Stance Detection and Stance Classification

Data Science Capstone at FiscalNote

Xiaodan Chen, Xiaochi Li

What’s Stance ?

“The health bill is too expensive for people, I totally support the healthcare reform act ! ”

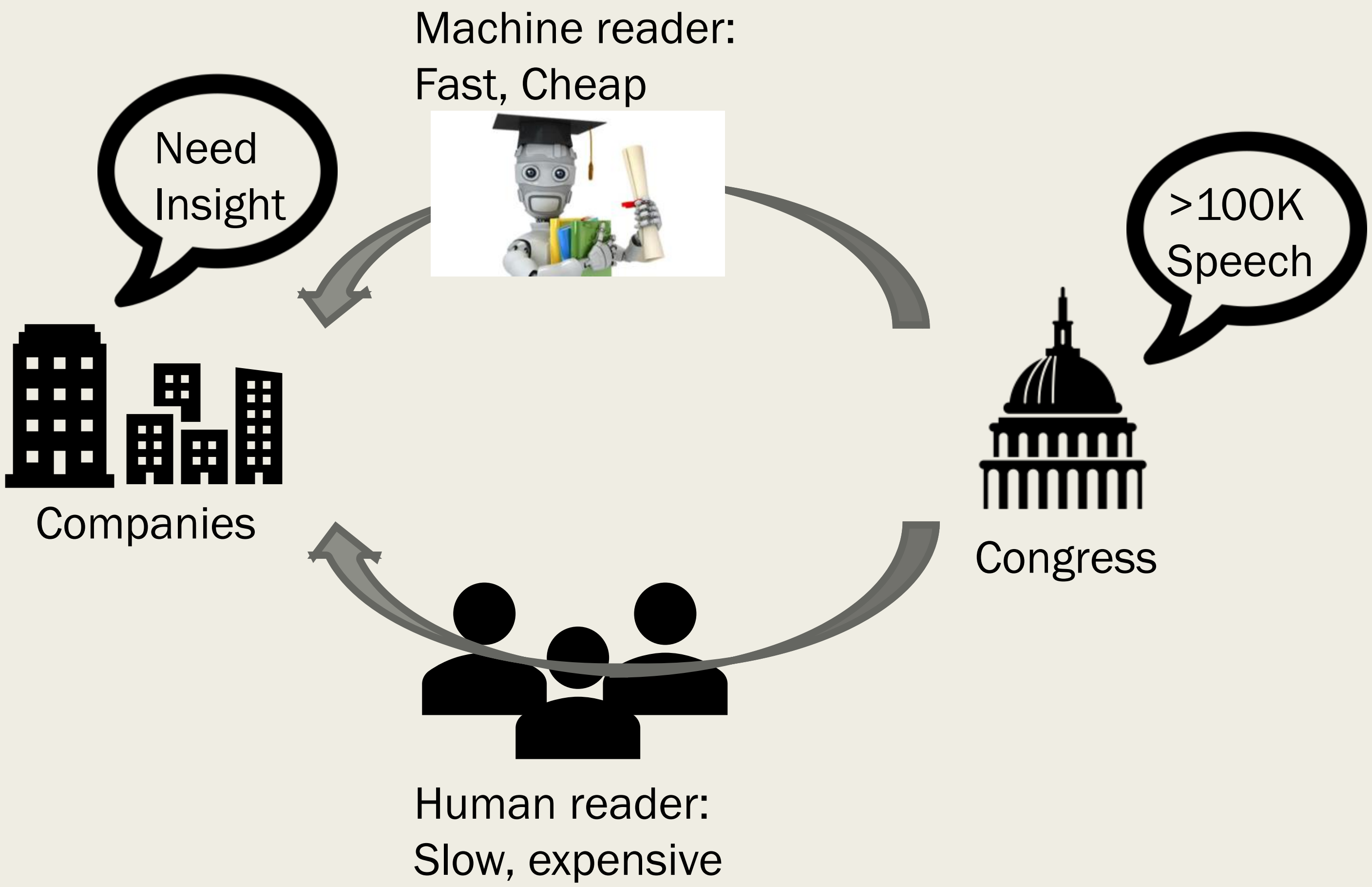
Have attitudes on something → Exist stance

“For a long time, there has been a water shortage problem in Africa”

No attitudes → No stance

only fact statement

Why we care?



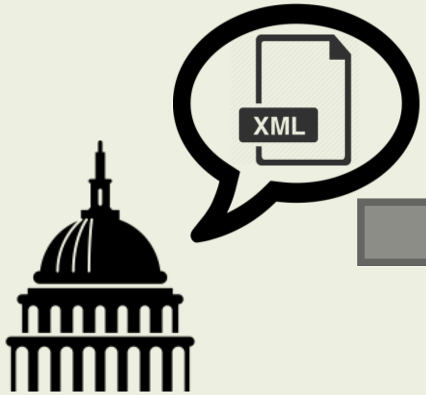
What's "Stance"?
Stance (political term) ≈ Attitude

- **Confliction:**
1. Companies need insights from congress speeches to make right decisions.
2. Tons of congress speeches are unstructured

- **Solution:**
Build an machine "reader" to get insights from the speech

Problem Statement

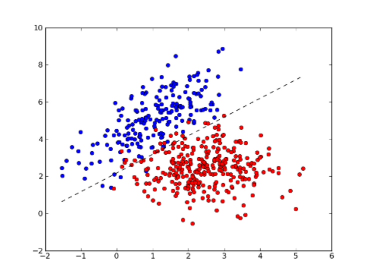
Phase 1 : Load Data



Load

```
<speaker person="VISCLOSKY" personId=
  <p style="I11"><person-ref name="VI
  <p style="I11">The spirit of <perso
  <p style="I11">This year, the Gary
  <p style="I11">Though very differer
  <p style="I11">Mr. Speaker, I urge
</speaker>
```

Phase 2 : Stance Detection



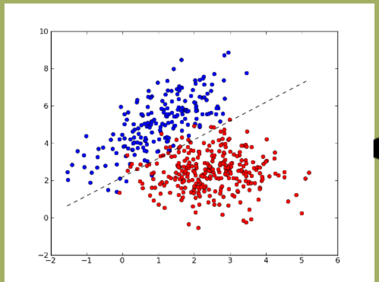
Is there a stance?

Yes

No

No stance:
Mr. MICA . Mr. Speaker, I rise today to congratulate our ally and friend, the Republic of slovakia, on her 20th anniversary of independence. In two brief decades, ...

Phase 3 : Stance Classification



Is the stance support/opposition?

Support

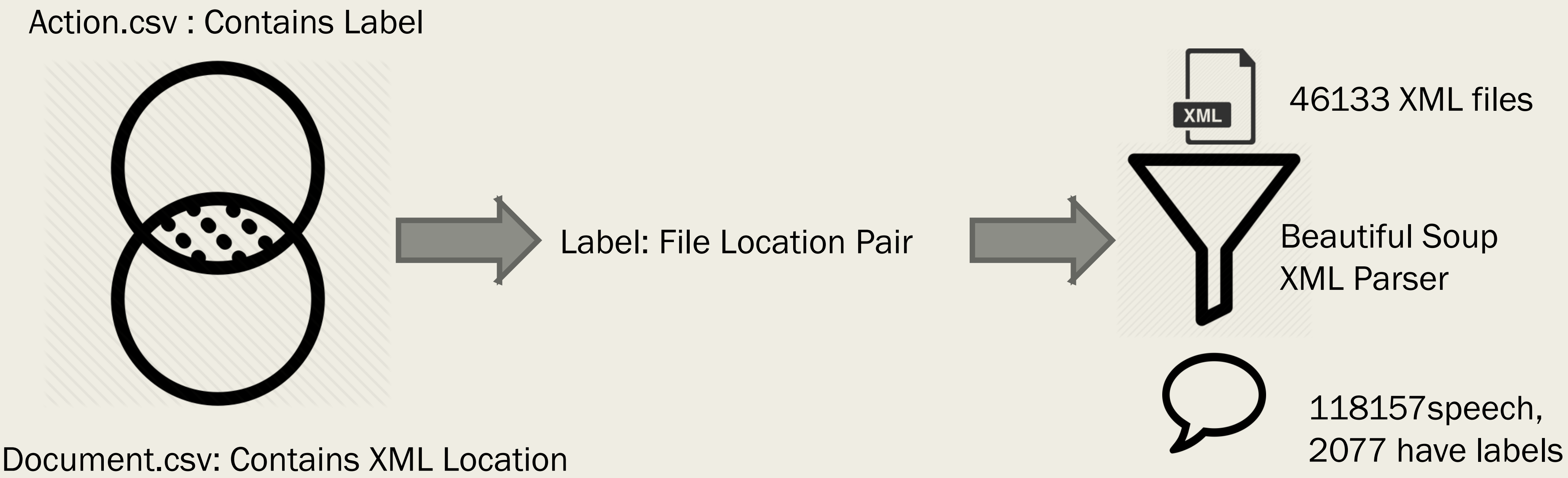
Opposition

'Ms. GABBARD . Mr. Speaker, I rise today in strong **support** of the “Helping Heroes Fly Act.” ...

‘Mrs. BEATTY . Mr. Chair , I rise in strong **opposition** to the devastating funding cuts to the Transportation and Housing initiatives in this appropriations bill, ...

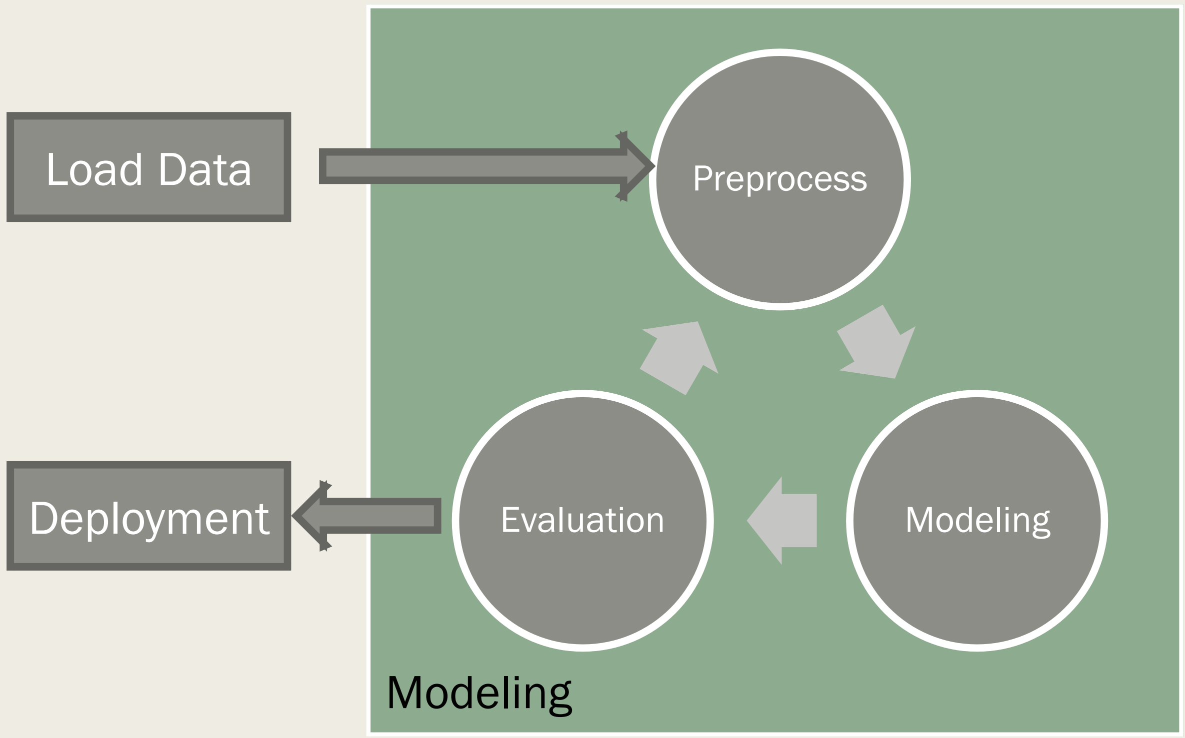
Dataset

- **Provided by:** FiscalNote
- **Labeled data:** 2 Sessions, 118,157speech, 2077 have labels (support or opposition)
- **Unlabeled data:** Approximately 20 Congressional Sessions, 10 times of labeled data.



Project Framework

- Load Data
- Two stage model
  - *Preprocess*
  - *Modeling*
  - *Evaluation*
- Deployment



Stance Detection

(Xiaodan Chen)

Label more data with Regular Expression

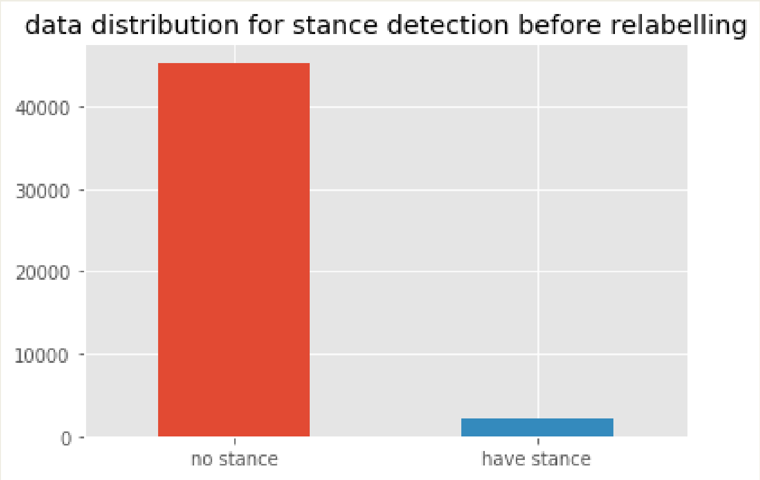
- **Problem for stance detection: Wrong labeling**

  The labeler did not label all the speeches containing stance

  Unable to train good machine learning model on a poor labeled data.

- **Methods Tried:**

  Resampling ❌,Class weight ❌



- **Observation:**

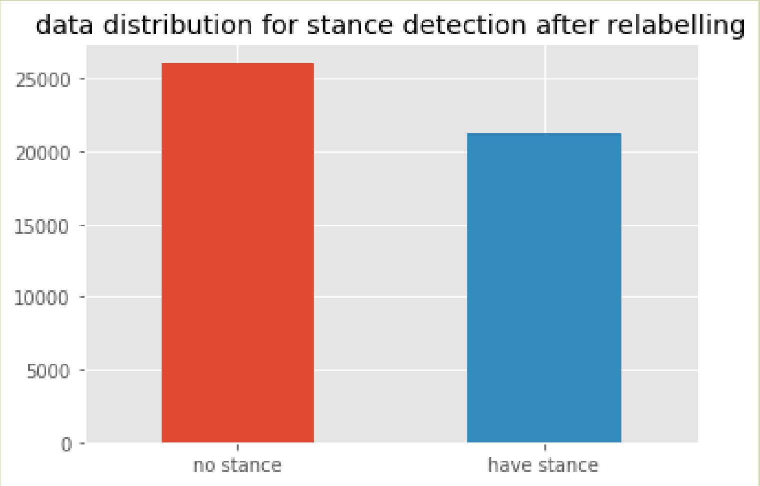
  Pattern:

  *"I rise in **support/opposition** of …"*

  often appear in the beginning.

- **Solution:**

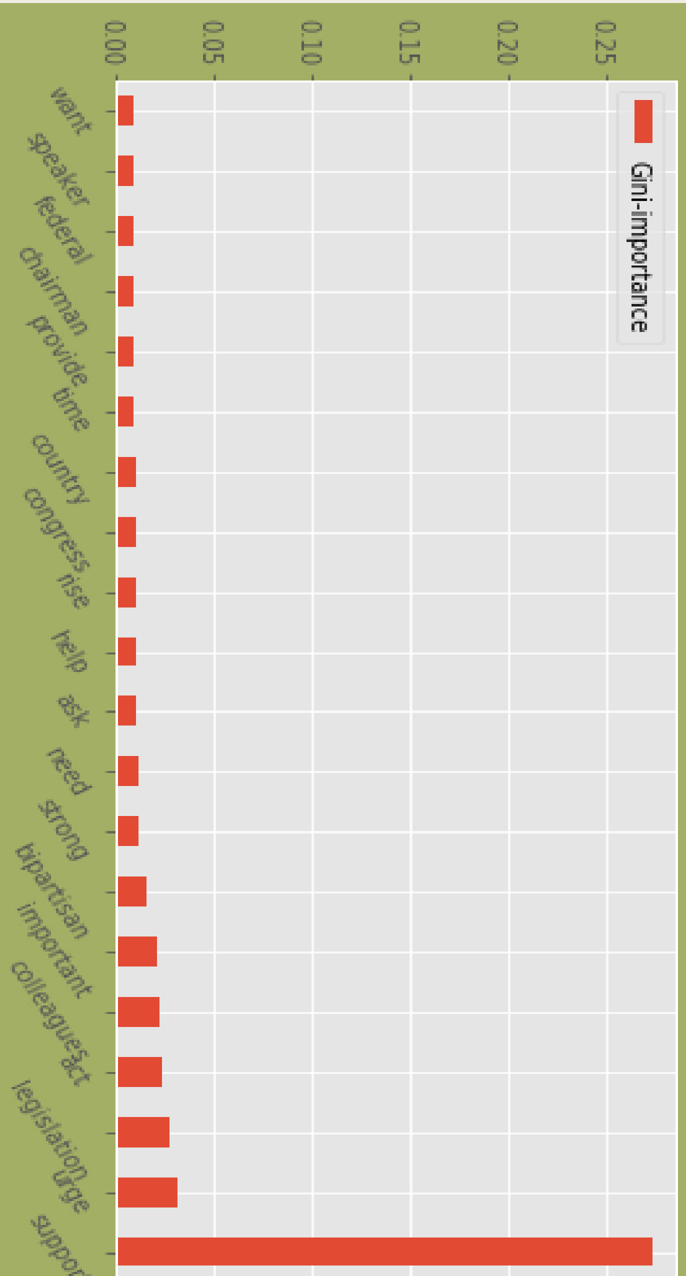
  *Use regular expression to **relabel** unlabeled data.*



Traditional Machine Learning

Balancing method	Models	F1 score (have stance)	F1 score (no stance)
Class weight	Logistic Regression	0.21	0.85
Class weight	Random Forest	0.16	0.96
Resampling	balanced ensemble method (balanced random forest)	0.22	0.83
Resampling	weighted ensemble method (weighted random forest)	0.19	0.78
Relabeling	Logistic Regression after Relabeling	0.80	0.83
Relabeling	Random Forest after Relabeling	0.88	0.91

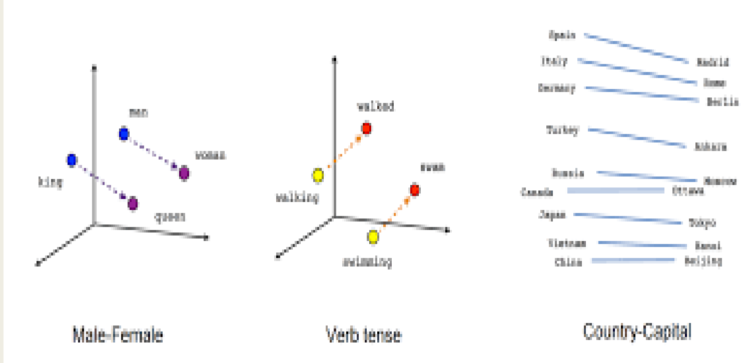
Logistic Regression & Random Forest



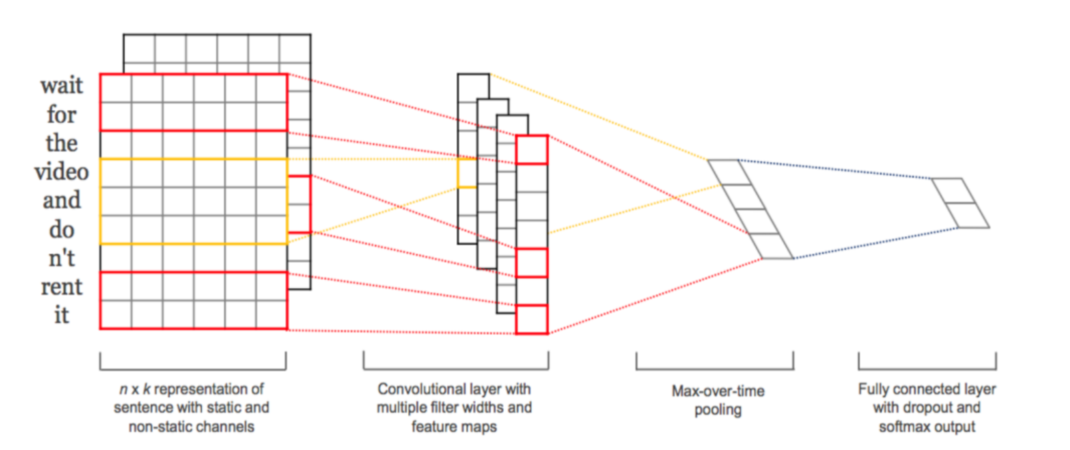
Deep Learning Models

Components for Neural Network

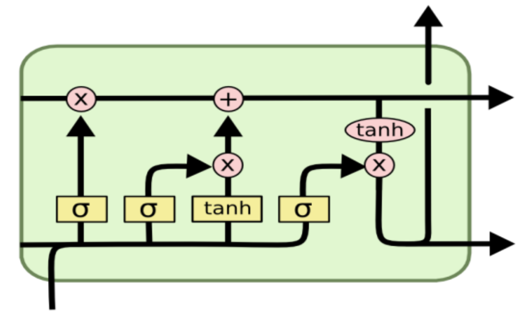
Word Embedding



Convolutional Neural Network

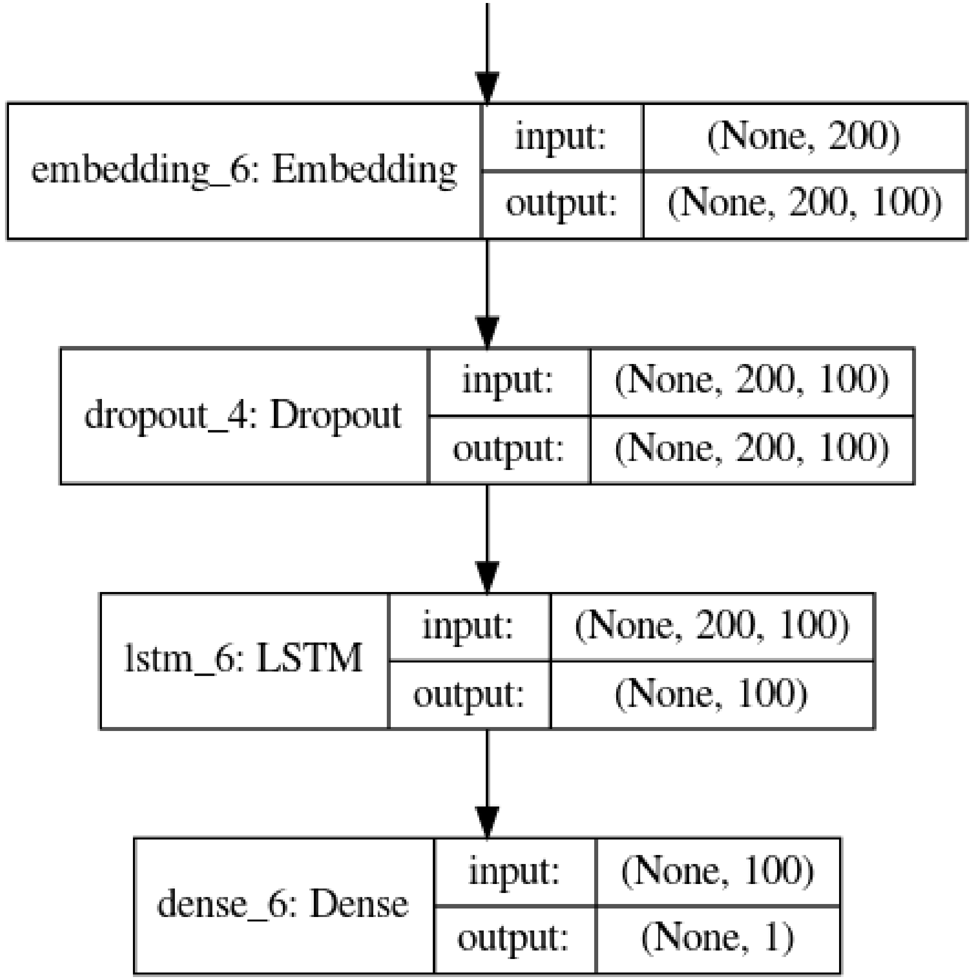


Long Short Term Memory

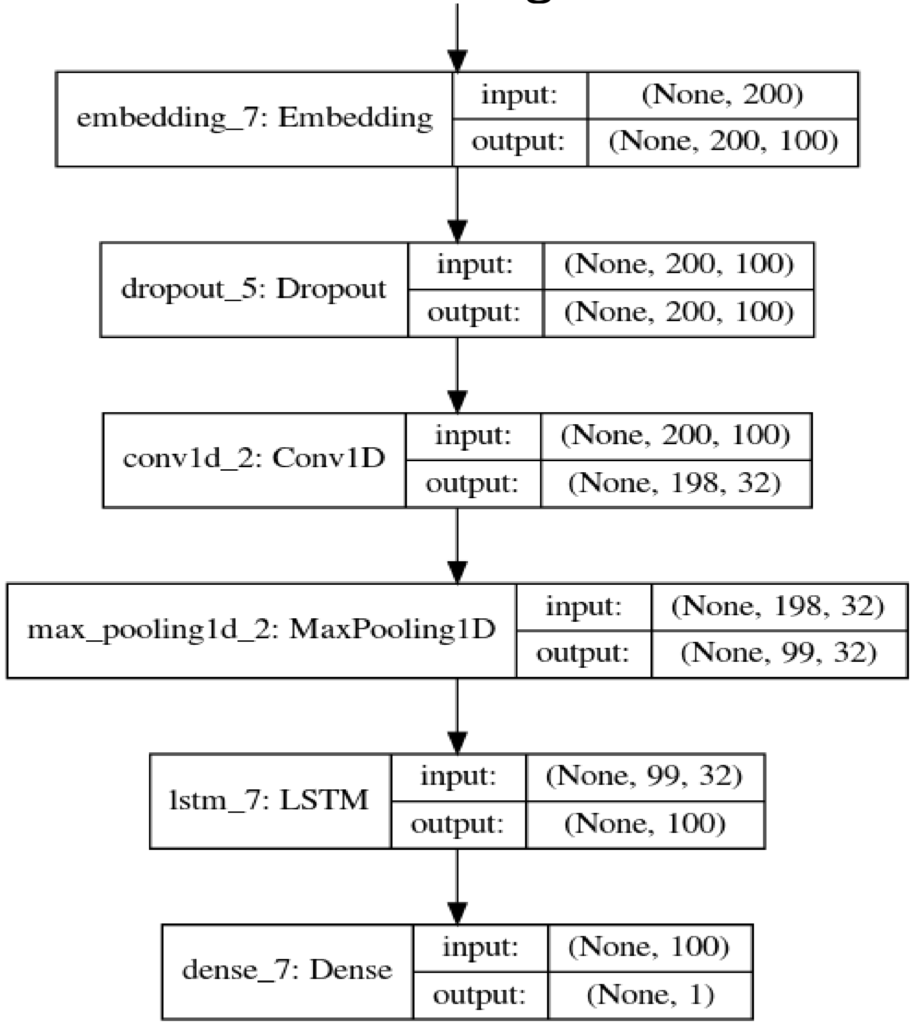


Neural Network Structures

pretrained GloVe embedding + LSTM

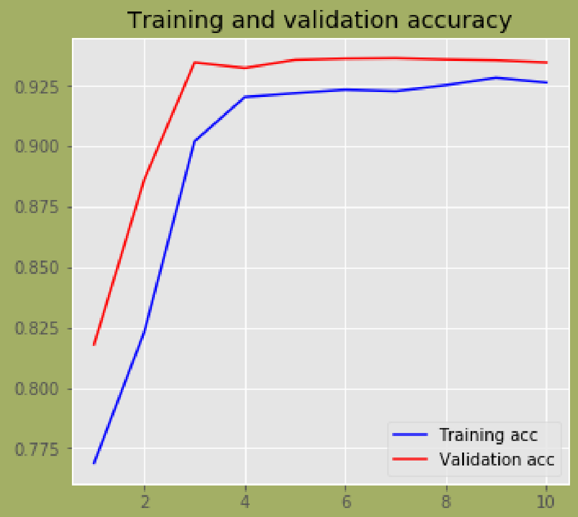


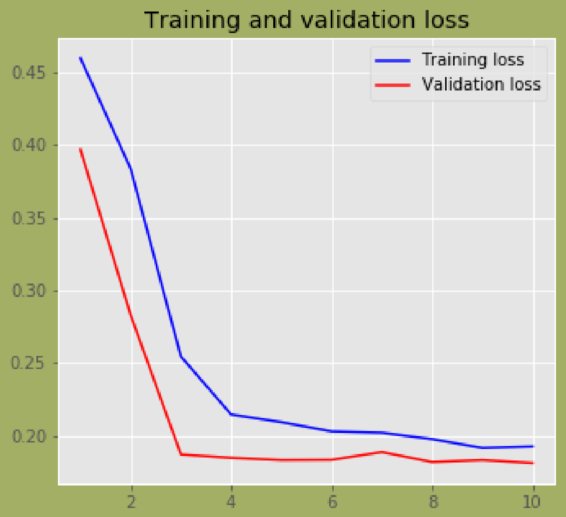
pretrained GloVe embedding + CNN + LSTM



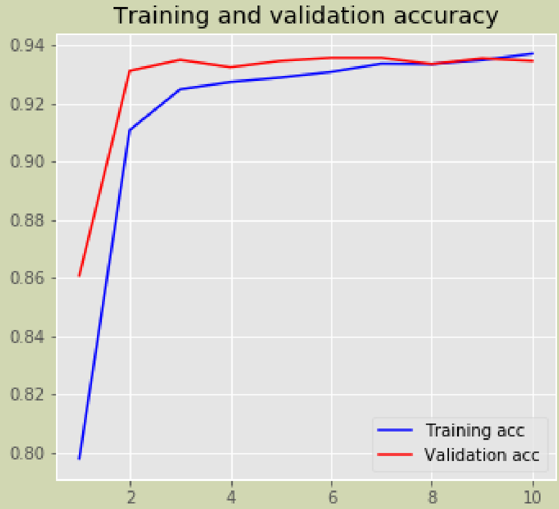
Deep Learning Result

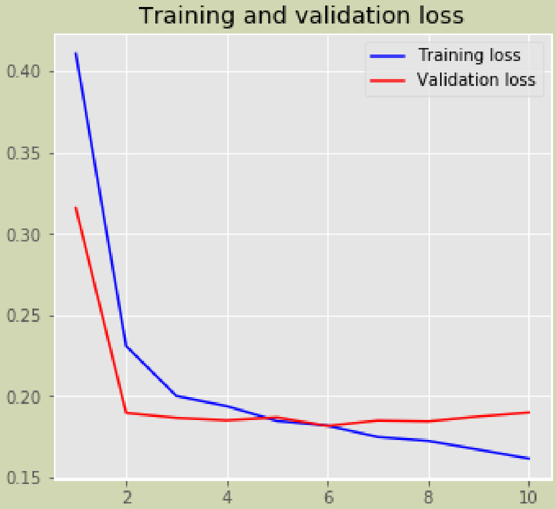
pretrained GloVe + LSTM



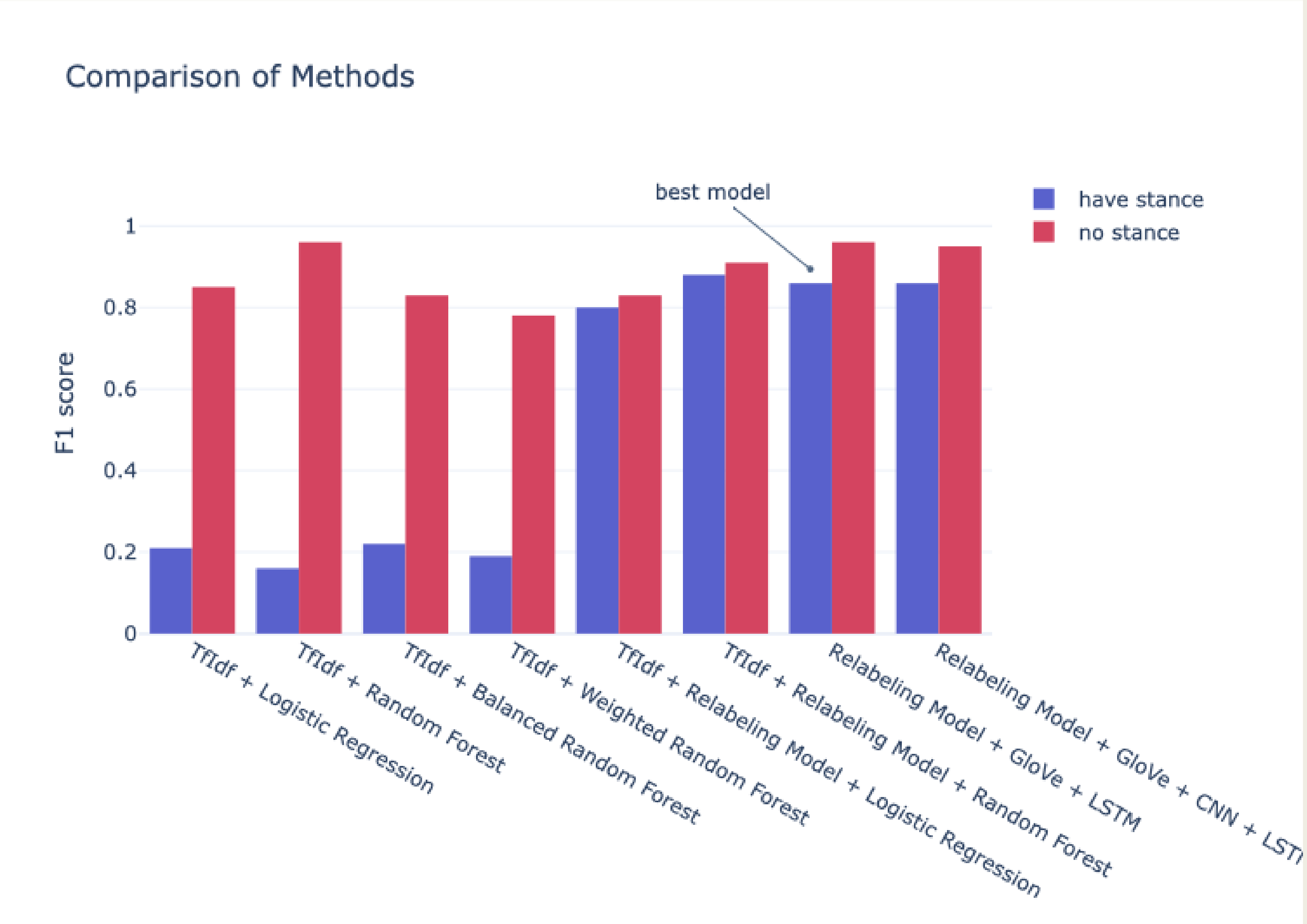


pretrained Glove + CNN + LSTM





F1 score for Stance Detection Models



See Table

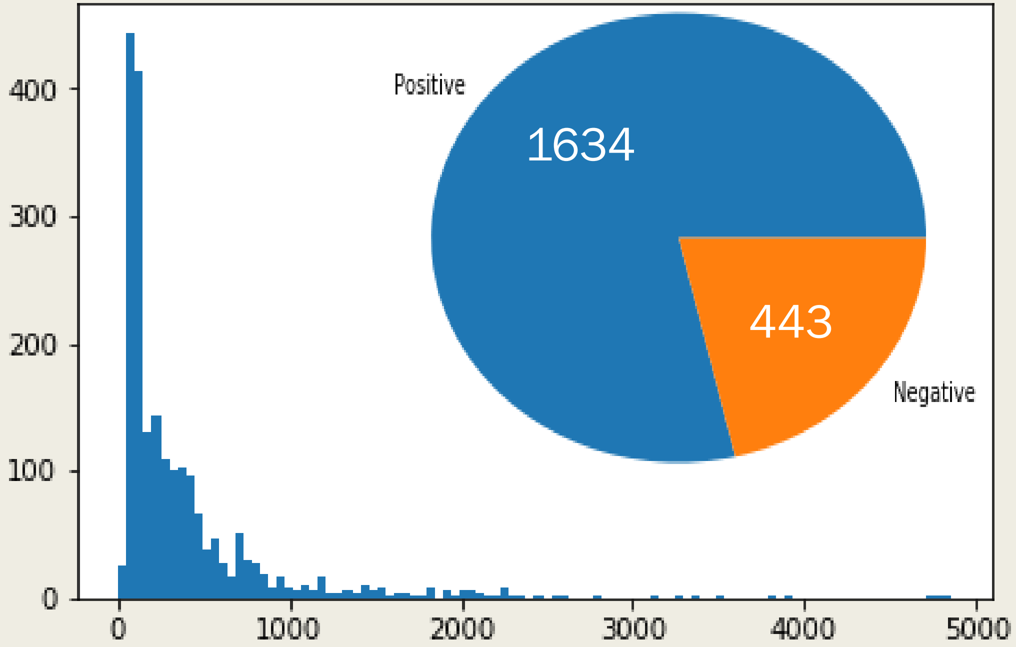
Summary for Stance Detection Models

	Performance	Interpretability	Speed
Logistic Regression (baseline model)	✗	✓	✓
Random Forest	✗	✓	✓
Balanced Random Forest	✗	✓	✓
Weighted Random Forest	✗	✓	✓
Relabeling model + Logistic Regression	✓	✓	✓
Relabeling model + Random Forest	✓	✓	✓
Pretrained GloVe word embedding + LSTM	✓	✗	✗
Pretrained GloVe word embedding + CNN + LSTM	✓	✗	✗

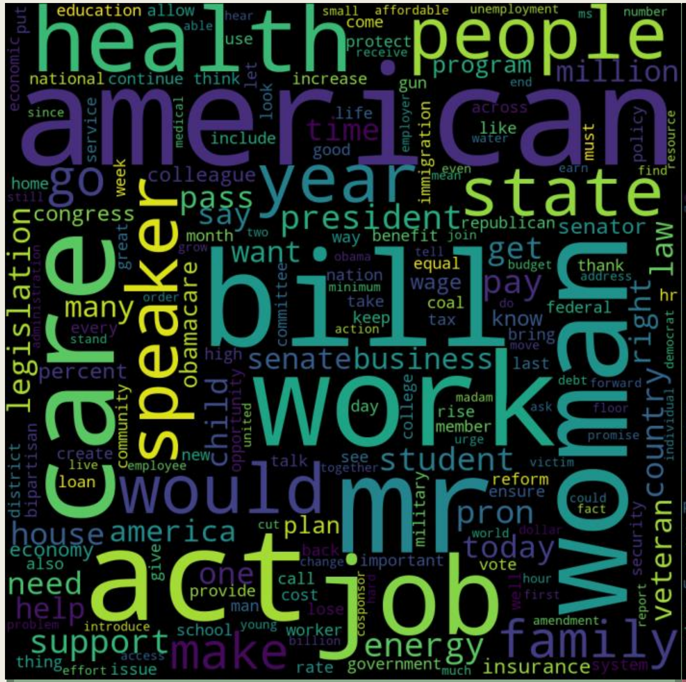
Stance Classification

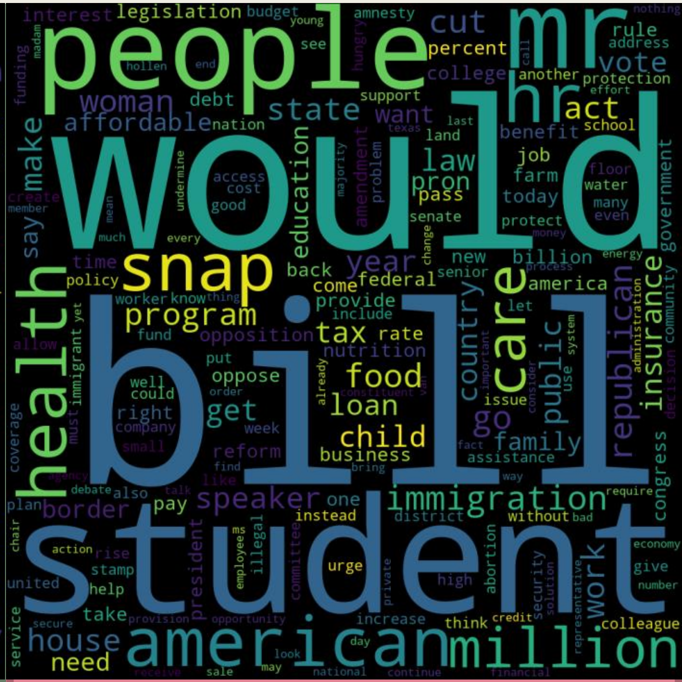
(Xiaochi Li)

Data Observation



- 75% of the speech has less than 439 words
- The class is imbalance
- Need to set cutoff, and balance data





Support Speech

Opposition Speech

Experiment Design

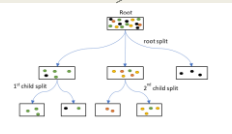
- Preprocessing:
  1. *Do nothing*
  2. *Remove stop words, punctuations and numbers*
  3. *Lemmatization*
- Vectorization:
  1. *Count Vectorization*
  2. *Tf-idf Vectorization*
- Balancing data:
  1. *Do nothing*
  2. *Balance data with SMOTE*

- Models:
  1. *Logistic Regression Model*
  2. *Random Forest*
  3. *SVM*
  4. *XGBoost*
- Method: Control Variable
- Metric: F1 score for both classes in train set and test set
- $F1 = 2 \times \frac{precision \times recall}{precision + recall}$
- We also care speed to train and interpretability

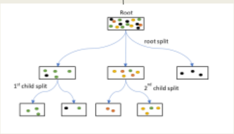
Ensemble Learning

Random Forest
Bagging

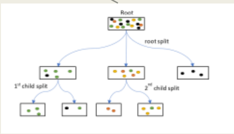
$$\hat{y} = {(\hat{y}_1 + \hat{y}_2 + \hat{y}_3)}/{3}$$



$\hat{y}_1$
target: y



$\hat{y}_2$
target: y



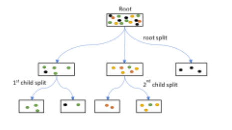
$\hat{y}_3$
target: y

- Combine trees by averaging their prediction

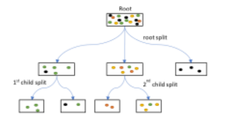
- Reduce: Variance

XGBoost (eXtreme Gradient Boosting)
Boosting

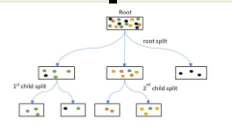
$$\hat{y} = \hat{y}_1 + \hat{y}_2 + \hat{y}_3$$



$\hat{y}_1$, target: y



$\hat{y}_2$,target: y $- \hat{y}_1$

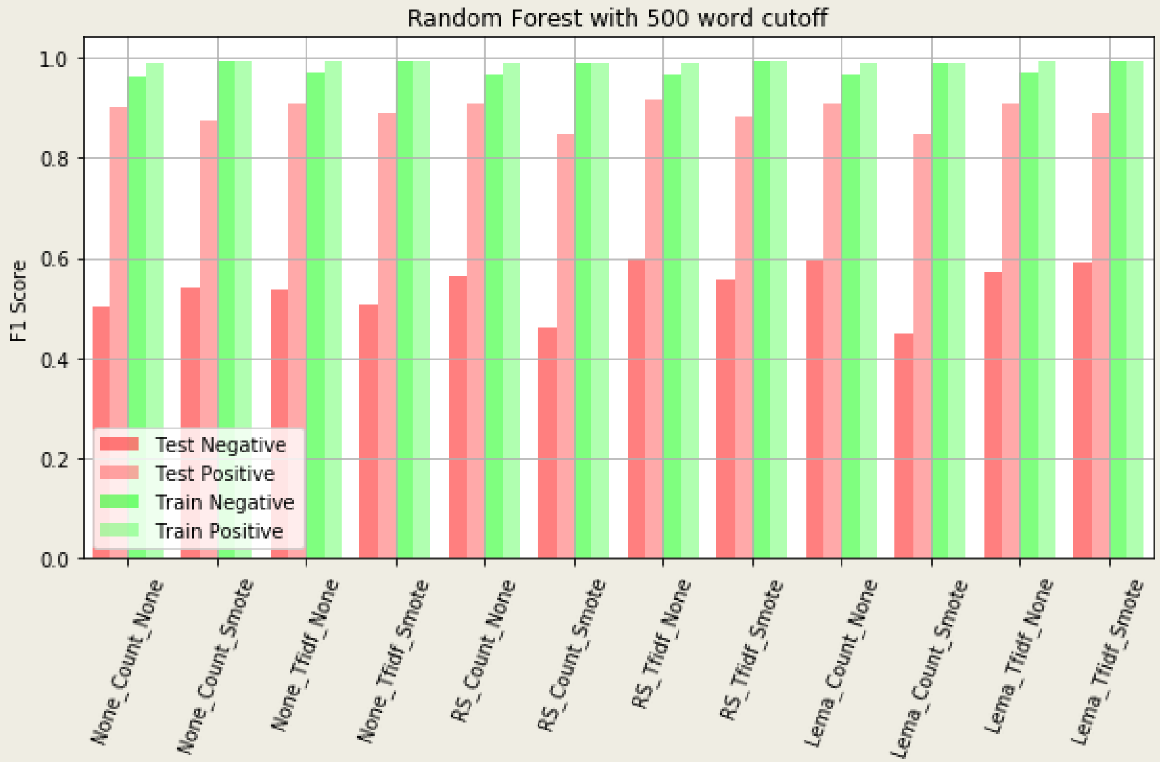


$\hat{y}_3$,target:y $- \hat{y}_1 - \hat{y}_2$

- Learn on residual, Combine trees by summing their prediction

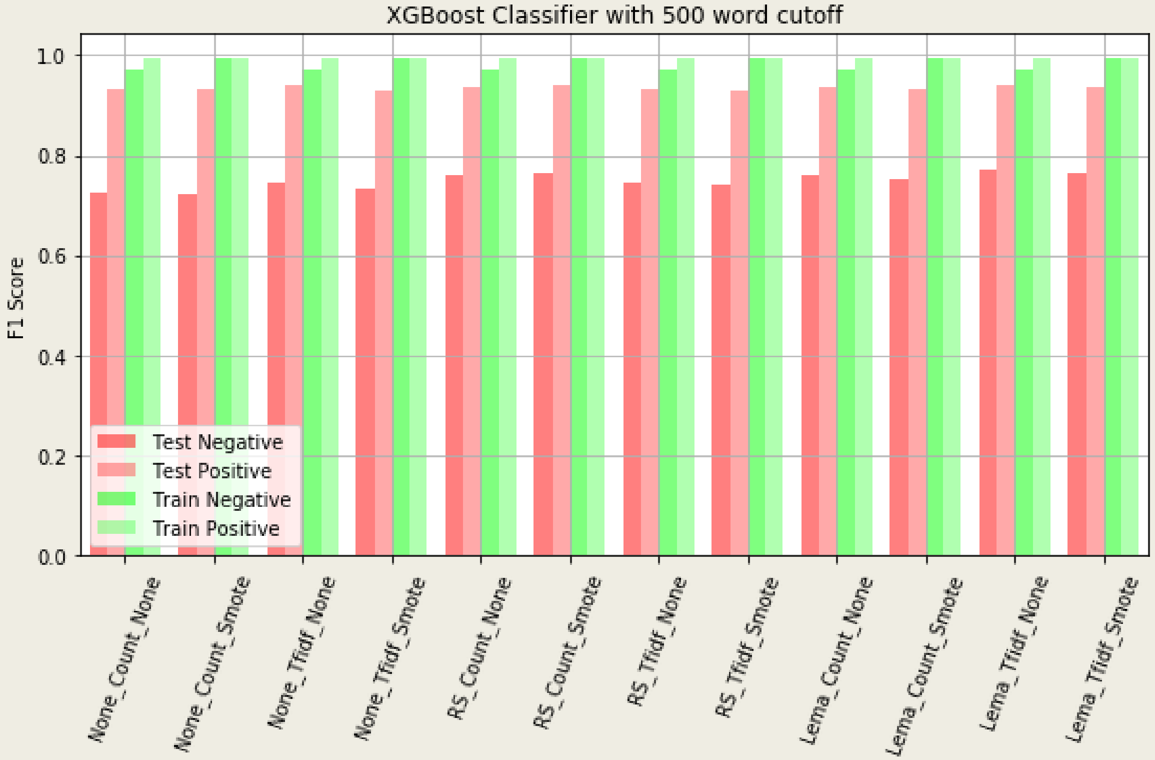
- Reduce: Bias

Random Forest



- Random Forest did not performed well
- It is still influenced by the imbalanced problem.
- Fast to train.

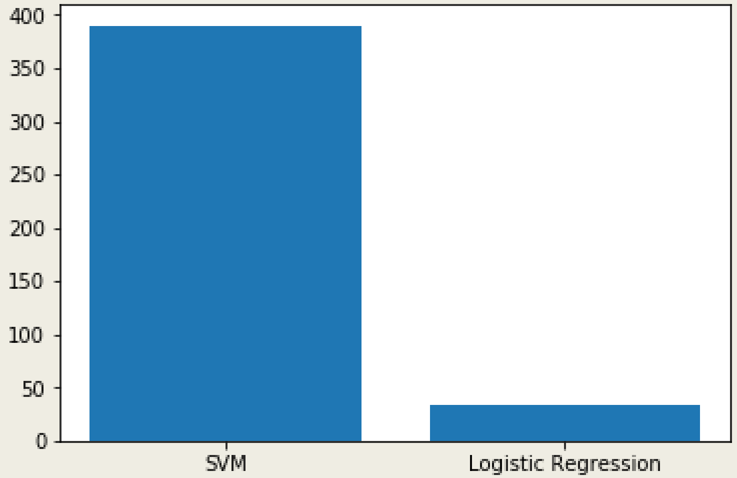
XGBoost Classifier



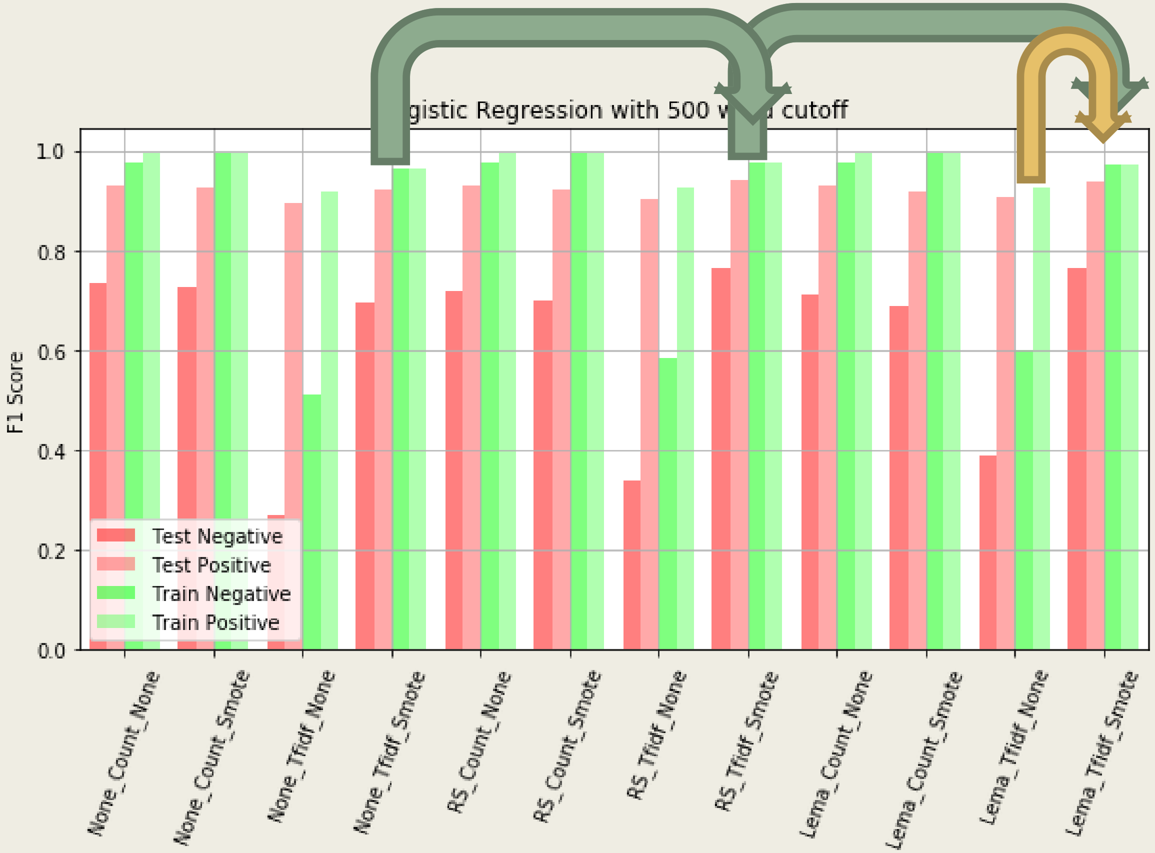
- XGBoost performs better compared to Random Forest.
- XGBoost is more focused on reducing bias, while Random Forest reduce variance in general
- Fast to train

Support Vector Machine

- SVM performs well when the train set is imbalanced
- However, it's too slow to train. (11 times slower than logistic regression)



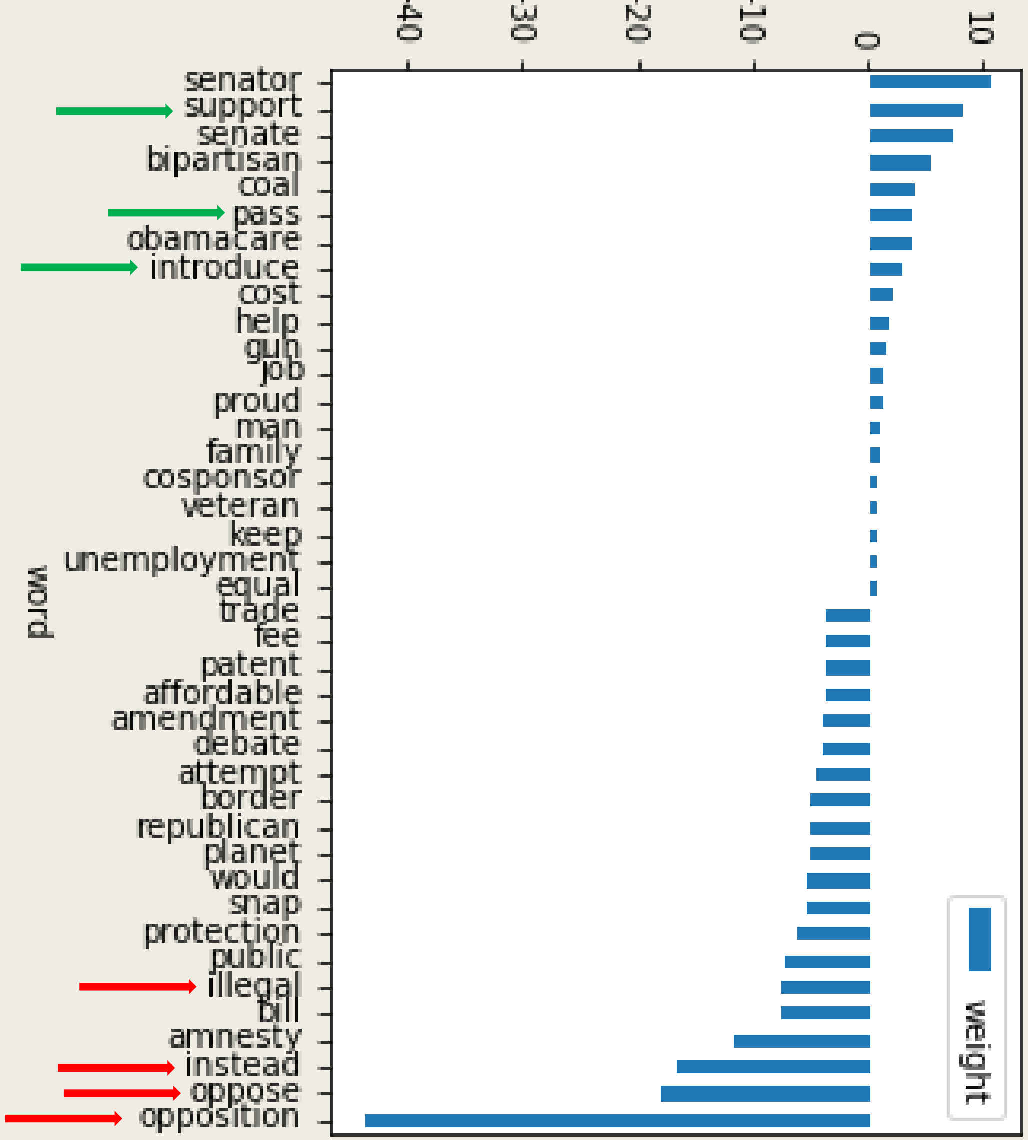
Logistic Regression



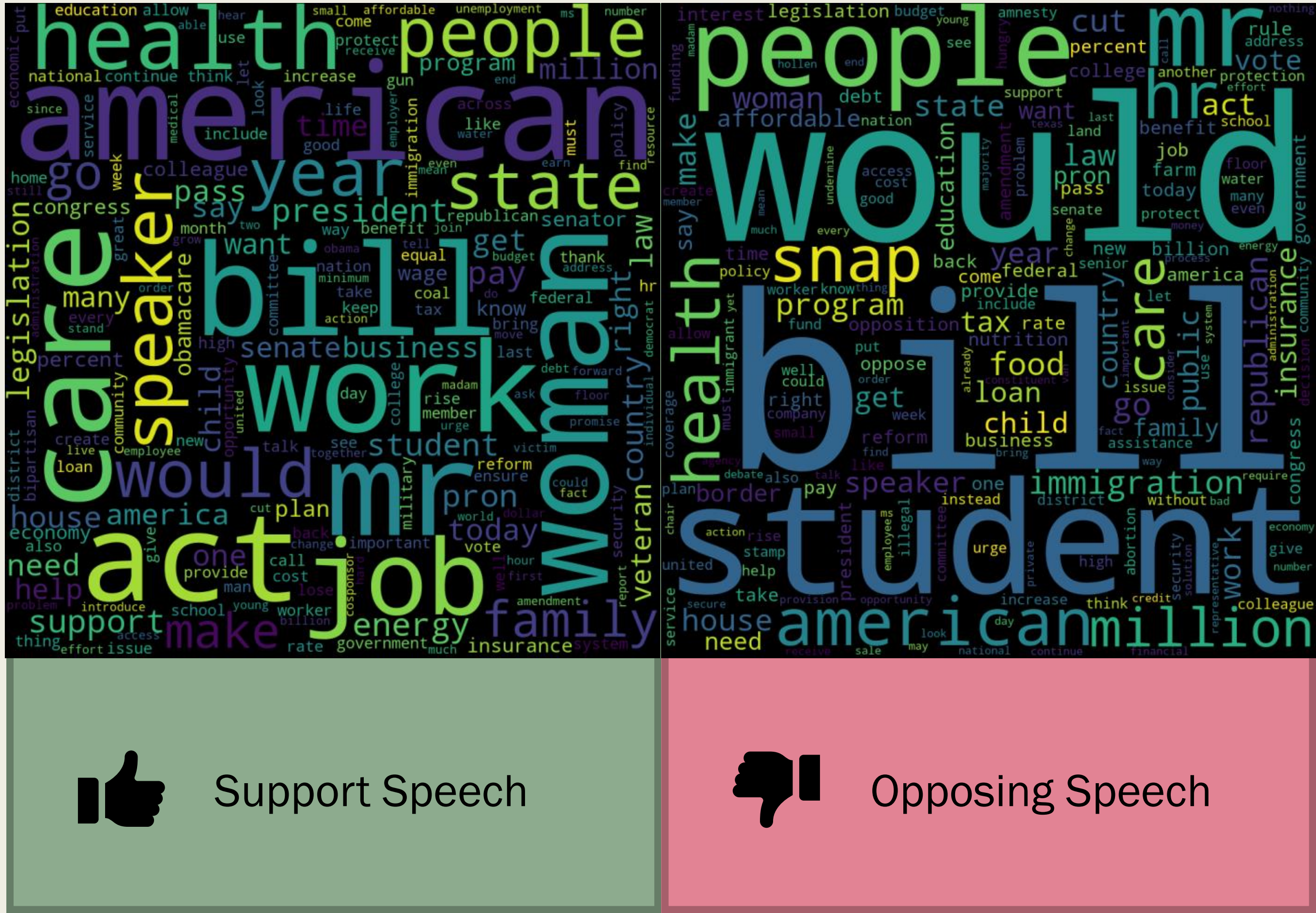
- Logistic Regression did not perform well on imbalance train set.
- However, it can reach same performance with balanced data.
- Fast to train and easy to interpret.

1. Balancing data improves most
2. TF-idf and SMOTE works well together.
3. Remove stop words contributes more than lemmatization.

Logistic Regression top features



Comparing with the word cloud,
the model has learnt something useful!



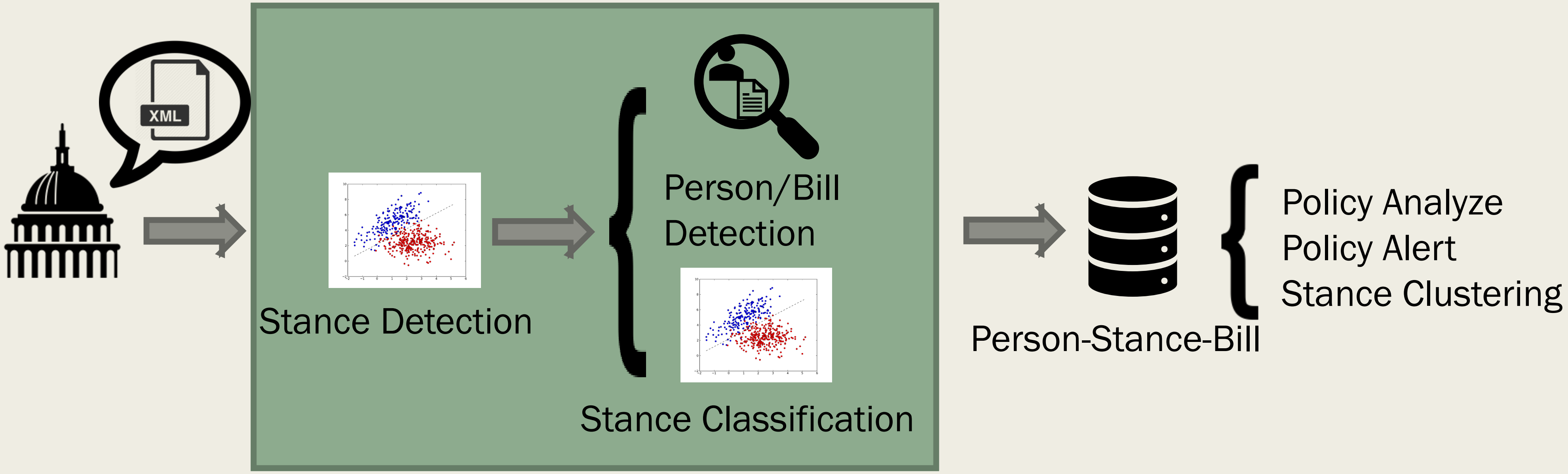
Summary for Stance Classification

Model	Performance	Interpretability	Speed
Random Forest	✗	✗	✓
XGBoost	✓	✗	✓
Support Vector Machine	✓	✓	✗
Logistic Regression	✓	✓	✓

Model	Test Negative	Test Positive	Train Negative	Train Positive
Random Forest	0.588	0.894	0.992	0.992
XGBoost	0.765	0.935	0.992	0.992
Support Vector Machine	0.768	0.941	0.987	0.987
Logistic Regression	0.756	0.936	0.969	0.968

Remove stop words + Lemmatization + Tf-idf + SMOTE

Deployment



Deploy into production:

- Well-encapsulated pipeline, input: speech, output: person-stance-bill pair
- Retrainable when more data is available

Conclusions and Learnings

- Best model:
  - *Stance Detection: pretrained GloVe embedding + LSTM*
  - *Stance Classification: Tfidf + Lemmatization + SMOTE + Logistic Regression*
- Main challenge: Data quality (mislabeled data, imbalanced data)
- Learnings:
  - *The quality of data determines the quality of model (Garbage in garbage out)*
  - *Preprocessing(Feature Engineering) is more effective than tuning the hyper parameters.*

Thank you

- Dr. Brian Wright and Dr. Vladimir A. Eidelman for offering this opportunity.
- Dr. Daniel Argyle for mentoring us during this project.
- Dr. Nima Zahadat for supporting us
- All the professors in the Data Science Program.

Backup Pages
We are ready for questions.

- SMOTE
- Remove stop words, lemmatization
- Count Vectorization
- Tf-idf Vectorization
- GloVe

GloVe

- GloVe is an unsupervised learning algorithm for obtaining vector representations for words developed by stanford NLP group.
- Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

GloVe + LSTM

```
glove_model = models.Sequential()

glove_model.add(Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=200, trainable=False))

glove_model.add(Dropout(0.2))

glove_model.add(LSTM(100, dropout=0.5, recurrent_dropout=0.2))

glove_model.add(layers.Dense(1, activation='sigmoid'))
```

GloVe + CNN + LSTM

```
model_conv = Sequential()
model_conv.add(Embedding(vocab_size, 100, input_length=200))
model_conv.add(Dropout(0.2))
model_conv.add(Conv1D(32, 3, activation='relu'))
model_conv.add(MaxPooling1D(pool_size=2))
model_conv.add(LSTM(100))
model_conv.add(Dense(1, activation='sigmoid'))
model_conv.layers[0].set_weights([embedding_matrix])
model_conv.layers[0].trainable = False
model_conv.compile(loss='binary_crossentropy', optimizer='adam',metrics=['accuracy'])
```

Backup
What's SMOTE?

- Synthetic Minority Over-sampling Technique

- Generate new samples for minority class between the existing samples

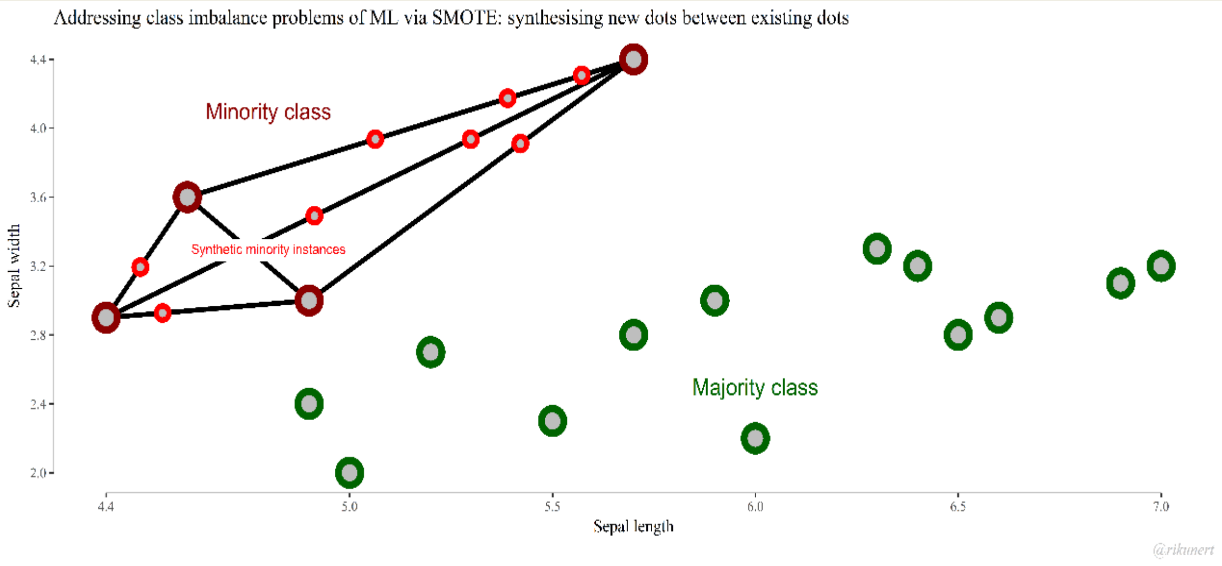


Image from: http://rikunert.com/SMOTE_explained

Backup
What's Remove stop words, Lemmatization?

```
In [1]: from preprocess_utility import *

In [2]: text = 'This is a good 123 TEST. walk walking walked'

In [3]: t= remove_stopwords(text)
        t
Out[3]: 'good test walk walking walked'
```

Remove Stop words

```
In [4]: spacy_lemma(t)
Out[4]: 'good test walk walk walk'
```

Lemmatization

Backup
What's Count Vectorization?

- *Text = "It was the best of times"*
  "It was the worst of times"
  "It was the age of wisdom"
  "It was the age of foolishness"
- *Features = ['It', 'was', 'the', 'best', 'of', 'times', 'worst', 'age', 'wisdom', 'foolishness']*
- Vectors =

  "It was the best of times" = [1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
  "It was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
  "It was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
  "It was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428

What’s Tf-idf vectorization?

term frequency–inverse document frequency

Example of tf–idf [edit]

Suppose that we have term count tables of a corpus consisting of only two documents, as listed on the right.

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

The calculation of tf–idf for the term "this" is performed as follows:

In its raw frequency form, tf is just the frequency of the "this" for each document. In each document, the word "this" appears once; but as the document 2 has more words, its relative frequency is smaller.

$$\mathrm{tf}(''\text{this}'', d_1) = \frac{1}{5} = 0.2$$

$$\mathrm{tf}(''\text{this}'', d_2) = \frac{1}{7} \approx 0.14$$

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

An idf is constant per corpus, and **accounts** for the ratio of documents that include the word "this". In this case, we have a corpus of two documents and all of them include the word "this".

$$\mathrm{idf}(''\text{this}'', D) = \log\left(\frac{2}{2}\right) = 0$$

$$IDF(t) = log_e\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it}\right)$$

So tf–idf is zero for the word "this", which implies that the word is not very informative as it appears in all documents.

$$\mathrm{tfidf}(''\text{this}'', d_1, D) = 0.2 \times 0 = 0$$
$$\mathrm{tfidf}(''\text{this}'', d_2, D) = 0.14 \times 0 = 0$$

The word "example" is more interesting - it occurs three times, but only in the second document:

$$\mathrm{tf}(''\text{example}'', d_1) = \frac{0}{5} = 0$$

$$\mathrm{tf}(''\text{example}'', d_2) = \frac{3}{7} \approx 0.429$$

$$\mathrm{idf}(''\text{example}'', D) = \log\left(\frac{2}{1}\right) = 0.301$$

Finally,

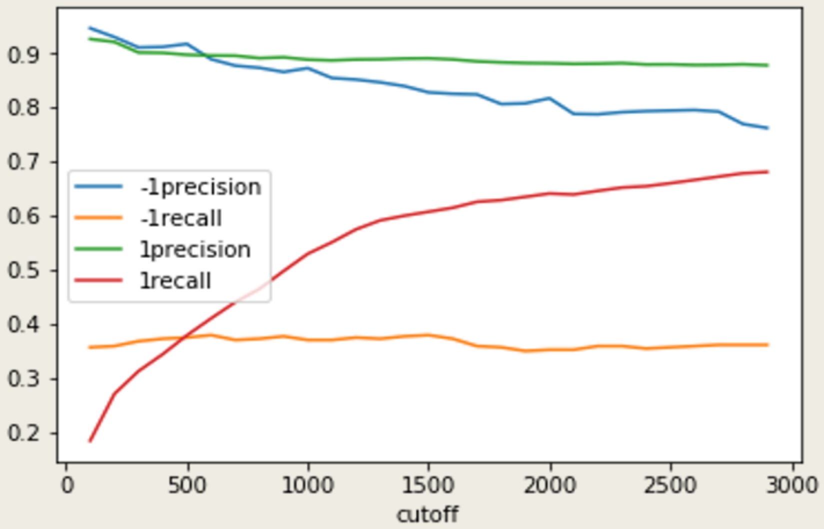
$$\mathrm{tfidf}(''\text{example}'', d_1, D) = \mathrm{tf}(''\text{example}'', d_1) \times \mathrm{idf}(''\text{example}'', D) = 0 \times 0.301 = 0$$
$$\mathrm{tfidf}(''\text{example}'', d_2, D) = \mathrm{tf}(''\text{example}'', d_2) \times \mathrm{idf}(''\text{example}'', D) = 0.429 \times 0.301 \approx 0.129$$

$$TF - idf\ score = TF \times IDF$$

Experiment on traditional balancing methods

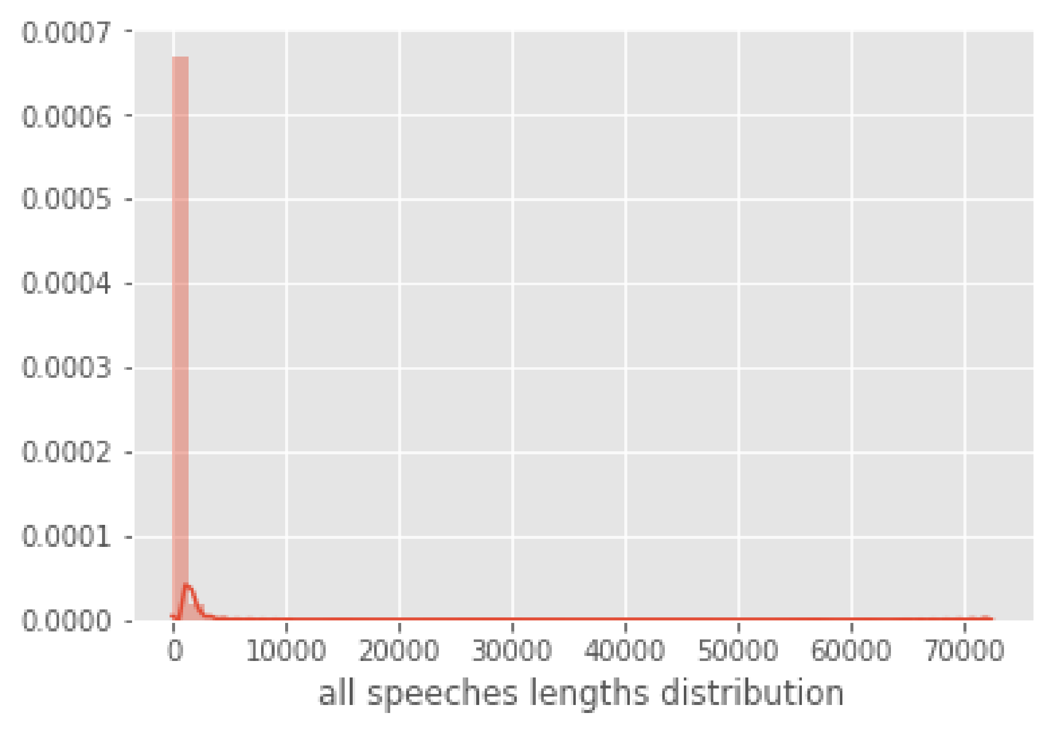
Models / Metrics	F1 score (have stance)	F1 score (no stance)
Logistic Regression before Relabelling	0.21	0.85
Random Forest before Relabelling	0.16	0.96
balanced ensemble method (balanced random forest)	0.22	0.83
weighted ensemble method (weighted random forest)	0.19	0.78

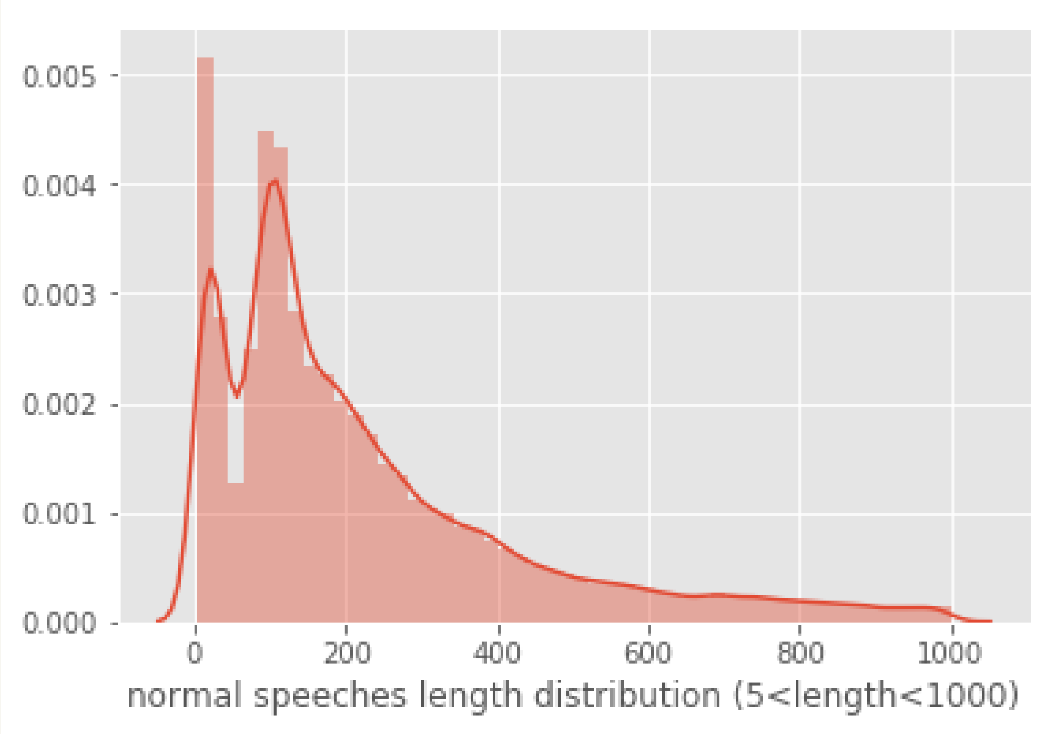
Logistic Regression Relabeling Accuracy



- We can get a good precision(~0.9) with a strict keyword detection range (500 characters from begin)
- Trade “quantity” for “quality”.

Speech Length Observation





	Speech Length
Max	79382
Min	2
Avg	346

Examples of speeches (length< 5):
"Mr Heller yield floor"

Examples of speeches (length< 15):
"Had I been here, I would vote yes"

Summary for Stance Detection Models

Models / Metrics	F1 score (have stance)	F1 score (no stance)
Logistic Regression	0.21	0.85
Random Forest	0.16	0.96
balanced ensemble method (balanced random forest)	0.22	0.83
weighted ensemble method (weighted random forest)	0.19	0.78
Logistic Regression after Relabelling	0.80	0.83
Random Forest after Relabelling	0.88	0.91
Pretrained GloVe + LSTM	0.86	0.96
Pretrained GloVe + CNN + LSTM	0.86	0.95