Recognizing Textual Entailment

M. A. Muneer Author[1], D. Sairam Author[2], B. Nikhil Author[3], Ch. Shivani Author[4]
[1,2,3,4] Member, Dept of Information Technology JB Institute of Engineering and Technology

Abstract: **Textual Entailment methods recognizes pairs of natural language expressions such that a human who reads (and trusts) the first element of a pair would most likely infer that the other element is also true. This simple abstraction of an exceedingly complex problem has broad appeal partly because it can be conceived also as a component in other NLP applications. If you have a system that is good at recognizing TE, it should be easier to build good systems for Information Retrieval (IR), Question Answering (QA), Paraphrase Recognition (PR), and Information Extraction (IE) and Summarization. Typically, entailment is used as part of a large system.**

Keywords— **Textual Entailment (TE), Natural Language Processing (NLP).**

1.INTRODUCTION

Textual Entailment (TE) in Natural Language Processing (NLP) is a leading link between text parts. The connection holds whether the fact of one content or not. TE is basically focused on conclusion skills rather than focusing on someone’s thinking or portrayal approach. Now-a-days we have talking assistants, for example, Alexa or Google Home which have been applying Natural Language Processing (NLP) to more willingly grasp the significance word and answer the people accordingly in a clever manner. The grade of normal dialect is that there are numerous tactics to express what we have to say. A word can have various significance and we can express our same feelings by using different words. For example, “Obviously, California can and must improve.” And “California can’t do any better.” Are they similar or not? Syntax of both the sentences might be correct but they contradict when we compare them semantically.

Entailment tasks can be done using different approaches:

• Lexical Approach: This involves preprocessing of sentences and matching of word order, sentence length, common words, longest subsequence match and giving out the similarity score on the basis of these features.

• Syntax Based Approach: The syntax trees are used for this purpose, they are used to discover relationships among the different parts of the sentences, like subject verb comparison, object comparisonetc...

• Semantic Approach: This approach takes into account the meaning of the sentences, the context in which a word is used in that particular sentence.

• Hybrid Approach: This approach is a combination of approaches mentioned above.

We have two sentences, one called text (t) and the other hypothesis (h), individually. The motive is whether the one sentence can be concluded from the other or not. TE involves labelling the sentence under category of: entailment, neutral and contradiction.

Example:

Text: Your gift is appreciated by each and every student who benefit from generosity.

Hypothesis: Hundreds od students will benefit from your generosity.

Label: Neutral

The grade of normal dialect is that there are numerous tactics to express what we have to say. A word can have various significance and we can express our same feelings by using different words. It would aid in preserving server space, time and also abstain from ambiguity through improved knowledge of terms. Textual Entailment is intended to emphasize attempts on common verbal concluding skills rather than focusing on someone’s thinking or portrayal approach. The significance of choosing this project is that the solution of this problem can be used for bigger problems such as data extraction, question responding and text overview. Usually, entailment is deployed as a characteristic of a vast model.

2. ARCHITECTURE

Architecture diagram explains the design of the project.
It acts as a Blue Print for theproject.Itgivesa brief ideaof theprojectoverview.

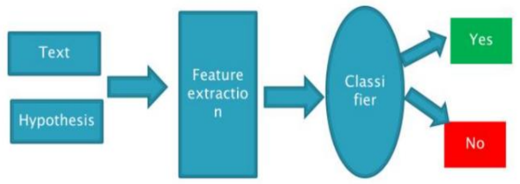

Fig1:Architecture of Recognizing Textual Entailment System

We have worked on constructing a unit which is able to calculate RTE between two sentences. To solve this problem, we are using ML. We have two sentences, one called text T and the other called hypothesis H individually. The motive with is whether one sentence can be concluded from the other one or not. It involves the labelling the sentence under the category of entailment, neutral and contradiction.

The project is divided into fourphases.
Phase 1: Exploratory Data Analysis:
Dataset is analyzed using various methods available in python, information about total number of rows columns, data-type of columns, unique genres and labels and number of rows belonging to each genre, length of sentences in hypothesis are observed.
A validation is performed on the dataset which checks if it has any sentence pairs that are repeated. Total number of unique and repeated sentences are counted using the promptID field that every sentence in premise has and maximum number of times it is repeated. The dataset is checked for null values. The sequence length distribution, average word count are some of the other features that are observed to take appropriate sentence lengths for the task.

Phase 2: Adjusting Sentence Length for taking appropriate sentences:
As sentence in hypothesis and premise are of varying lengths it is important to consider the appropriate length of sentences for our task, the maximum sentence length in premise is of 382 words while in hypothesis, it is of 69 words. GOV.UK [1] shows we need at least 3 words to describe opinion. So, we will apply this filtering criterion to our sentences. As our dataset contains sentences from different genres, it is possible that because of filtering there might be imbalance so under-sampling is done.

Phase 3: Pre-processing of Premise and Hypothesis Sentences for Word Embedding:
This is an important step to perform in the process of cleaning the document as it removes stop words, removes special characters and other important steps which are mentioned below.

1. Convert the given string to lowercase.
2. Removes the special characters such as dollar sign, ampersand, at the rate symbol, brackets etc...
3. Removes punctuation such as full stop, exclamation mark, question mark etc...
4. Performs tokenization i.e., breaking down a string into its corresponding words.
5. Removes words such as articles, prepositions, conjunctions and so on, from the documents which do not have any prophetic use (these words are also known as stop words).

Phase 4: Preparation of Word Embeddings:
This is the word representation layer that involves conversion of words in the dataset to sequences so that they can be processed easily.
A word index list is created that is uploaded on the basis of the premise and hypothesis sentences. The word index of a more frequently occurring word would be smaller in comparison to a less frequently occurring word. The purpose of creating this list to provide an integer-id for the word.
An embedding matrix is prepared where for a word in our word-index list at index I the embedding vector is stored at index i. For embedding Glove vector [2] is used that is trained on Common Crawl corpus on the basis of word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. The GLOVE word embeddings are not modified during training. Our dataset contains a total of 3,92,702 sentence pairs having 77,556 unique words. Our dataset and the GLOVE embeddings have 59,383 words in common and 18,174 words do not match. So for the out-of-vocabulary (OOV) words, the word embeddings are taken of the same length but are initialized randomly.

Phase 5: Working on Lexical Approach to solve the problem:

This methodology is divided into mainly two major parts. One is feature extraction and second is prediction using the XG-Boost model.

Feature Extraction: We have extracted three different kinds of features from the sentences-

- Basic features- involves extraction of features as length of questions, common words in the question etc...
- Advanced Features- involves extraction of features as fuzzy ratio, token set etc... to give an idea of sentence similarity.
- Distance Features- involves calculation of Word Mover's Distance (WMD), Cosine distance etc...

Machine Learning Model Application:

Various combinations of basic features (BF), advanced features (AF), distance features (DF) and weighted word2vector are then fed to the XG-Boost ensemble model which we have chosen as it has a faster processing speed processing speed than Random Forest and the classification is multi-class. Then the model is trained with hyper-parameters like mma_depth, n_estimators, learning rate etc... The results of the models are compared using log loss values and accuracy through confusion matrix. The results were not as expected so we moved to using a semantic approach for solving the problem.

Phase 6: Working on Semantic Approach to solve the problem:

This methodology involves calculation of word order similarity and sentence semantic similarity [3] and then obtaining a similarity score for the sentences in the test set.

Word Order Similarity: Two vectors are formed in relation with word order with respect to each other and the difference between them is calculated.

Sentence Semantic Similarity: Word Sense Disambiguation using lesk- Leskis a dictionary based method used to identify the sense of a word in a particular context that is a word can have different meanings.

Calculation of sentence vectors- For each word in the premise shortest path distance is calculated to each word in hypothesis sentence. After calculation of word vectors the semantic similarity score is calculated usin the algorithm given in by using words having similarity greater than a benchmark similarity of 0.8025 [4].

The final similarity score is calculated using a weighted average of the abovementioned similarity scores and we have considered a similarity score of above 0.5 to be labeled as 'entailment', below 0.4 as 'contradiction' and rest as 'neutral'.

Phase 7: Construction of Different Layers of Deep Learning Model:

Word embeddings are used in this methodology and then the following layers are implemented to calculate the final result.

1. Context Representation Layer- The embedded words are then passed onto this layer that uses Bi-LSTM for extracting contextual information.
2. Matching Layer- This is the most important layer and we will be using two strategies for the same – Max pooling matching and taking a weighted sum for each dimension. This layer is why the model says Multi-perspective[5] matching as we will be matching each time-step of the contextually embedded hypothesis sentences with each time-step of premise sentence and vice-versa.
   The matching is done using cosine similarity measure, using the following formula-
   $m = Fm(v1, v2)$(Equation 4.1)
3. Aggregation Layer- This layer is used to obtain a fixed length matching vector using Bi-LSTM to embed the two sequences obtained from the previous layer and then by aggregating the vectors obtained from the model.
4. Prediction: This layer is used to predict the probability of each sentence pair to belong to each of the three labels. For this a feed forward neural network is used and a softmax function is applied at the end layer.

3. ALGORITHMS

1. XG-Boost:It is an ensemble Machine Learning (ML) model that is is an implementation of gradient boosted decision trees which is faster than other ensemble models. XG-Boost optimizes gradient boosting algorithms through parallel processing, tree-pruning, handling missing values and regularization to avoid bias/ overfitting. We train the XG-Boost ML model

with hyper-parameters like max_depth, n_estimators, learning rate etc...

2. Bi-MPM: This model has five different layers and involves taking into consideration the contextual embeddings and matching the sentences from different perspectives to give a better idea about the context of words.

The XG-Boost model is fed a different combination of features in the feature set, these features include- basic features (BF), advanced features (AF), tf-idf weighted vector and weighted word2vector. The Bi-MPM model employs ma pooling technique for matching and is adjusted for dropout as it uses Bi-LSTM [6] which is prone to over-fitting. The results of the models are compared using log loss values and accuracy.

4. MODULES

A. Pre-processing and Word Representation:

The first module is of preprocessing in which the sentence undergoes through the different operations which focusto lessen the sentence length by eliminating unwanted words, grammatical mistakes, finding the end words etc... and choosing the appropriate words for computing the similarity. These operations are: removing tags, tokenization, removing stop words, converting symbols to words etc… For representation of sentences as vectors we have used various embedding-word level, sentence level and contextual so that they can be interpreted by our Machine Learning Models.

B. Feature Extraction:

Since we have applied different methods for resolving the problem, each approach requires a different set of features for implementation. Different sets of features are considered for lexical and semantic analysis of the sentences.

C. Model Application:

Machine Learning Models such as Bi-LSTM, XG-Boost and Bi-MPM are used for our final prediction and for evaluation of the above models we have used accuracy and log loss values.

5. CONCLUSION

We have tried to focus on both lexical and semantic to resolve this problem. We have not only used the basic features but also fuzzy and distance features to give us better results as feature extraction is the most important part for any ML model. Since it was important to take care of semantic similarity while implementing the model, we have taken into consideration Word Mover's Distance to take care of that. We choose XG-Boost to process our feature set as it has a lower processing time and is better in comparison to other ensemble models, the model accuracy of 54.16% when we used the distance features(6), advanced features(15), basic features(7) and average word vectors of sentences(300 dimensions) (total number of features = 28) and log loss is 0.9459. Then for taking into consideration semantic analysis we computed semantic similarity scores which gave us a poor accuracy, this pushed us to use Bi-LSTM for taking into consideration the context of the words. We applied dropouts to reduce the over fitting and improve the results. Then we implemented Bi-MPM model and performed matching through Max-pooling and mean strategies and it gave the highest accuracy and minimum log loss that is 78.15% and 0.4618 respectively in comparison to the other approaches and also did better than the baseline models that is mentioned [7].

6. FUTURE ENHANCEMENT

Although the accuracy seems promising for this task, we can include Parts of Speech tagging to improve the embeddings which in turn will help the model to perform better.
Syntactic similarity can be taken into account to make entailment decisions. Syntax trees showing relationships among different parts of speech can be used for comparison and calculation of different features.
It can be used as a base for recommendation systems, paraphrase detection and information retrieval systems to give better results that are more similar in meaning to the information typed in by the user.

REFERENCES

[1] Government Digital Service, "Content design: planning, writing and managing content", 25 February 2016. Available: https://www.gov.uk/guidance/content-design/writing-for-gov-uk#how-people-read.

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation, 2014, [Online], Available: https://nlp.stanford.edu /projects/glove/.
nyu.edu/projects/bowman/multinli/.

[3] Pawar, Atish, and Vijay Mago. "Calculating the similarity between words and sentences using a lexical database and corpus statistics." arXiv preprint arXiv:1802.05667 (2018).

[4] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," Communications of the ACM, vol. 8, no.10, pp. 627-633, 1965.

[5] Haldar R, Wu L, Xiong J, Hockenmaier J, "A Multi-perspective Architecture for Semantic Code Search" .arXiv preprint arXiv: 2005.06980. May 6, 2020. [Online] Available: https://https://www.aclweb.org/anthology/S14-2114/.

[6] Wang, Zhiguo, Wael Hamza, and Radu Florian. "Bilateral multi-perspective matching for natural language sentences." arXiv preprint arXiv:1702.0.3814 (2017).

[7] Adina Williams, Nikita Nangia and Samuel R. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference, "arXiv preprint arXiv :1704.05426 (2017), [Online], Available: https://www. nyu. edu/projects/bowman/multinli/paper.pdf.