# CS5228 Final Project

Team 48

Our team is made up of 3: Chen Song, Zhang Junda, and Miao Yisong

# Our Project Timeline

1. We first use Naive Bayes to set a baseline （0.84 level）

2. We parallel refine our feature engineering, model selection, and obtaining the webpage content

3. On Tuesday, only with title, we reached 0.875, **how?**

4. And then we was stucked at 0.875, **why?**

5. On the last day, we finally find an amazing phenomenon(**what?**), and finished at 0.896!
(1.1% higher than Guan's Team at 3rd place :PP 😝😝)

# The 0.875 model – Feature engineering

In all model we used, we only use word as feature. (**why?**)
So a good tokenizer is very important!

Here the tokenizer goes:

1. Lowercase

2. Find all string made up of 2+ [A-Za-z] (means we ignore all symbols like **$%&**, **why?)**

3. For everything left, we use nltk stemming to process. (**why not lemmatize?** )

4. N-gram feature selection: Unigram and Bigram (by local experiment)

**Interesting discovery: Stem makes 1% better, using Bigram makes 2% better, why?**

# Why stem is good?


Stem can have better effect on word(i.e. feature) clustering
It is quite effective when the sample size is not large

# Why Bigram is good?

['High Level', ]

**Intuition:**

When 2 word combine together, it is more important.

Especially good when w1, w2 has high document frequency,
but [w1, w2] is much lower.

**Logical thinking:**

Bigram is actually considering the interaction between unigram features.
Can try Factorization Machine also, but we haven't had time.

# The 0.875 model – Model Selection

We basically tried all the classification method taught in class, namely:
Naive Bayes, Boosting Tree, SVM, etc

Finally we found SVM has the best performance(1% higher than others in local cv)

Advice for model selection:

- Can use sk-learn grid search to automatically find the optimal parameter!

Trick for SVM:
**loss='hinge', penalty='l2'**

# Why we stuck at 0.87 level for few days?

With pure title as input, we tried ensemble those model in last page together, makes no improvement.

**Why?**

**Divergence between different model's prediction is too low!**

# Parallel Working – get the webpage content

We use the URL to obtained the webpage content, but only 55% are valid.

The 45% falls to [404 error, 503 error, permission denied, content removed etc etc]

We used a simple way to prune those 45% (webpage content length < 500)

# Another 0.874 level model

Our group obtained the webpage content of the samples (only 55% are valid though)

We use title + webpage content as input, with exactly the same SVM, to reach 0.874.

**Why webpage content is so important?**

# Why webpage content is so important?

## Title: Saving Catcher

March 22, 2014 at 12:00a.m.


**Pure title prediction: 4(others)**
**With webpage prediction: 3(new product coverage)**

Associated Press

NEW YORK

Wal-Mart told The Associated Press that it has rolled out an online tool that allows shoppers to compare its prices on 80,000 food and household products to those of its competitors. The world's largest retailer began offering the feature that's called "Savings Catcher" on its website late last month in seven big markets that include Dallas, San Diego and Atlanta.

The move by Wal-Mart, which has a long history of undercutting competitors, could change the way people shop and how other retailers price their merchandise. After all, Americans already increasingly are searching for the lowest prices on their tablets and smartphones while in checkout aisles.

# The amazing finding!

On the prediction, although **pure title** and **title + webpage content** all reached ~ 0.875

The two model prediction diverge by **10%!!!**

This finding is too strong for us to ignore!

So in this dataset, we conclude that:
The divergence between different model is much smaller than that between different input,
It directs us to the right method of ensemble!

# A tough question, how to ensemble 2 model?

Generally we need >= 3 model to ensemble,
but now we only have 2.

We spent half day finding the third model who has 2 properties:
- Also reach 0.875 level
- Diverge a lot(like 10%) with our existing 2 model.

We tried hard, but cannot find such model

# Rule-based 2-model ensemble

Later, we compromise to find a workable solution:
Use a rule based method to ensemble the existing 2 models.

The rule is as follows:

| Model 1- Pure title | Model 2- Titile + webpage content | Output |
|---|---|---|
| s | s | s |
| 4 | x (!= 4) | x |

**Why this rule works?**

When model 1 predicts a sample to 4(others), it is due to the lack of evidence(feature) to make a good prediction.
So when the prediction diverge in this case, we listen to model 2 (who has more evidence).

# Future work – better ensemble

1. We can find the 3rd classifier who meets such criteria:

- Also reach 0.875 level
- Diverge a lot(like 10%) with our existing 2 model.

2. More sophisticated rule:

We can not only consider the (4, x) rule, but also consider the other features (like length of topic string)

# Future work – feature engineering

We didn't consider many good properties from the NLP perspective,

- Named entity as feature (there're many company name, country name, number, money etc etc)

- Part of speech as feature (Maybe different sentence structure of title can lead to different class?)

- Sentiment analysis. (We can clearly see that in class 1, the sentiment is very negative)

- Verb analysis. (The key meaning of the topic was conveyed by 1~2 verb in the title!)

- Anything else?

# Thank You!