Rumour Stance Classification: NLP Final Project Report

Aditya Adhikary
2015007
IIIT-Delhi

Sanidhya Singal
2015085
IIIT-Delhi

Saurabh Kapur
2015087
IIIT-Delhi

Abstract

The task of determining veracity and authenticity of social media content has been of recent interest to the field of NLP. False claims and rumours affect peoples perceptions of events and their behaviour, sometimes in harmful ways. Here, the task is to classify tweets in threads of tweets based on the stance possessed by the tweet, which can be of 4 categories: supporting (S), denying (D), querying (Q), or commenting (C), i.e., SDQC (1).

Keywords: NLP, Text Classification, Social Media, Rumour, Stance, Veracity, Pheme.

1 Introduction

1.1 Definitions

1. **Memes**: According to Wikipedia, a meme is “an element of a culture or system of behaviour passed from one individual to another by imitation or other non-genetic means.” In the current context, these are thematic motifs that spread through social media.

2. **Phemes**: In 2014, this new definition was made to describe “memes which are enhanced with truthfulness information”. The PHEME project was focused on this.

3. **Rumour**: “A circulating story of questionable veracity, which is apparently credible but hard to verify, and produces sufficient skepticism and/or anxiety”.

1.2 Motivation and Problem Statement

There are three stages of Rumour Analysis:

1. **Rumour Detection**: Identify pieces of information that need to be verified, and classify as Rumour or Non-rumour.

2. **Rumour Stance Classification**: Classify public stance towards the rumour.

3. **Veracity Classification**: Determine if a rumour is true, false or remains unverified.

Rumour detection is a well studied problem, while the other two are not. In our project, we focus on the second one.

In Rumour Stance Classification, we classify the stance possessed by the tweet into 4 categories: supporting (S), denying (D), querying (Q), or commenting (C), i.e., SDQC. There are two major uses of this task:

1. It shows journalists the unfolding online debates on a given pheme.

2. It further helps in stage 3 of rumour analysis, i.e., veracity classification.

Fig. 1.1 shows an example rumourous thread of Twitter conversations with the rumour stance indicated alongside each tweet.

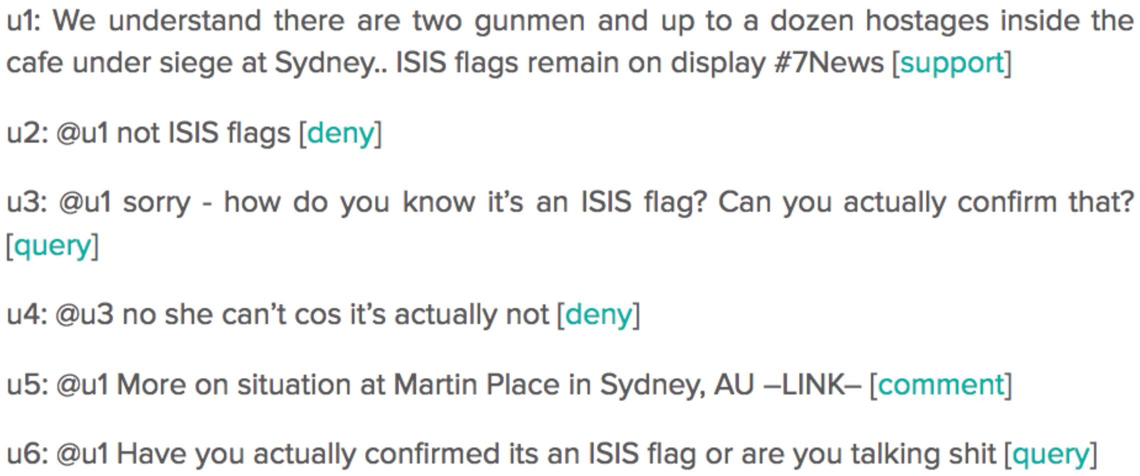
2 Dataset

Our training dataset comprises 297 rumourous threads collected for 8 events in total, which include 297 source and 4,222 reply tweets, amounting to 4,519 tweets in total. These events include well-known breaking news such as: The Charlie Hebdo shooting in Paris; The Ferguson unrest in the U.S.; The Germanwings plane crash in the French Alps; etc. **Gold Standard Testing dataset** has 28 rumorous threads, with 1,049 tweets in total. You can find the public dataset here (3).

3 State of the art

This is a difficult classification task because each tweet is not sufficiently descriptive on its own,

Figure 1: Example Thread



u1: We understand there are two gunmen and up to a dozen hostages inside the cafe under siege at Sydney.. ISIS flags remain on display #7News [support]

u2: @u1 not ISIS flags [deny]

u3: @u1 sorry - how do you know it's an ISIS flag? Can you actually confirm that? [query]

u4: @u3 no she can't cos it's actually not [deny]

u5: @u1 More on situation at Martin Place in Sydney, AU –LINK– [comment]

u6: @u1 Have you actually confirmed its an ISIS flag or are you talking shit [query]

but needs to be viewed in the context of an aggregate discussion consisting of tweets preceding or extending it in the thread. RumourEval by team Turing at SemEval-2017 Task 8 is the SOTA model with an accuracy of 78.4% (2). They used a LSTM-based sequential model for the challenge of Rumour Stance Classification in the Subtask A.

4 Progress And Evaluation Metrics

4.1 Progress

1. Data retrieved, cleaned, preprocessed and stored as pickle file.

2. Created several different models for feature extraction.

   (a) n-gram Models with frequencies of words and tf-idf score as features. (Here, n = 1, 2, 3)

   (b) Word2Vec Model

   We provide the details of these models in Section 5.

3. Trained and Tested on different classifiers.

   (a) Multinomial Naive Bayes Classifier (NB): A fundamental generative baseline classifier for text classification problems.

   (b) Support Vector Machines (SVM): sklearn's SGD Classifier (defaults to Linear SVM) with squared hinge loss for max 1000 iterations. SVM supports high-dimensional vectors such as that of text by finding the separating hyperplanes for the classes.

   (c) Logistic Regression (LR): sklearn's Logistic Regression with LBFGS solver. LR is used commonly as it is discriminative (estimates $P(y|x)$ directly) and often gives better results than NB.

   (d) Random Forest (RF): sklearn's Random Forest classifier with default settings. A random forest fits a number of decision trees on various chunks of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The chunk size is always the same as the original input sample size but the samples are drawn with replacement.

   (e) XGBoost Classifier (XGB): A recent popular classifier (we used the freely available python library API here) which depends on an implementation of gradient boosting decision tree algorithm, which produces a prediction model in the form of an ensemble of weak prediction models, by learning them iteratively.

   (f) Neural Network (NN): Fully connected, dense NN with 5 layers (512, 512, 128, 32, 4 nodes with respective layers). ReLU activation function for first 4 layers and Sigmoid for the last layer. Hyperparameters: Epochs - 40, Batch size - 128 and Validation split - 0.1.

4.2 Metrics

We evaluate the outputs of the classifiers using Accuracy, Precision, Recall and F1-score.

5 Feature Extraction

5.1 N gram model

5.1.1 Preprocessing

For each sentence, the words were tokenized using Regexp Tokenizer, stray alphabets removed, numbers removed, stemmed using Snowball Stemmer, default and some specific stopwords removed (like ”http”, ”rt”, ”co” which are common to many tweets). The nltk python package was used.

We analyse 3 types of n-gram models, namely, Unigram (Bag of Words), Bigram and Trigram models. For each of these models, we look at 2 types of features:

1. The counts of words in the corpus, unnormalised.
2. The tf-idf scores of all the words in the corpus.

5.2 Word2Vec model

5.2.1 Preprocessing

1. Used NLTK word tokenizer to tokenize the text.
2. Removed stop words from the text.

5.2.2 Features

Trained the Word2Vec model with GoogleNews-vectors-negative300 dataset and generated the feature vector for the tweets using this model. Following are the set of features:

1. Count of the number of negative words like not, no, never, etc.
2. Count of the number of Punctuations.
3. Presence of Question Mark.
4. Presence Exclamation Mark.
5. Presence of Hashtag.
6. If there are any User mentions.
7. Presence of URL.
8. Presence of Media (any photo or video).
9. Result of Sentiment Analysis (1 for positive, 0 otherwise).
10. Presence of any swear words.

6 Results and Observations

We trained several classifiers on the two models (n-gram and Word2Vec). This section discusses the results obtained based on the metrics described in Section 4.2.

1. Table 1 shows the accuracies obtained on n-gram models. N-gram models are of two types – A: Bag of Words Model and B: tf-idf Model. Clearly, the tf-idf model (B) shows a better performance as compared to the corresponding simple count-based n-gram model (A), as it is better able to capture the overall context of words over the corpus of tweets (over a particular thread of discussion) due to the idf factor.

2. For the baseline MultinomialNB, due to the low number of training samples in the Deny and Query category, this model is unable to predict any samples of these classes, and P/R is 0. It predicts these samples as Comment, because it does not take into account negations for deny, or questioning words for queries.

3. Table 2 shows the accuracies obtained on Word2Vec model. Logistic Regression obtains the best accuracy of 77.6%.

4. Table 3 shows Precision, Recall, F1-score for SVM, LR and NN classifiers on Word2Vec model. Again Logistic Regression has the best values among all three.

5. The confusion matrix for LR classifier with Word2Vec model is shown in Table 4.

6. As shown in Table 3, Neural Network had higher Precision-Recall values for Deny class as compared to any other classifier.

Table 1: Classification Accuracies (in %) for n-gram Models

	NB	SVM	LR
Unigram Model A	70.54	63.96	71.97
Unigram Model B	73.87	72.16	73.40
Bigram Model A	72.35	73.02	73.87
Bigram Model B	74.07	68.73	73.59
Trigram Model A	73.97	73.97	74.26
Trigram Model B	74.26	73.59	74.16

Table 2: Classification Accuracies (in %) for Word2Vec Model

	NB	SVM	LR	RF	XGB	NN
Word2Vec	72.25	75.40	**77.59**	71.30	72.73	74.16

Table 3: Precision, Recall & F1-Score for Word2Vec Model

	SVM			LR			NN		
	P	R	F1	P	R	F1	P	R	F1
Support	0.77	0.26	0.38	0.71	0.26	0.37	0.43	0.24	0.31
Query	0.44	0.39	0.41	0.59	0.45	0.51	0.55	0.34	0.42
Deny	0.11	0.01	0.03	0.00	0.00	0.00	0.20	0.06	0.09
Comment	0.79	0.93	0.86	0.80	0.95	0.87	0.79	0.92	0.85
Avg/Total	0.71	0.75	0.71	0.71	0.78	0.73	0.69	0.74	0.70

Table 4: Logistic Regression Word2Vec Model Confusion Matrix

	S	Q	D	C
S	24	3	0	67
Q	1	48	0	57
D	1	3	0	67
C	8	27	1	742

7 Additional Datasets

To solve the problem of low representation of deny and query samples in the existing dataset, we tried to append it with newly collected tweets. We crawled tweets related to two more incidents - The recent California shootings (Nov 9, 2018) and the Las Vegas shooting of 2017. We collected 15000 tweets of both, but after deduplication we could get a total 17377 tweets, of which we could label a total of 60 tweets as deny/ query. When added to the training dataset however, the results (precision and recall of deny/query class) were not reinforced to a great extent with any particular model. Hence possibly, additional labeling is required.

8 Conclusion

By improving feature extraction and incorporating tweet dependent (and also other textual) features such as hashtags, link content etc., we were able to achieve accuracies close to that of the state of the art on most of our models. Our highest reported accuracy is 77.6%, which is comparable to that of the state of the art model (78.4%). We also tried to improve the recall on Deny and Query classes by augmenting the training dataset. We also tried to do the same using ensemble methods, and bagging/boosting techniques.

References

[1] Derczynski et. Al, *SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours.*

[2] Kochkina et. Al, *Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM.*

[3] Dataset Link, *SemEval-2017 Task 8 Dataset.*

[4] Bahuleyan et. Al, *UWaterloo at SemEval-2017 Task 8: Detecting Stance towards Rumours with Topic Independent Features.*