COMP9417 Assignment 2:
Fake News Challenge Stage 1 (FNC-I):
Stance Detection

Andy (studentID) Edrian (studentID) MJ (studentId)

NOTE: might also implement other (basic) classifiers such as a multi-class logistic regression classifier, so it doesn't look like our approach is a pure copy of *Solat in the Swen*'s approach.

1. Introduction

1.1. The problem. Fake news is news created to deliberately misinform or deceive readers. Detecting fake news is currently a major issue for social media companies such as Facebook, who are investing vast amounts of resources to find ways to detect and prevent its circulation. The Fake News Challenge Stage 1 (FNC-1) held from late 2016 to mid-2017, was a competition based on tackling one aspect of fake news detection – stance detection. Stance detection involves comparing the perspectives (or stances) of two pieces of text relative to a topic, claim or issue (reference FNC website). This involves comparing a headline and body of text and determining one of four relationships between them: agree, disagree, unrelated, neutral. This would prove useful in identifying what different news organizations are saying about a topic. This challenge seeks to explore ways in which we can automate this process.

1.2. The approach. We engineered features to capture the different aspects of the relationship between the headline and body of an article. The features we engineered all aim to address one or more of the target labels i.e. agree, disagree, unrelated, neutral. We then train an XGBoost classifier on some training data published by the FNC-I website(reference) and validate our model on some test data also provided by the website(reference).

1.3. Important aspects. Elaborate on any important aspects of our approach here.

2. Implementation

2.1. Data pre-processing. Our implementation begins by pre-processing the headline-article pairs in the training set. This stage consists of four stages: tokenizing, stemming, stop-word removal, and n-gram conversion.

2.1.1. Tokenizing. Describe tokenizing phase here.

2.1.2. Stemming. Describe stemming phase here.

2.1.3. Stop-word removal. Describe stop-word removal here.

2.1.4. N-gram conversion. Describe n-gram conversion here.

2.2. Feature engineering. The most important part of our task is to engineer features which reflect some aspect of the relationship between the headline and article body. The features we chose are mostly based on the competition winner's approach, but we have also incorporated

a couple of our own features. These features include: n-gram count similarity, TF-IDF cosine similarity, SVD cosine similarity, word2vec, sentiment analysis, and subject-verb-object (SVO) similarity. The process of engineering each feature are explained below.

2.2.1. N-gram count similarity. Explain it here.

2.2.2. TF-IDF cosine similarity. Explain it here.

2.2.3. SVD cosine similarity. Explain it here.

2.2.4. word2vec. Explain it here.

2.2.5. Sentiment analysis. Explain it here.

2.2.6. Subject-verb-object (SVO) similarity. Explain it here.

2.3. Model implementation. We implement a XGBoost model that takes in the features listed above and generates an optimal(?) classifier.

2.4. Test data. Each new test instance must be converted the set of features listed above before classification. This process is explained below:

2.4.1. N-gram count similarity. Explain it here.

2.4.2. TF-IDF cosine similarity. Explain it here.

2.4.3. SVD cosine similarity. Explain it here.

2.4.4. word2vec. Explain it here.

2.4.5. Sentiment analysis. Explain it here.

2.4.6. Subject-verb-object (SVO) similarity. Explain it here.

3. Experimentation

3.1. Subset feature selection

3.2. Cross validation(?)

References
Appendix