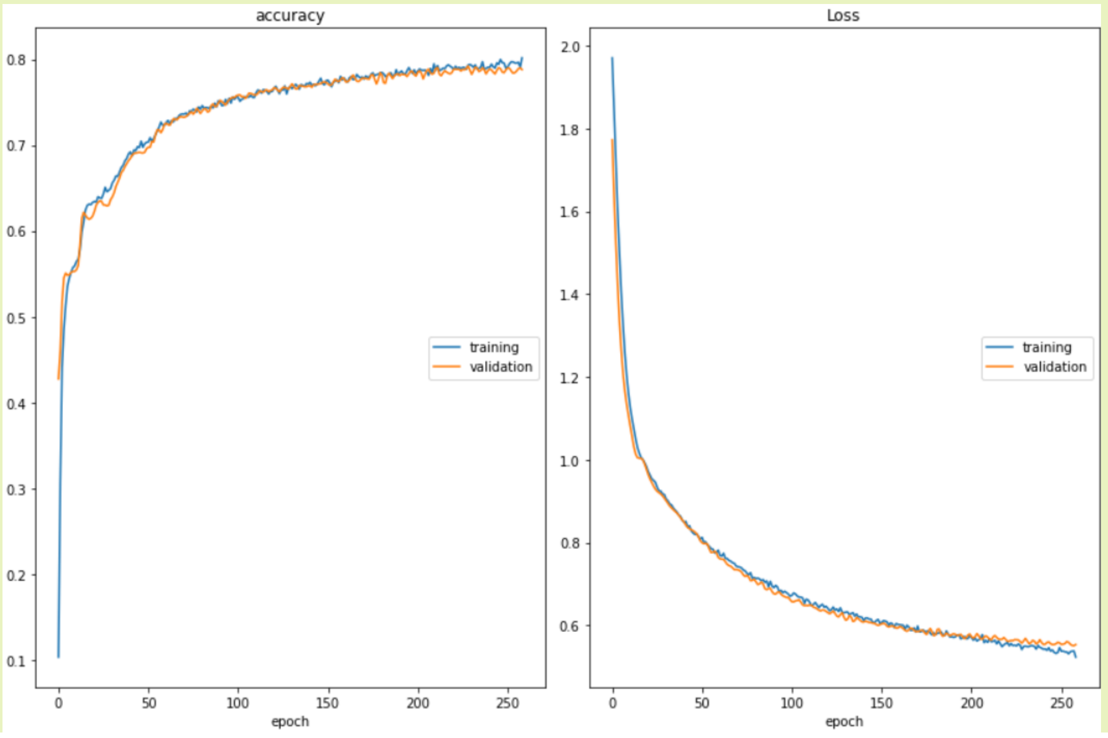
Ex) identified entities
1) Radio-frequency_identification
2) automatic_identification_method
Ex) Triplet identification
Arg 1) Radio-frequency_identification
Arg 2) automatic_identification_method
Relation) ISA
Raw Text Sentence
Named Entity Recognition
Sentence Parsing
Triplet Identification
Extracted Triplet in OWL
Ex) Radio-frequency identification is an automatic identification method
Ex) in OWL format
radio-frequency_identification
ISA
automatic_identification_method

- Recurrent Neural Network:
  - RNN
  - LSTM
  - Bi-LSTM (figure below)



Experiments:

We started by writing a small model and overfitted it to a subset of data to validate if the model would work or do we need to employ another strategy. We trained the models using our data set of 900 samples. We faced challenges such as high bias, huge overfitting, variance and class imbalance in our data. To mitigate these problems, we carefully pre-processed the data such that important information was not lost, yet overfitting was removed. We employed regularisation and dropout layers and ran the model. We employed class weights to prevent the model from overfitting the majority class. Despite all these efforts, we achieved an accuracy of 73% on the test set with minimum overfitting using Bi-LSTM.
Finally, we applied pre-trained embedding, GlOVE, instead of generating our own embedding with LSTM. It showed the least overfitting and gave the best performance.
After having our best final models, LSTM and Bi-LSTM, with balanced class, for all RNN, LSTM and Bi-LSTM and with GLOVE Embedding for LSTM, we created 2 Test Datasets manually, which is Abstract 1 Test Set and Abstract 2 Test set to test out our models in a real-life scenario and concluded the results based on that.



Conclusions

- Bi-LSTM with balanced class weights got the best result of 40% accuracy for custom test set compared to baseline rule-based 60, for Abstract 1 Test set LSTM with balanced class got best results of 20% compared to baseline rule-based 40.
- We added external features (Glove Embedding) to one of the LSTM models. It outperformed all other trained models as well as the baseline with an accuracy of 79% on Abstract 2 and 41 on Abstract 1 Test sets. Which is considerably higher than baseline if we consider the size of our training data which is just 900 samples.
- We discussed that the reason that LSTM with GLOVE [11] and Bi-LSTM stands out among the four architectures is that the prediction task (sequence tagging) which requires the model to learn long references, complex relationships and large vocabulary which can only be achieved by having good embeddings or model architecture which can handle long sequences. LSTM does a good job of extracting features from the sentences.
- With such great accuracy on our very small dataset, it's very promising and can be further extended for Triplet Extraction tasks.

Future Work

- The dataset is small and has high variance. In future, as we collect more data or provide the model with more features, such as POS tags, our models can be further improved upon.
- In this project, we used a synthetic dataset to train our models. To keep the scope of the project considered only simpler sentences. By simpler we mean that the sentences were of limited length, and had only a single triplet, the entity first appeared before entity two with a relation in between. In future, we intend to consider complicated sentences without any of the above assumptions.
- Lastly, the model can only predict the relationships it have seen in the training dataset . Therefore we need to explore techniqueslike one shot learning or zero shot learning to be able to perform well on unseen relationships in natural language text.