Fake News Stance Detection

Xiaowei Wu Sizhu Cheng Zixian Chai

Abstract

Social network and online news media are gaining popularity in recent years. Meanwhile, online fake news are becoming widespread. As a result, automating fake news detection is essential to maintain robust online media and social network. In this work, machine learning methods are employed to detect the stance of newspaper headlines on their bodies, which can serve as an important indication of content authenticity. If the newspaper headline is defined to be "unrelated" to their bodies, it indicates a high probability of the news to be "fake". Specifically, multiple methods are used to extract features relevant to stance detection from a collection of headlines and news article bodies with different stances. These features are then used to train multiple machine learning models including support vector machines, multinomial Naive Bayes, Softmax, and multilayer perceptron. We have demonstrated very high accuracy to detect relevance between the headlines and bodies. This work can be used as a important building block for fake news detection.

1. Introduction

Fake news is deliberate misinformation fabricated with intention of deception, misleading, grabbing attention or even financial and political gain. While human fact checker can identify fake news with high accuracy, the sheer volume of online news renders manual fact checking impractical. Recent development of machine learning provides a possible solution to automate this process. However, accurately and repeatedly identifying fake news is still proven difficult due to the complex nature of human language. With the popularity of online media and detrimental effect of fake news on many aspects of our society, developing a reliable machine learning model for fake news identification becomes very important. Stance is one of the most important indications of news authenticity. In this project, we have studied the stance of the news article headlines on their body texts by predicting the relevance between the headlines and body texts, and if relevant, further identifying the opinion of the title on its body. This is an important part of fake news identification process.

2. Related Work

2.1. SVM - ngram

Sobhani, et al. [9] compared different classification models and feature sets on a stance detection task to categorize opinions of tweets towards given subjects (SemEval-2016 Task 6). SVM with n-gram features yielded the best performance. One limitation is that the SVM model yielded a 10% lower score on the "against" stance compared with the "favor" stance, and indicated slight imbalance in classification.

2.2. SVM - multiple features

Sobhani, et al. [9] extended the SVM-ngram model by adding two types of features: word-embedding, and sentiment features. Sentiment features are extracted from the NRC Word-Emotion Association Lexicon [6][7]which labels each word with either positive or negative sentiments. Compared to the SVM model with single n-gram features, this model better detected positive and negative attitudes. Sobhani, et al. (2016) suggested that the model could be further improved by extraction of entity-relationship features [9].

2.3. BoW MLP

Davis and Proctor [1] developed the model BoW MLP (Bag of Word Multilayer Perceptron) for fake news detection. The BoW MLP model first converts the corpus into bag-of-word vectors, and then uses two softmax layers (one for relevance and one for attitudes) and an entropy-based cost function to make classification.

2.4. Convolutional Neural Network

Yang Shao [11] developed a convolutional neural network system for semantic textual similarity detection. Semantic vector is generated by max pooling over all word vectors and similarity score between two sentences is calculated using the semantic vectors of these sentences.

2.5. Multiple Domain Knowledge Features

Krejzl and Steinberger [4] focused on using domain knowledge related features in SemEval-2016 Task 6, a stance detection task on tweets. Part-of-speech tags, general inquirer, and entities-centered sentiment dictionaries were applied to extract syntactic and semantic features. Also, they defined a domain stance dictionary which lists the most frequent words in each

stance. Their model yielded high accuracy scores but did not exceed the SVM n-gram model [9].

3. Dataset and Features

3.1. Dataset

The stance of a headline is compared to its news body from a data set provided by Fake News Challenge (FNC-1) [2]. Each data instance consists of headline, body and stance. Each stance is one of the {unrelated, discuss, agree, disagree}. An example instance is shown in the appendix. 40350, 9622, and 25413 instances are randomly selected as the training, dev, and test sets, respectively. Features which are believed to differentiate the stances of the corpus are first extracted out. Multiple learning models are used to predict the stance given a headline/body pair.

3.2. Feature Extraction

- **Baseline Features**

For each pair of headline and body, the baseline provided by FNC-1 contains four types of features: the number of overlapped words, the number of overlapped n-grams, the number of negative words in the headline, and polarity of headline and body. Polarity features are modified and described below.

- **Similarity Features**

Cosine similarity is used here as a feature to indicate the similarity between the body and headline. Specifically, the headline and body are converted to sparse vectors H and B, respectively, to indicate the frequency of each word. The cosine of the angle between these two vectors H and B in the high dimensional space is calculated as an indication of similarity.

$$cosine\_similarity = \cos(\theta) = \frac{H \cdot B}{||H||_2 ||B||_2} = \frac{\sum_{i=1}^{n} H_i B_i}{\sqrt{\sum_{i=1}^{n} H_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

According to the above definition, cosine similarity only measures the angles of the vectors instead of magnitude and thus length difference doesn't contribute to the similarity number. If similarity is high, the number should be close to 1, while numbers close to 0 indicate unrelatedness.

- **BOW (Bag-of-word) Vectors**

The BOW model represents each document as a bag of words. Assume a corpus has a set of unique words $\{w_1, w_2, w_3...w_n\}$ and each word w_i is an n-dimensional vector which has value 1 on the i^{th} position and 0 on other positions. Assume all words in each document are exchangeable. A document is defined as an N-dimensional vector $\{w_1:c_1, w_2:c_2...w_n:c_n\}$ in the vocabulary space with each word given a specific weight value c. The BOW vector for the headline-body pair is defined by overlapped words with binary weights. The python package "gensim" is used here to construct "word-id" mappings[3].

- **Word Sentiment**

The idea of word sentiment refers to the sentiment of article specifically. A "refuting words", containing 15 words including {false, deny, refute, doubt,…} is provided by FNC-1. We use a database called WordNet [8], which is embedded inside the python nltk package [5] to find out all the synonyms of the words inside a "refuting words" list. The stance of the newspaper article is predicted to "refute" the associated contents if any of the "refuting word" inside that list is found on the headline.

- **Polarity**

The polarity of the headline and body is determined by counting the number of negative sentiment words. Here, the NRC Word-Emotion Association Lexicon [7] is used to obtain a list of negative emotion associated English words. The identity of a negative word is determined by whether the word is found on the negative word list or not. We assign the polarity of a corpus based on whether it has odd or even number of negative words.

4. Methods

Four different types of classification models, including support vector machines (SVM) (linear and nonlinear), softmax, multinomial Naive Bayes, and multilayer perceptron classifier (MLP) are leveraged for this task. A combination of these models are also tested in order to further improve the accuracy of prediction. Using scikit learn [10], these models are implemented to learn from the training data using k-fold (k=10) cross-validation, and then predict using the test sets.

Support vector machines are learning algorithms which convert features to points in high dimensional space and divide points from different categories by a

gap as wide as possible. Specifically, SVM solves the following optimization problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i (w^T \phi(x_i) + b) \geq 1 - \zeta_i$$
$$\zeta_i \geq 0, i = 1, ..., n \quad (2)$$

With kernels, SVM can also perform nonlinear classification.

As a multinomial generalization of the logistic regression, softmax is a classical method for classification when there are more than two categories. It is intuitive to implement softmax in this problem, when considering the relatedness and the stances of the news as outputs. With a certain parameter θ, the probability of the output classified to class *i* is listed as the following:

$$p(y = i|x; \theta) = \phi_i = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}} \quad (3)$$

Multinomial naive Bayes defines a generative process for the data set and assumes that for features, $p(x_1|y=c)$, $p(x_2|y=c)$, ..., $p(x_n|y=c)$ are independent given a specific category label $y=c$. Therefore, the joint probability of all features conditioned on $y=c$ is the product of each feature conditioned on $y=c$.

$$p(x_1, x_2, ..., x_n|y = c) = \prod_{i=1}^{n} p(x_i|y = c) \quad (4)$$

Then the maximum likelihood and predictions of new data could be calculated through Bayes theorem. Due to these properties, naive Bayes classifier is often used in text classification problems in which the order of words does not matter.

Consisting of an input layer, an output layer, multiple hidden layers each with multiple neurons, neural network is a very power tool for text stance classification as it relies less on accuracy of feature extraction and can work on some crude features. As one of the neural network models, multi-layer perceptron algorithm takes all the features as the input, return classification as output and use backpropagation for training. In this project, ReLU function g(z) = max(z, 0) is used as activation for each neuron of the neural network

5. Experiments, Results and Discussion

The FNC-1 competition metric is used to evaluate model performance. The metric is a weighted accuracy score, with 25% weight on correctly classifying "related" stances, which includes "agree", "disagree" and "discuss", and "unrelated" stances, and 75% weight on correctly classifying three "related" stances.

5.1. Effect of Feature Extraction on Performance

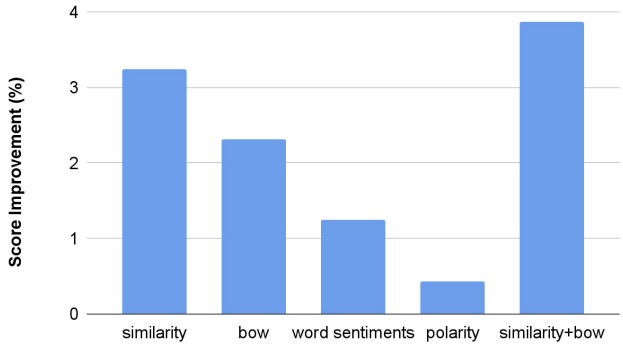

Figure 1: Performance improvement after adding following features: similarity, bag-of-word (BOW), and sentiment features

The performance improvement percentage from different features is shown in Figure 1. It is found that the "similarity" and "bow" features better describe the stances of the headlines towards the bodies, than the "word sentiments" and "polarity features". The addition of both "similarity" and "bow" features improve the performance best than adding a single type of features.

5.2. Model Performance on Each Category

A summary of all models are shown in **Table 1**. Overall, all models have above 90% accuracy on prediction of unrelated stance and below 5% accuracy on disagree stance. Moreover, all models have around 80% accuracy rate on the discussed stance. This discrepancy could be explained by a) the difference in the number of test instances, b) feature extraction, and c) model parameters. For 25413 test instances, 18349 has stance unrelated and 7064 related. Related instances contains 1903 agree, 697 disagree, and 4464 discussed. Data with disagree stance are significantly less than data with unrelated stance. Lack of training data might contribute to the low accuracy rate of disagree stance, and agree stance. Also, extracted features such as overlapped words or cosine similarity focus more on relevance between each pair of headline and body rather than positive and negative attitudes. Finally, model parameters, such as the numbers of

layers and nodes in the MLP model, are yet to be optimized to improve the accuracy of classification.

The accuracy rate for all models in test set is shown in **Table 2**. Across different models, MLP Classifier has the overall best performance. Softmax and linear SVM have better performance on unrelated stance compared with non-linear models, such as SVM with RBF kernel or MLP classifier. Among related stances (agree, disagree, and discussed), MLP works best compared with other models. Multinomial naive Bayes has comparatively better performance on agree stance and disagree stance, and MLP Classifier has the best performance on discussed stance.

Table 1. Test set labels output by multilayer perceptron (MLP), softmax(SF), multinomial naive Bayes (MNB) and support vector machine (SVM)

A\P	Agree	Disagree	Discuss	Unrelated
Agree	**147** (MLP) **116** (SF) **378** (MNB) **97** (SVM)	0 4 39 0	1528 1446 1246 1513	228 337 240 293
Disagree	34 27 62 12	**0** **0** **11** **0**	444 381 436 420	219 289 188 265
Discuss	198 122 625 86	0 0 38 0	**3761** **3556** **3168** **3691**	505 786 633 687
Unrelated	0 7 70 6	0 0 0 0	304 168 1096 202	**18045** **18174** **17183** **18141**

Table 2. Accuracy rate of each stance for all models. Here, the percentage for each category is the percentage accuracy for the test set. The total scores percentage is calculated using score.py (as described above). (bold values: the highest accuracy rate of each stance; related = Agree+Disagree+Discussed)

Accuracy Rate	Linear SVM	Softmax	Multinomial Naive Bayes	MLP Classifier
Unrelated	**99%**	**99%**	94%	98%
Related	82%	80%	85%	**87%**
Agree	5%	6%	**20%**	8%
Disagree	0%	0%	**2%**	0%
Discussed	83%	80%	71%	**84%**
Total Score	75.89%	74.76%	72.48%	**77.74%**

5.3. Performance of model combination

Single models alone have below 20% accuracy on classification of “agree” stances and “disagree” stances and different models show advantages and disadvantages on classification of different stances. We proposed the reason behind the bad performance of the model to be the disproportionate number of the “agree”, “disagree” news instances versus the “unrelated” instances. Therefore, the algorithms tend to predict more “test” newspaper headlines to be “unrelated” to its bodies. To handle this problem, we decided to use a sub-category classification here. The idea is to use a method to classify “related” from “unrelated” first, and use other methods to do further classifications.

Two types of model combinations are proposed to specify the classification process in more details. The two-model combination splits the stance detection task into two classification subtasks and each subtask is completed by a classification model. The first subtask is to classify all headline-body pairs into unrelated and related stances. Related stances include “agree”, “disagree”, and “discuss” stances. For headline-body pairs with related stances, the second subtask further classifies “agree”, “disagree”, and “discuss” stances. The three-model combination splits the task into three subtasks, each completed by a classification model. The first model classifies “related” and “unrelated” stances. For “related” stances, the second model classifies whether a stance is neutral(“discuss” stance) or not. For non-neural stances, the third model classifies whether a stance is “agree” or “disagree”. Models in the combination could use different feature sets. For example, in the 3-model combination, the BOW feature is helpful for the first two classification subtasks. When the BOW feature is applied to the classification of “agree” and “disagree” stances,

overfitting is observed and damages the overall performance.

Overall, two-model combinations and three-model combinations achieved above 75% score on the test set. In the three-model combination, when the third task is completed by multinomial naive bayes, classification of "agree" and "disagree" stances will be significantly improved and the same for the overall performance. The overall performance for SVM+MLP+SVM combination has achieved the highest score of 78.46%

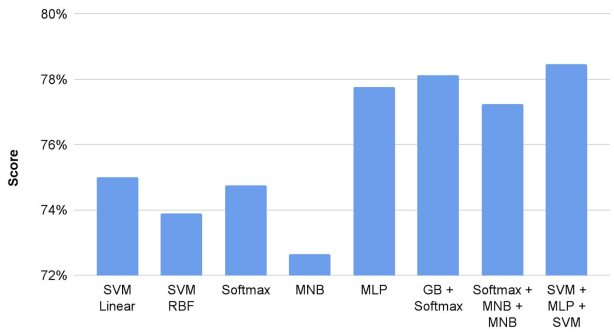

Figure 2: Comparison of test set score generated from different models GB = Gradient Boosting Classifier. LR = Logistic Regression (when softmax is applied to binary classification). MNB = Multinomial Naive Bayes. M1+M2(+M3) = 2(3)-model combination.

6. Conclusion and Future Work

The current project investigates performance of different classification models on a stance detection task proposed by FNC-1. The MLP model yields the best score among all classification models. Compared with single model, splitting the stance detection task into two or three subtasks and utilizing combination of models improved the overall performance. Among different features, combination of cosine similarity and bow features significantly improved the performance.

6.1. Prediction Improvement on Distinguishing Agree *vs.* Disagree Categories

Currently, over 70% of the headline-body pairs in the dataset are "unrelated" and only below 10% are "agree" or "disagree". More "agree" and "disagree" headline-body pairs could be added to the dataset to improve the training process.

6.2. Feature Extraction based on domain knowledge

The current project did not include domain knowledge related features, such as entity-relationships. Future studies could extract name entities from each pair of news headline and news body, and analyze their relationships through a knowledge base.

Appendices

I. Example Data Instance

Body ID: 313

Headline: ADVISORY-Islamist rebel leader killed in U.S. strike - Somali government

Body:

MOGADISHU, Sept 5 (Reuters) - The alert and story on Sept. 5 headlined "Islamist rebel leader killed in U.S. strike-Somali government" is withdrawn and no substitute story will be issued. The story was sourced to a website purporting to be the Somali prime minister's Facebook page. A Somali government spokesman said the page was not official and said the government had not yet commented on whether Ahmed Godane, the head of the Somali Islamist militant group al Shabaab, had been killed in a U.S. strike on Monday. STORY_NUMBER: L5N0R62Q3 STORY_DATE: 05/09/2014 STORY_TIME: 1324 GMT (Writing by Edmund Blair; Editing by Sonya Hepinstall)

Stance: Disagree

Contributions

Xiaowei Wu: Similarity Features, Support Vector Machines, Neural Network

Zixian Chai: BOW features, Multinomial Naive Bayes, Two/Three-model Combination

Sizhu Cheng: Word Sentiment features, Polarity features, Softmax

Parts of feature extraction and models in this report are used by Xiaowei Wu also in his CS221 final report.

References and Bibliography

[1] Davis, Richard, and Chris Proctor. "Fake News, Real Consequences: Recruiting Neural Networks for the Fight Against Fake News."

[2] Fake News Detection Challenge: FNC-1. <http://www.fakenewschallenge.org>

[3] Gensim 3.1.0: Python framework for fast Vector Space Modelling. <https://radimrehurek.com/gensim/>

[4] Krejzl, Peter, and Josef Steinberger. "UWB at SemEval-2016 Task 6: Stance Detection." *SemEval@ NAACL-HLT*. 2016.

[5] NLTK 3.2.5: Natural Language Processing Toolkit.

<https://pypi.python.org/pypi/nltk>

[6] Mohammad, Saif, et al. "SemEval-2016 Task 6: Detecting Stance in Tweets." *SemEval@ NAACL-HLT*. 2016.

[7] Mohammad, Saif M., and Peter D. Turney. "Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon." *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, 2010.

[8] Princeton University "About WordNet." WordNet. Princeton University. 2010. <http://wordnet.princeton.edu>

[9] Sobhani, Parinaz, Saif Mohammad, and Svetlana Kiritchenko. "Detecting stance in tweets and analyzing its interaction with sentiment." *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*. 2016.

[10] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

[11] Yang Shao. “HCTI at SemEval-2017 Task 1: Use Convolutional Neural Network to Evaluate Semantic Textual Similarity” *Proceedings of SemEval-2017*, pages 130-133