Preliminary proposal: Classifying Subjectivity Using News Sources
Team members: Xiaorui Tang, Lingwei Cheng
Changes are highlighted (Revision)
1/22/2016

Introduction

The goal of our project is to distinguish subjective and factual remarks in news articles. We will approximate subjective remarks by using articles in the opinion sections, assuming that articles from the opinion sections are subjective while the articles in the other categories are objective. This is a very broad assumption which if time permits we will address and propose improvement.

We will scrape articles from http://www.factcheck.org/askfactcheck/ as our source dataset for all things considered truth. For opinions, we will use articles from http://www.usatoday.com/opinion/. We will build our corpus and assign labels to help us build the classifiers.

We will write our code in Python and use libraries including nltk, scikit-learn, bs4, and matplotlib. After building the model, we will construct a web interface using Django. If we find it helpful to analyze the network structure of in-document links, we will use snap to conduct network analysis.

Deliverables

In addition to a final report and presentation as required, we will deliver a simple web application interface where the user will be able to submit some texts and the application will return how probable the text is opinionated and highlights the words that signifies opinions.

Plan of Action

1. Literature review: we will conduct a brief literature review on what has been done in the past in terms of classifying subjective opinions from objectivitity. We will use these approaches to enlight our solution.

2. Scrape the data: we will scrape ~200 articles from Factcheck.org and USA News website if permitted. In the meantime, we will also contact the university library to request permissions for bulk download articles from the news article databases.

3. Machine Learning: This will be an iterative process based on the evaluation results.
    a. Preprocessing: Extract main part of the article from the raw html structure, and tokenize raw string into words. We will use stop words and Parts-Of-Speech (POS) tags to filter out unwanted texts. We will discard texts in quotes from the fact articles to be more cautious and ensure that we only have statements in our truth corpus.
    b. Feature generation: Select a vectorizer from the sklearn library to extract features from the text. Some features we plan to generate based on our review of existing literature include:
        i. TF-IDF: The frequency of the word in a given text.

ii. Parts of speech tag: Many researches have shown that a large number of adjectives indicate a high probability of the text being subjective. We will therefore use POS tags to inform our model. Opinion indicator seed word: This include both positive and negative adjective words such as “good”, “awesome”, “bad” as well as verbs such as “think”, “believe”. The existing literature we reviewed currently uses candidate sets of such words. We will either build our own candidate sets by finding the most common words occur in our opinionated texts or use the ones from existing research.

iii. Tense: We will genearte a feature based on the number of words in past tense as it is indicative of statements.

iv. Negation: According to existing literature, the presence of negation is another good indicator of opinon. We will set the feature value of the negation of a word to 1 if there is a negation. We realize that this is a naive approach as double negation will become a confirmation. We plan to tweak this feature later.

v. Modifier: Modifiers such as adjectives and adverbs are indicative of opinion. The feature value of the word will be 1 if it is a modifier。

vi. Besides using unigram, we might also break the texts into bi-grams or tri-grams for additional cues for our model.

c. Feature selection: If the feature space is too large, we will reduce the number of features using functions in sklearn.feature_selection and only feed the important features to the model. We will also include in the final report the most predictive features in our model.

d. Algorithm: Build the model using different combinations of algorithms, parameters, and thresholds, and select the ones with better performances.

e. Evaluation: Run the models against the testing sets and output statistics including accuracy, precision, recall, f1, auc_roc, training time, testing time, etc. We will also compare the accuracy to the baseline of the whole dataset.

4. Construct the interface: After building the machine learning model, we will construct a user interface on top of it. After receiving a text input from the user, the interface will call the model, return the result, and present it via the interface.

5. Optional Further Steps: Opinion pieces may reference both news and opinions, but news articles may not be as likely to reference opinions. If there is time left after finishing the steps above, we will try to analyze the relationships among these news articles using the in-document links. Apart from visualizing the network, we will also try to extract some features from the relationships and add them to our models.

Midterm Presentation

By the time of midterm presentation, we will finish literature review, data collection, preprocessing, and feature generation. We will also run the first pass of the model and output the evaluation results, and develop a basic version of user interface. If time permits, we will also implement the selection of features, algorithm, parameters and thresholds into our pipeline. We will present the evaluation results for our first pass and the beta version of the user interface.

Final Presentation

We will try to finish steps 1-4 in our action plan above. We will present both the evaluation results of the models selected by us and the web interface which allows the user to test the models interactively. If we do the network analysis part, we will also present the visualization of the document network.

Evaluation

We will evaluate our work by how well the solution performs on test data as well as by comparing it to results from existing literatures. We will perform a standard 5-fold cross-validation on our training/testing datasets. We will use evaluation metrics including accuracy (baseline), precision, recall, f1, auc_roc score, training time, testing time, precision-recall curve, and ROC curve.

Work Division

Lingwei: obtain data source, preprocessing, building the web interface

Xiaorui: feature generation, pipeline setup, feature selection

Both: literacture review; machine learning model iteration; constructing the crosss-validation process; producing F1 scores, accuracy scores, and precision-recall graphs.

Literatures

http://cs229.stanford.edu/proj2012/Busch-DistinguishingOpinionFromNews.pdf
http://arxiv.org/ftp/arxiv/papers/1312/1312.6962.pdf
http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf
http://www.cs.columbia.edu/nlp/papers/2003/yu_hatzivassiloglou_03.pdf
https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf