Automatic Fact Verification

Arushi Sinha
887764
University of Melbourne

Snigdha
947194
University of Melbourne

***Abstract*—The research paper aims at exploring the classical problem of fact verification in NLP. The two aspects taken into consideration for fact classification/labelling, are information retrieval and label classifier. PyLucene is used for information retrieval, text indexing and searching capabilities. The baseline model uses, feature engineering and naive bayes as the text classifier. To improve upon the performance decompositional attention model has been used. The paper discusses evaluation of different results generated by the model over datasets and simulation.**

Keywords—Naive Bayes, Decompositional Attention model.

I. Introduction

Studies related to text classification, fact verification are gaining more importance recently because of the availability of electronic documents from a variety of sources. Text categorisation (also known as text classification or topic spotting) is the task of automatically sorting a set of documents into categories from a predefined set [2].

The goal of natural language processing (NLP) is to process text with computers in order to analyse it, to extract information and eventually to represent the same information differently. We may want to associate categories to parts of the text (e.g. POS tagging or sentiment analysis), structure text differently (e.g. parsing), or convert it to some other form which preserves all or part of the content (e.g. machine translation, summarisation). The level of granularity of this processing can range from individual characters to subword units[3] or words up to whole sentences or even paragraphs.[1]

The use of neural networks for NLP applications is attracting huge interest in the research community and they are systematically applied to all NLP tasks. However, while the use of (deep) neural networks in NLP has shown very good results for many tasks, it seems that they have not yet reached the level to outperform the state-of-the-art by a large margin, as it was observed in computer vision and speech recognition.[1]

Natural Language Processing (NLP), Data Mining, and Machine Learning techniques work together to automatically classify and discover patterns from the electronic documents. The main goal of text mining is to enable users to extract information from textual resources and deals with operations like retrieval, classification (supervised, unsupervised, unsupervised and semi supervised) and summarisation. However how these documented can be properly annotated, presented and classified. So it consists of several challenges, like proper annotation to the documents, appropriate document representation, dimensionality reduction to handle algorithmic issues [1], and an appropriate classifier function to obtain good generalisation and avoid over-fitting. [2]

The aim of the research report analyses the performance of text classification model, over parameters like precision, recall, F1 score. Also, the paper explores information retrieval(IR) techniques like PyLucene. The baseline model uses Naive bayes classifier and to improve upon the performance of baseline model, decompitional attention model has been explored.

Fact Extraction and VERification (FEVER) dataset [4] consists of 185,445 claims manually verified against the introductory sections of Wikipedia pages and classified as SUPPORTED, REFUTED or NOT ENOUGH INFO. For the first two classes, the dataset provides combination of sentences forming the necessary evidence supporting or refuting the claim. Obviously, this dataset is more difficult than existing fact-checking datasets. In order to achieve higher FEVER score, a fact-checking system is required to classify the claim correctly as well as retrieving sentences among more than 5 million Wikipedia pages jointed as correct evidence supporting the judgement.

In the following paper, section II, cover the related work on text classification and the use of neural nets in NLP. Section III- In detail explanation of the information retrieval, the baseline model, decompositional attention mode, learnings and parameters. Section IV- Discusses the results and simulations.

II. Related work

A. Text classification

Many Information Extraction tasks such as Named Entity Recognition or Event Detection require background repositories that provide a classification of entities into the

basic, predominantly used classes location, person, and organisation. Several available knowledge bases offer a very detailed and specific ontology of entities that can be used as a repository. However, due to the mechanisms behind their construction, they are relatively static and of limited use to Information Extraction(IE) approaches that require up-to-date information. In contrast, Wikidata is a community-edited knowledge base that is kept current by its user base, but has a constantly evolving and less rigid ontology structure that doesn't correspond to these basic classes. The author presents the toolNECKAr, which assigns Wikidata entities to the three main classes of named entities, as well as the resulting Wikidata NE dataset that consists of over 8 million classified entities[5]

Text classification models have been heavily utilised for a slew of interesting natural language processing problems. Insufficient and imbalanced datasets will lead to poor performance. In this paper, the use of ConceptNet and Wikidata to improve sexist tweet classification by two methods (1) text augmentation and (2) text generation. In the text generation approach, new tweets are generated by replacing words using data acquired from ConceptNet relations in order to increase the size of our training set, this method is very helpful with frustratingly small datasets, preserves the label and increases diversity. In the text augmentation approach, the number of tweets remains the same but their words are augmented (concatenation) with words extracted from their ConceptNet relations and their description extracted from Wikidata. In our text augmentation approach, the number of tweets in each class remains the same but the range of each tweet increases. The experiments show the approach improves sexist tweet classification significantly in our entire machine learning models. The approach can be readily applied to any other small dataset size like hate speech or abusive language and text classification problem using any machine learning model.[6,7]

B. Neural network

Natural Language Inference (NLI) or Recognising textual entailment (RTE) detects the relationship between the premise-hypothesis pairs as “entailment”, “contradiction” and “not related”. With the renaissance of neural network[3] and attention mechanism[4], the popular framework for the RTE is “matching-aggregation” [5]. Under this framework, words of two sentences are firstly aligned, and then the aligning results with original vectors are aggregated into a new representation vector to make the final decision. The attention mechanism can empower this framework to capture more interactive features between two sentences. Compared to Fever task, RTE provides the sentence to verify against instead of having to retrieve it from knowledge source.

Another relative task is question answering (QA) and machine reading comprehension (MRC), for which approaches have recently been extended to handle large-scale resources such as Wikipedia[6]. Similar to MRC task which needs to identify the answer span in a passage, FEVER task requires to detect the evidence sentences in Wikipedia pages. However, MRC model tends to identify the answer span based on the similarity and reasoning between the question and passage, while similarity-based method is more likely to ignore refuting evidence in pages.

III. TEXT CLASSIFICATION MODEL

Dataset : The dataset consists of wiki-text.zip a collection of wikipedia documents training.json a set of training claims and answers devset.json a set of development claims and answers, test-unlabelled.json a set of test claims to do predictions.

For the text classification model two approaches were adopted.

A. Base model

For the base model we have designed a pipeline architecture starting with text pre processing, and then trying out different supervised machine learning algorithm on the data set, comparing the accuracy of different algorithms.

Text preprocessing

For the text preprocessing we followed the following steps:-

1. Removal of stop words from the claim and evidence text
2. Converting the text of claim and evidence to lower case
3. Removal of special characters and punctuation marks
4. Removal of less frequent words from the evidence. The frequency was set to 5 occurrences.

Feature Engineering:-

1. The text document is converted to a matrix of token counts(CountVectorizer),
2. Then transform the count matrix to a normalized tf-idf representation(tf-idf transformer).

After the feature extraction is done Naive Bayes is used as the baseline model for the text classification. Multinomial variant of the Naive Bayes has been used. This works well for data which can easily be turned into counts, such as word counts in text. To make the vectorizer => transformer => classifier easier to work with, we will use Pipeline class in Scikit-Learn that behaves like a compound classifier.

B. Decompositional attention model.

Document and Sentence Retrieval:
Pylucene was used for indexing the documents and for the document retrieval process.Each sentence in the corpus was

treated as a document. The index stored the Page identifier , sentence number, and the corresponding text . It was observed that Named Entities were missing from sentence text that belonged to a particular page identifiers eg:

986_NBA_Finals 1
Sentence Text : It pitted the Eastern Conference champion Boston Celtics against the Western Conference champion Houston Rockets

To make the query more focused towards the Named Entity , each sentence was preprocessed and Page identifier were appended in the sentence, Incase the identifier term was missing :

986_NBA_Finals 1
Sentence Text : 986 NBA Finals It pitted the Eastern Conference champion Boston Celtics against the Western Conference champion Houston Rockets

We then used SpaCy's document similarity which is a cosine similarity that uses vectors trained from GloVe's word2vector model and selected the top 5 evidence for each claim .

Learning Methodology:

For developing a model that could predict whether the evidence "SUPPORTS", "REFUTES" or is "NOT ENOUGH INFO" for a given claim we tried to implement the Decomposable Attention Model for Natural Language Inference [7] which utilises alignment in machine translation [8] and alignment, attention [9], uses a technique where the phrase is divided into sub-phrase and then the sub phrases that are aligned are compared[7].

Let c = (c1, …..cm) represent Claim of length m and e = (e1...en) represent Evidence of length n which are the concatenated top 5 evidence retrieved in the previous process.

The entire process can be described as follows [7]:

1. Create subphrase alignment by computing the attention weights from a feed forward neural network using ReLU as the activation function for claim and evidence separately. The claim and evidence dot product is normalised .
2. A two layered feed forward network ReLU Network is used to compare each word to its aligned phrase separately , which Indicates how strongly a word associates to a given phrase.
3. These comparisons are aggregated separately such that we get two vectors : One for claim to evidence association and another for evidence to claim association
4. The two vectors are then given to a Dense layer and softmax is applied.

The total complexity of the model is O(ld^2 + l^2d) where l is sentence length and d represents all hidden dimensions[7]

Implementation and Parameters:

The model was developed using Keras and SpaCy .During the data preprocessing we did not remove claims that do not have any evidence as opposed to the implementation [7] . A blank space is considered for claims with no evidence . We had to specify a Max character length of 1000 due to system and memory limitations of the development environment .

To represent words 300 - dimensional GloVe embeddings were used to represent words and the dimensionality was projected down to 200. SpaCy was used for the embeddings and all the out-of-vocabulary words were assigned numbers between 0 and 100.[7] The Feed-forward networks had a dropout of 0.2 .The batch size used for training was 256 and Dataset had a validation split of 0.2. The model was trained on the entire devset but the final trained model used for codelab was trained on first 12000 claims of the training set for 5 epochs (due to system limitations of the development environment)..

IV. Results and Simulations

Model	Dataset	Feature/Training	Validation Accuracy
Base model	Devset	CountVectorization Tf-idf	0.43
Decomposition Attention model	Devset	10 Epochs 256 batch size 0.5 dropout	0.66
	Training Set (max length = 12000)	5 Epochs 256 batch size 0.2 dropout	0.8588

Table1:The table shows the validation accuracy of the models on devset.json and train.json . Where 80% data was used for training and 20% for testing .

Performance evaluation of Baseline Model

The Naive Bayes classifier was run on the Devset, the validation accuracy is 0.43. The baseline model classifier assumption, that features are independent of one another when conditioned upon class labels, is rarely accurate. The dependance of features on one is not taken into consideration which did not work well as compared to the decompositional attention model. It was also observed from the results that the

classification accuracy is better in the model as it considers each classification independent of other, but the class label accuracy roughly averages out for the entire dev-set.

Our Entire Pipeline performance seems to be limited by the Document and Sentence Recall and Precision . A lower Precision shows that a lot of relevant information is not being extracted by the IR system . The tables given bellow (Table2) shows the performance of our overall system . It can be seen that there was a significant improvement in document and sentence selection after using Cosine similarity (SpaCy) and choosing top 5 documents rather than top 2 during IR.

Decompositional Attention Model	Evaluation on Devset with top 5 according to Spacy cosine similarity Score.py	Code Lab with top 2 Documents	Code Lab top 5 according to Spacy cosine similarity
Label Accuracy	43.31%	41..42%	43.29%
Sentence Precision	8.68%	4.37%	10.39%
Sentence Recall	35.92%	7.11%	42.99%
Sentence F1	13.98%	5.41%	16.73%
Document Precision	12.62%	4.95%	16.34%
Document Recall	47.67%	7.70%	57.64%
Document F1	19.96%	6.03%	25.46%

Table2:The table shows the performance of the entire system on first with devset using score.py choosing top 5 documents using SpaCy similarity,, Code Lab submission result when choosing top 2 documents and Code lab submission of choosing top 5 documents using SpaCy similarity

Error Analysis : The pylucene IR framework could match the tokens of claim with evidence to a minimum of 5 evidences . This happened because each token can have multiple meanings . Also different spellings for the same proper noun can be misunderstood by the system.

V Conclusion

The research report analyses the performance of text classification/fact verification model, over parameters like precision, recall, F1 score. The paper explores information retrieval(IR) package like PyLucene and SpaCy for text preprocessing. We experimented with Supervised Machine Learning Algorithm for the baseline model, which used naive bayes classifier and Decompositional Attention Model. Evaluation of the models were done using parameters like precision, F1 score and recall for information retrieval of documents and sentences. While accuracy validation for the model of the claims were used to evaluate the model. The Decompositional Attention Model outperforms Naive Bayes. Advancement in Neural Networks and Deep Learning techniques seems promising in providing solution. Better performance can be achieved with deep learning algorithm with better computational power. Sentence and Document Precision plays a pivotal role in the overall performance of the model.

V Future Work

Information retrieval was crucial for the classifier model performance and improvement. It can be improved further by exploring advanced frameworks, packages or even trying to build a customised IR for the problem. As the claim and evidence were from all categories, general name entity relationship could be applied. But if we could cluster the claims to different categories and develop name entity relation for specifically for them, it would yield better results as the ambiguity of the tokens is removed. Label accuracy can be improved further by training the model over advanced and ambiguous claim verifications.

References

1. Very Deep Convolutional Networks for Text Classification Alexis Conneau, Holger Schwenk, Yann Le Cun.
2. A REVIEW PAPER ON ALGORITHMS USED FOR TEXT CLASSIFICATION Bhumika1 , Prof Sukhjit Singh Sehra2 , Prof Anand Nayyar3
3. Neural Machine Translation of Rare Words with Subword Units. Rico Sennrich and Barry Haddow and Alexandra Birch, School of Informatics, University of Edinburgh
4. The Fact Extraction and VERification (FEVER) Shared Task James Thorne1 , Andreas Vlachos , Oana Cocarascu , Christos Christodoulopoulos3 , and Arpit Mitta
5. NECKAr: A Named Entity Classifier for Wikidata Johanna Geiß and Andreas Spitz and Michael Gertz Institute of Computer Science, Heidelberg University Im Neuenheimer Feld 205, 69120 Heidelberg
6. Xu et al., 2015; Luong et al., 2015; Bahdanau et al., 2014
7. Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.
8. Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.
9. Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems* (pp. 577-585).
10. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
11. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
12. K. Elissa, “Title of paper if known,” unpublished.
13. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
14. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
15. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
16. Zhou, Peng, et al. "Attention-based bidirectional long short-term memory networks for relation classification." *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vol. 2. 2016
17. Zhao, S., Cheng, B., & Yang, H. (2018, November). An End-to-End Multi-task Learning Model for Fact Checking. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)* (pp. 138-144).
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*(pp. 5998-6008).
19. Apache Lucene https://lucene.apache.org/pylucene/features.html
20. Keras python deep learning Library https://keras.io/
21. Industrial Strength Natural Language Processing https://spacy.io/