Project Design：

1/ Preprocessing

We plan to use some preprocessing methods on the given dataset like removing stopwords and doing lemmatization before vectorizing documents. Besides we’ll also try to find more popular preprocessing approaches in NLP to discover more information from the data.

2/ Model Selection

We decide to try different combinations of vectorizesrs and classifiers, including Tfidf, Word2Vec, Doc2Vec and Logistic Regression, Random Forest, XGBoost and SVM respectively. We’ll also use GridSearch methods on these combinations to obtain the best set of hyperparameters. After building our pipeline we’ll apply f1 metrics to evaluate the performance and choose the best combination to present.

3/Testing the effect of models

Only using the dataset, fake\_or\_real\_news.csv, may lead to overfitting, which means that the well-trained model may not behave as well as expected in other situations. Thus, extra dataset are needed to be sure that the final model can generalize efficiently. We have already found other similar dataset, such as BuzzFeedNews (https://github.com/BuzzFeedNews/2016-10-facebookfact-check/tree/master/data) and LIAR(https://www.cs.ucsb.edu/william/data/liar dataset.zip)

4/ Other points deserve to be considered

1. People always regard titles as part of the corpus and train them together. However, as we all know, title is the most informative message part in an article, and lots of fake news are clickbait. We wonder if we can train with the title alone and combine the results with those trained from the corpus, in this way we may get a better result.
2. We referred to the solution from the winner of a fake news challenge competition “<http://www.fakenewschallenge.org/>” and learned how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem. Their team found that the best results came from combining multiple models in an ensemble. The team’s final submission was an ensemble based on an 50/50 weighted average between gradient-boosted decision trees and a deep convolutional neural network. We would like to try reproducing this results based on their theory from “<https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html>”
3. The first model used by the team applies several different neural networks used in deep learning. This model applies a one-dimensional convolutional neural net (CNN) on the headline and body text, represented at the word level using the Google News pretrained vectors. CNNs allow for efficient, effective parallel computation while performing The output of this CNN is then sent to an multi-layer perceptron (MLP) with 4-class output -- “agree,” “disagree,” “discuss,” and “unrelated” -- and trained end-to-end. The model was regularized using dropout (p=.5) in all convolutional layers. All hyperparameters of this model were set to sensible defaults, however, they were not further evaluated to find better choices.
4. The other model employed in the ensemble is a Gradient-Boosted Decision Trees (GBDT) model. This model inputs few text-based features derived from the headline and body of an article, which are then fed into Gradient Boosted Trees to predict the relation between the headline and the body.

Labor Division:

Mengze Zhang: Preprocess the raw data, use TfIdf as the vectorizer and experiment these vectors on all traditional machine learning classifiers discussed above to choose some most fitted classifiers and build a new classifier that can combine the results from those chosen ones. Test the theory in 4.1, train the title and text respectively.

Haorui Ji: Preprocess the raw data, use Word2Vec as the vectorizer and train it with with all traditional machine learning classifiers discussed above to find the best result. Test the theory in 4.2.(b), extract features using multiple methods, combine them altogether and fed into classifiers.

Weihao Tan: Preprocess the raw data, use Doc2Vec as the vectorizer and train it with with all traditional machine learning classifiers discussed above to find the best result. Test the theory in 4.2.(a), try to build a deep learning model and see if there’s improvement compared with traditional ones.

All group members will all search other academic material and take this as a good opportunity to learn more about NLP and data engineering.

Schedule Flow Chart:  
