Summary

Binkai Tan

Group 1

Group Ideas

In the final project, we have tried 4 pipelines to explore possibilities, while I am responsible for the first two pipelines. Instead of trying just one method, we compared several combinations and classifiers with a base-line model.

As is shown in the following flow chart:

1. First of all, we tried the base-line method, in which we used doc2vec plus XGBoost.
2. Then considering Inter-sentence relationship and context we used Elmo to transfer doc into vectors.
3. Thirdly, we tried Glove with several kinds of classifiers and compare the results.
4. Finally, we tried BERT, which is really powerful.

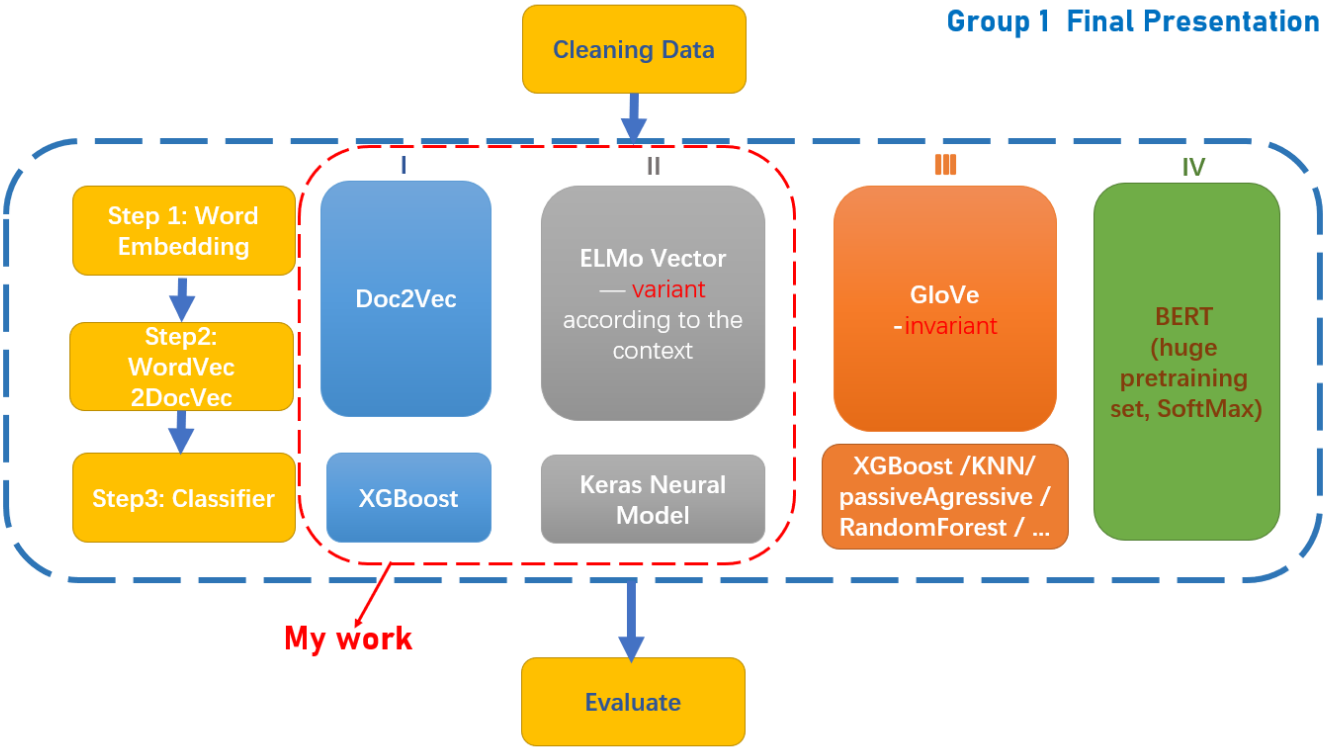

Group 1 Final Presentation
Cleaning Data
Step 1: Word Embedding
Step2: WordVec 2DocVec
Step3: Classifier
I
Doc2Vec
XGBoost
II
ELMo Vector — variant according to the context
Keras Neural Model
III
GloVe -invariant
XGBoost /KNN/ passiveAgressive / RandomForest / ...
IV
BERT (huge pretraining set, SoftMax)
My work
Evaluate

Fig.1 Flow Chart

Personal Contribution

In the final project, we have tried 4 pipelines to explore possibilities, while I am responsible for the first two pipelines.

1. Pipeline 1

In the first pipeline, I tried a base-line method, where Doc2Vec and XGBoost were used.

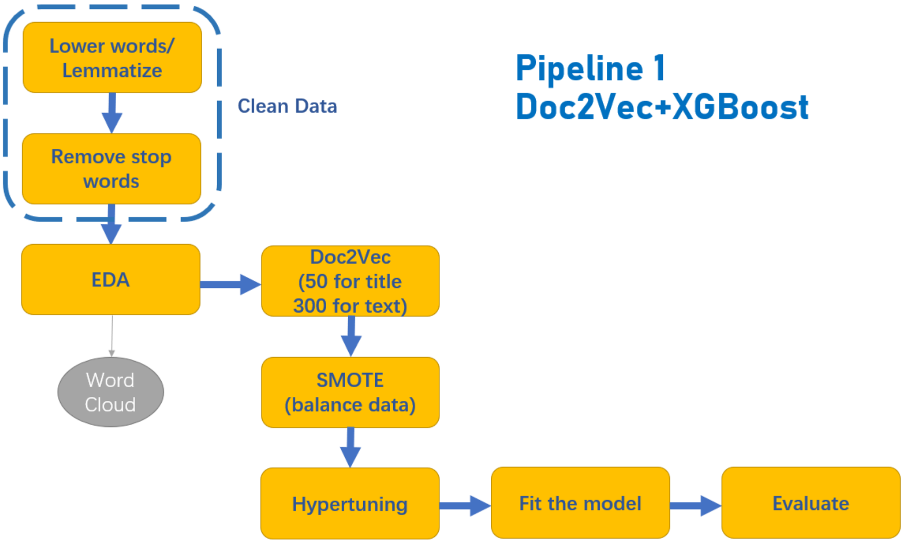
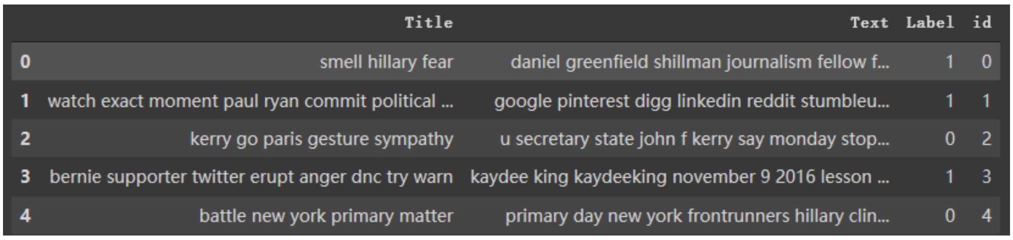


Fig.2 Pipeline 1 Doc2Vec+XGBoost

Firstly, I cleaned the data, which contains getting lower-case, lemmatizing words, and removing stop words. After processed, the title and text is like this:



	Title	Text	Label	id
0	smell hillary fear	daniel greenfield shillman journalism fellow f...	1	0
1	watch exact moment paul ryan commit political ...	google pinterest digg linkedin reddit stumbleu...	1	1
2	kerry go paris gesture sympathy	u secretary state john f kerry say monday stop...	0	2
3	bernie supporter twitter erupt anger dnc try warn	kaydee king kaydeeking november 9 2016 lesson ...	1	3
4	battle new york primary matter	primary day new york frontrunners hillary clin...	0	4

Fig.3 Title and text after processed

And then EDA, count words frequency and got the top 15 in both fake news and real news. It goes along with word clouds, also both for fake news and real news.

Moreover, Doc2Vec, imported from genism.model, I set the dimension of text as 300 and title as 50 (tried some other combinations and this was the best). Then combine them as the feature.

Smote, it was used to balance data, and the number of samples in each label would become the same both in training data and test data. Then split it into training set and test set.

Finally XGBoost, I used grid search and cross validation to hypertune the model. Then fit the model with X_train and y_train, predicted the y_test and then drew a confusion matrix, calculated the precision, recall and f1 score.

The result of f1 score here is 0.877.

2. Pipeline 2

In the previous model (Doc2Vec), we did not take the Inter-sentence relationship and context into account. So In the second pipeline, I used ELMo as a new way to convert the text into vectors.

ELMo is a deep contextualized word representation that models both (1) complex characteristics of word use (e.g., syntax and semantics), and (2) how these uses vary across linguistic contexts (i.e., to model polysemy). These word vectors are learned functions of the internal states of a deep bidirectional language model (biLM), which is pre-trained on a large text corpus. They can be easily added to existing models and significantly improve the state of the art across a broad range of challenging NLP problems, including question answering, textual entailment and sentiment analysis. It can be well adapted to the following environments: one is the complex features of word usage in semantics and grammar, the other one is that word usages change when language environment changes.

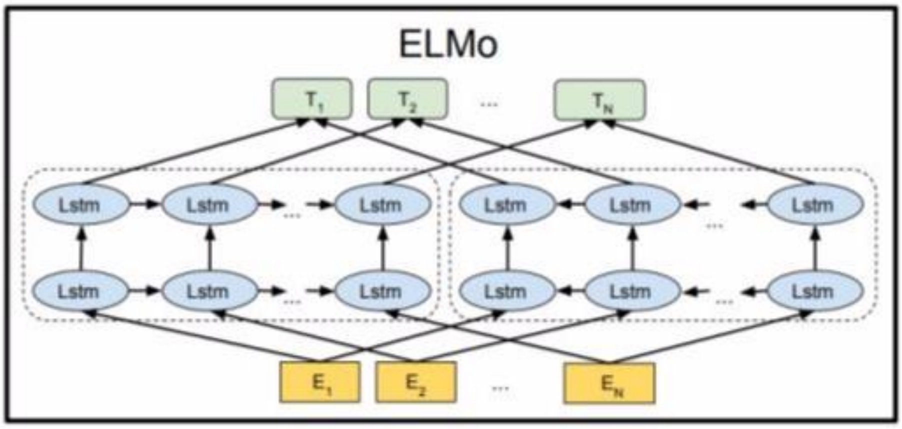


Fig.4 ELMo

In the implementation stage, I did the following work:

1. Map Textual labels to numeric using Label Encoder,
2. Convert Sentence to Elmo Vectors,
3. Divide dataset to test and train dataset,
4. Train Keras neural model with ELMo Embeddings.

Unfortunately, running all of the samples will overflow the Colab's memory, so I tried 100 samples to display it symbolically. The f1 score here is 0.85.

Summary

I really appreciate Mike, Amit and our TA's effort. In this six-week program they were always there to answer our questions and solve problems for us.

In the program, I have learned a lot about text processing, data exploring, word embedding, different classifiers (Logistic Regression, XGBoost, KNN, Neural Networks) with hypertuning, evaluating and so on. I enriched my knowledge of text mining and machine learning in this six week. More importantly, I practiced much in the Google Colab and also debugged a lot, which was really significant to get improved. In the final project, I built a group with Zhao, Wang and Liu, in which we exchanged learning experience and worked in cooperation.

Overall, I'm honored to participate in this program with all of us, really fascinating!

Reference

[1] https://github.com/huixugec/FakeNewsTutorials

[2] https://radimrehurek.com/gensim/models/doc2vec.html

[3] https://allennlp.org/elmo

[4] https://blog.csdn.net/triplemeng/article/details/82380202