

# Real & Fake News Classification: How & Why

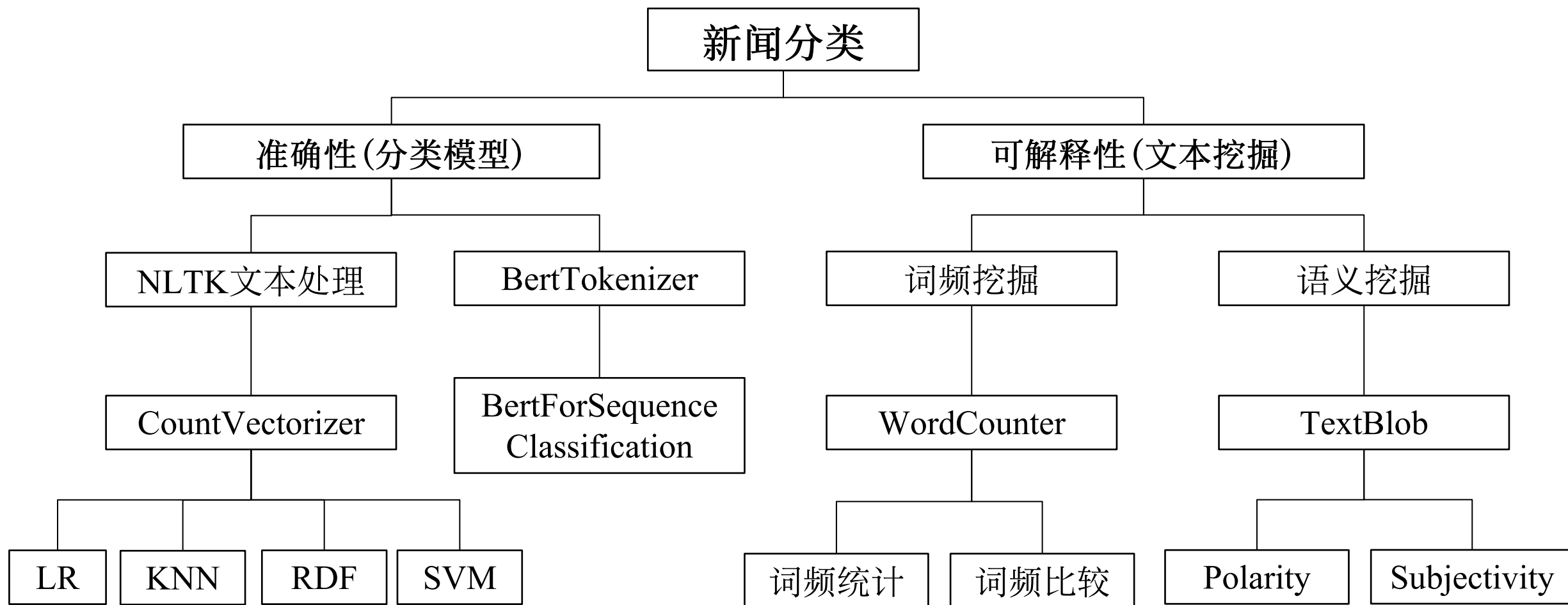
赵煜东

12月22日

To Begin With...

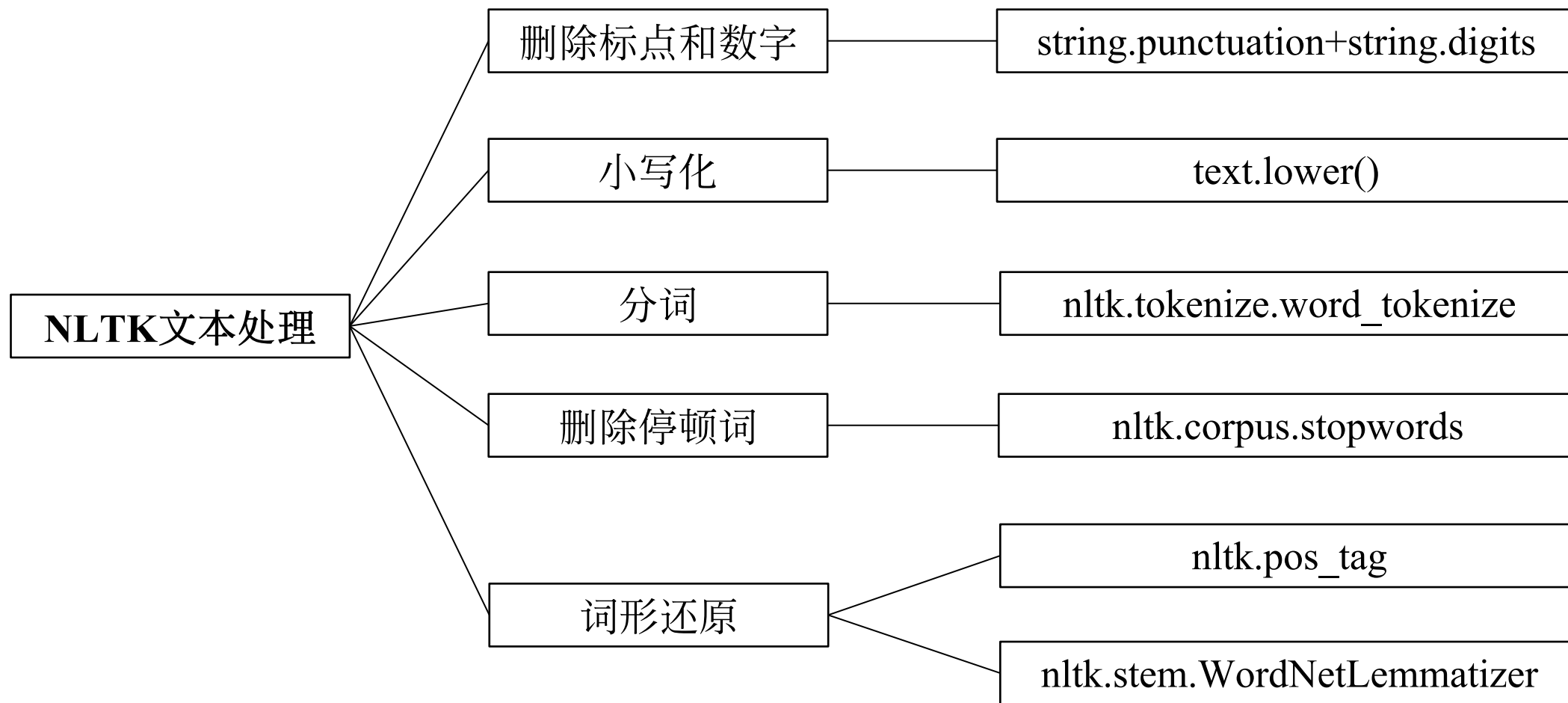
分类准确性 vs 可解释性?

# My Framework



如何分类?——How

# NLTK文本处理



# CountVectorizer

**功能：**先拟合提取出样本的整体特征，再进行降维等操作将文本向量化。

```
[25]: from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()

corpus = [
    'This is the first document.',
    'This document is the second document.',
    'And this is the third one.',
    'Is this the first document?',
]

X = vectorizer.fit_transform(corpus)

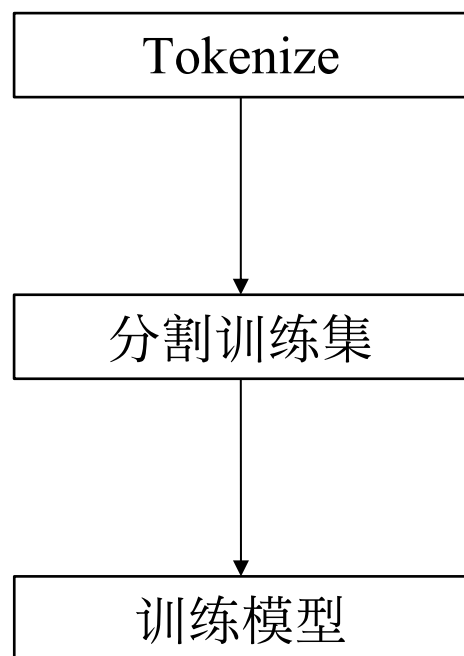
print(X.toarray())

[[0 1 1 1 0 0 1 0 1]
 [0 2 0 1 0 1 1 0 1]
 [1 0 0 1 1 0 1 1 1]
 [0 1 1 1 0 0 1 0 1]]
```

# Classification Model

	Model	Avg_Accuracy	Avg_Precision	Avg_Recall	Avg_F1Score
0	Logistic Regresstion	0.995	0.994	0.995	0.995
1	Random Forest	0.991	0.991	0.991	0.990
2	SVM	0.995	0.993	0.995	0.995
3	KNN	0.617	0.563	0.719	0.719

# Bert



```
[7]: from transformers import BertTokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-cased')#选用large-cased版本

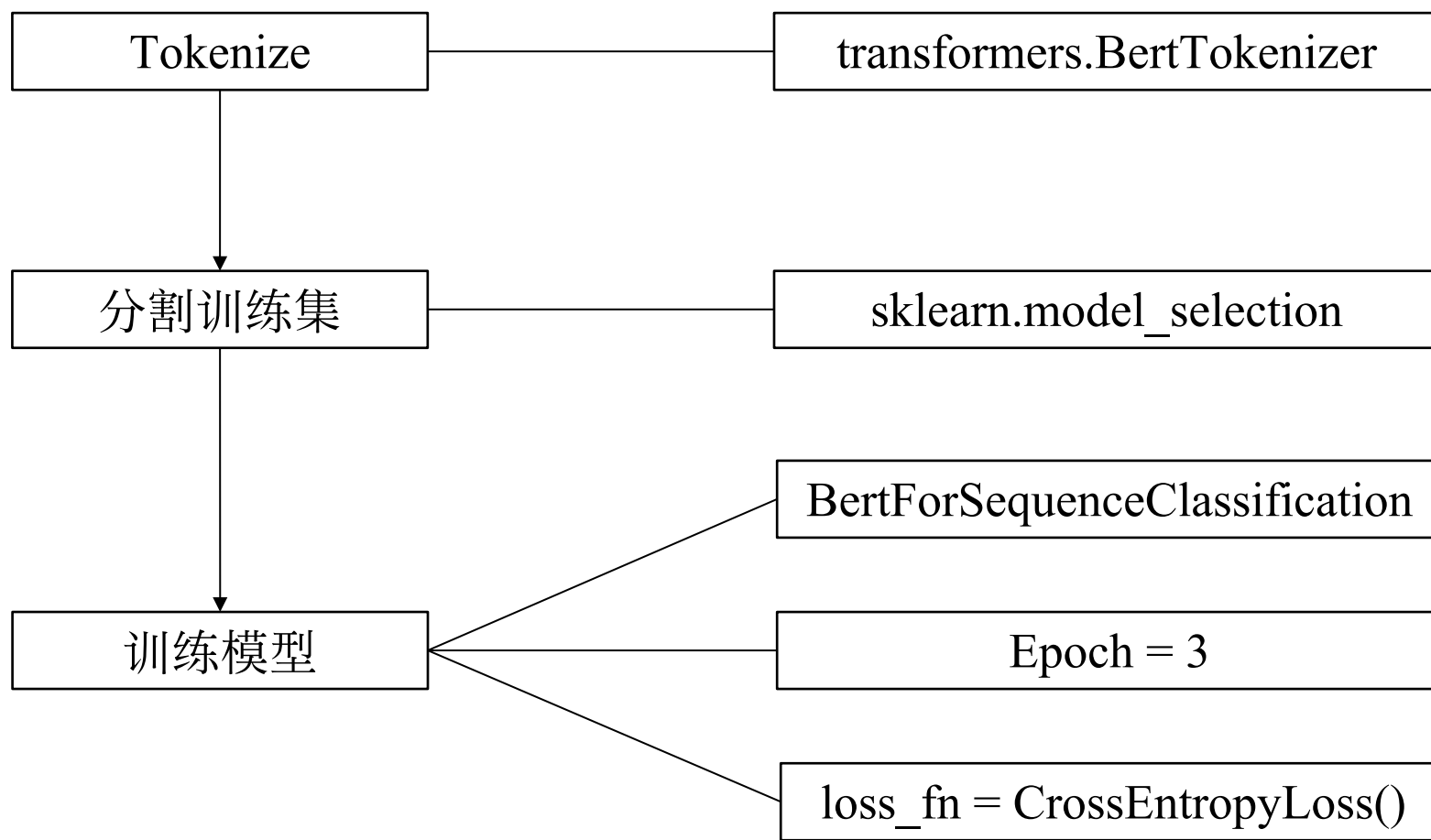
tokenized_text = tokenizer('This document is the second document.',
                           padding='max_length', truncation=True,
                           max_length=16, return_tensors='pt')

print(tokenized_text['input_ids'])

tensor([[ 101, 1188, 5830, 1110, 1103, 1248, 5830, 119, 102,  0,  0,  0,
          0,  0,  0,  0]])
```



# Bert



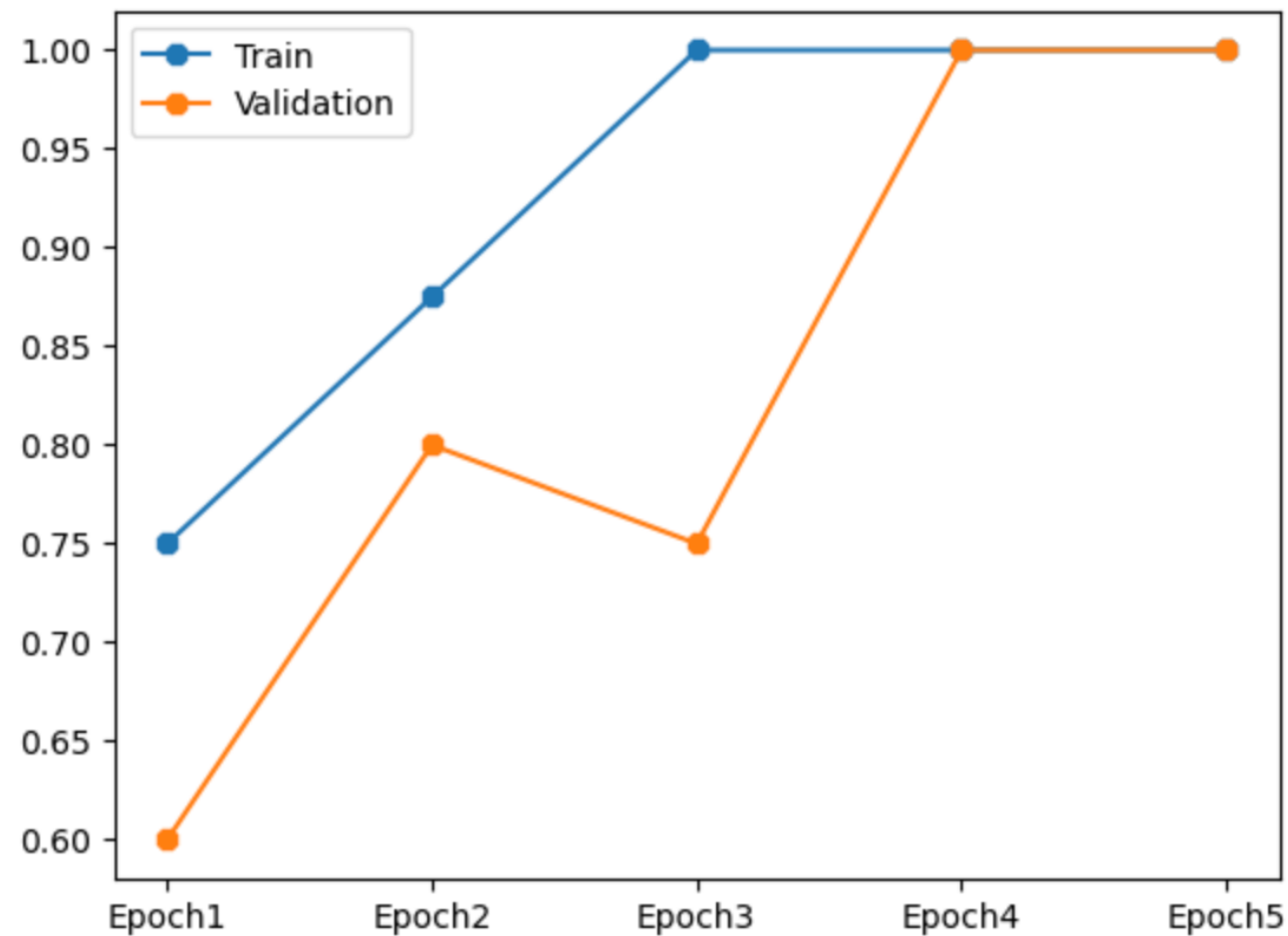
# Bert-MiniSample

样本大小：50条真、50条假

训练集：80      测试集：20

训练轮数：5

训练时间：约10分钟



# Bert-LargeSample

样本大小：1000条真、1000条假

训练集：1600      测试集：400

训练轮数：3

训练时间：约7.5小时

```
Epoch 1/3: 100%|██████████| 50/50 [2:38:47<00:00, 190.54s/it]
```

```
Epochs: 1
```

```
| Train Accuracy: 1.000  
| Train Precision: 1.000  
| Train Recall: 1.000  
| Train F1Score: 1.000  
| Val Accuracy: 1.000  
| Val Precision: 1.000  
| Val Recall: 1.000  
| Val F1Score: 1.000
```

```
Epoch 2/3: 100%|██████████| 50/50 [2:38:58<00:00, 190.76s/it]
```

```
Epochs: 2
```

```
| Train Accuracy: 1.000  
| Train Precision: 1.000  
| Train Recall: 1.000  
| Train F1Score: 1.000  
| Val Accuracy: 1.000  
| Val Precision: 1.000  
| Val Recall: 1.000  
| Val F1Score: 1.000
```

```
Epoch 3/3: 100%|██████████| 50/50 [2:37:58<00:00, 189.57s/it]
```

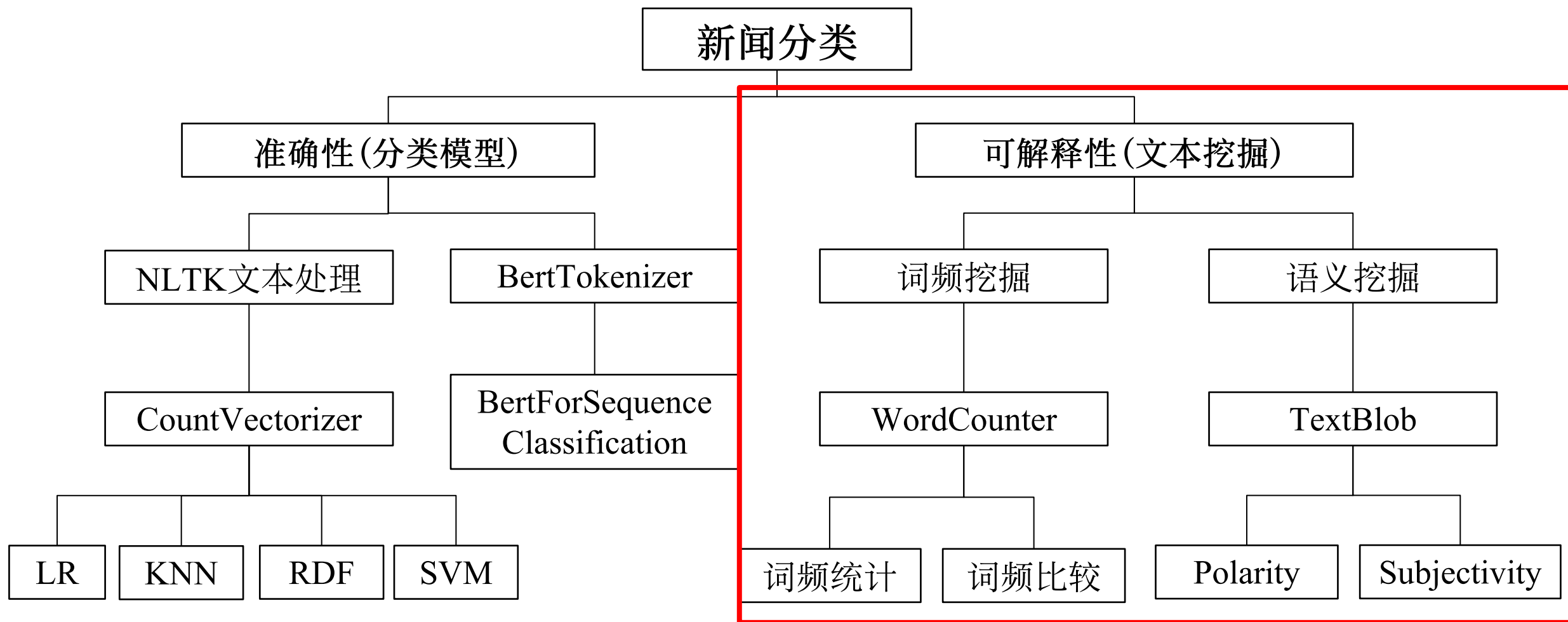
```
Epochs: 3
```

```
| Train Accuracy: 1.000  
| Train Precision: 1.000  
| Train Recall: 1.000  
| Train F1Score: 1.000  
| Val Accuracy: 1.000  
| Val Precision: 1.000  
| Val Recall: 1.000  
| Val F1Score: 1.000
```

But...

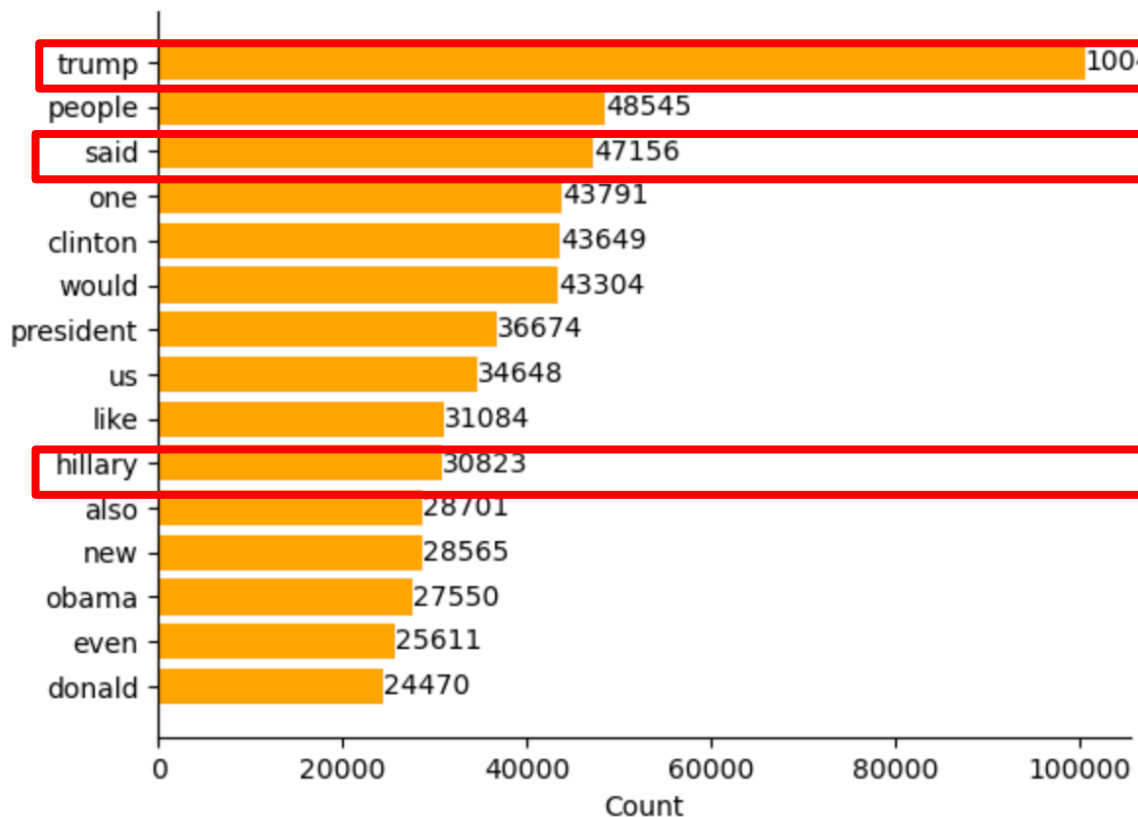
为什么分类准确性这么高?——Why

# My Framework

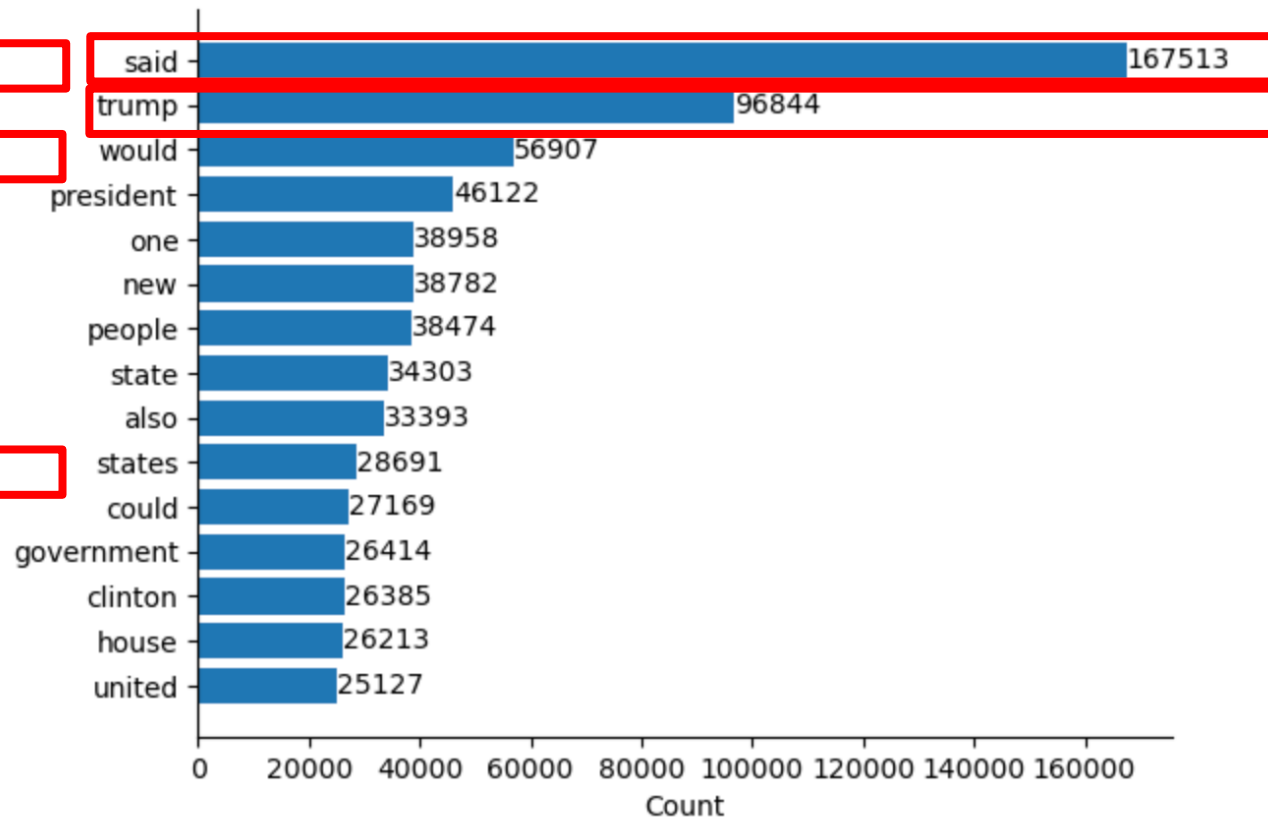


# 词频挖掘

Word in Fake News



Word in Real News



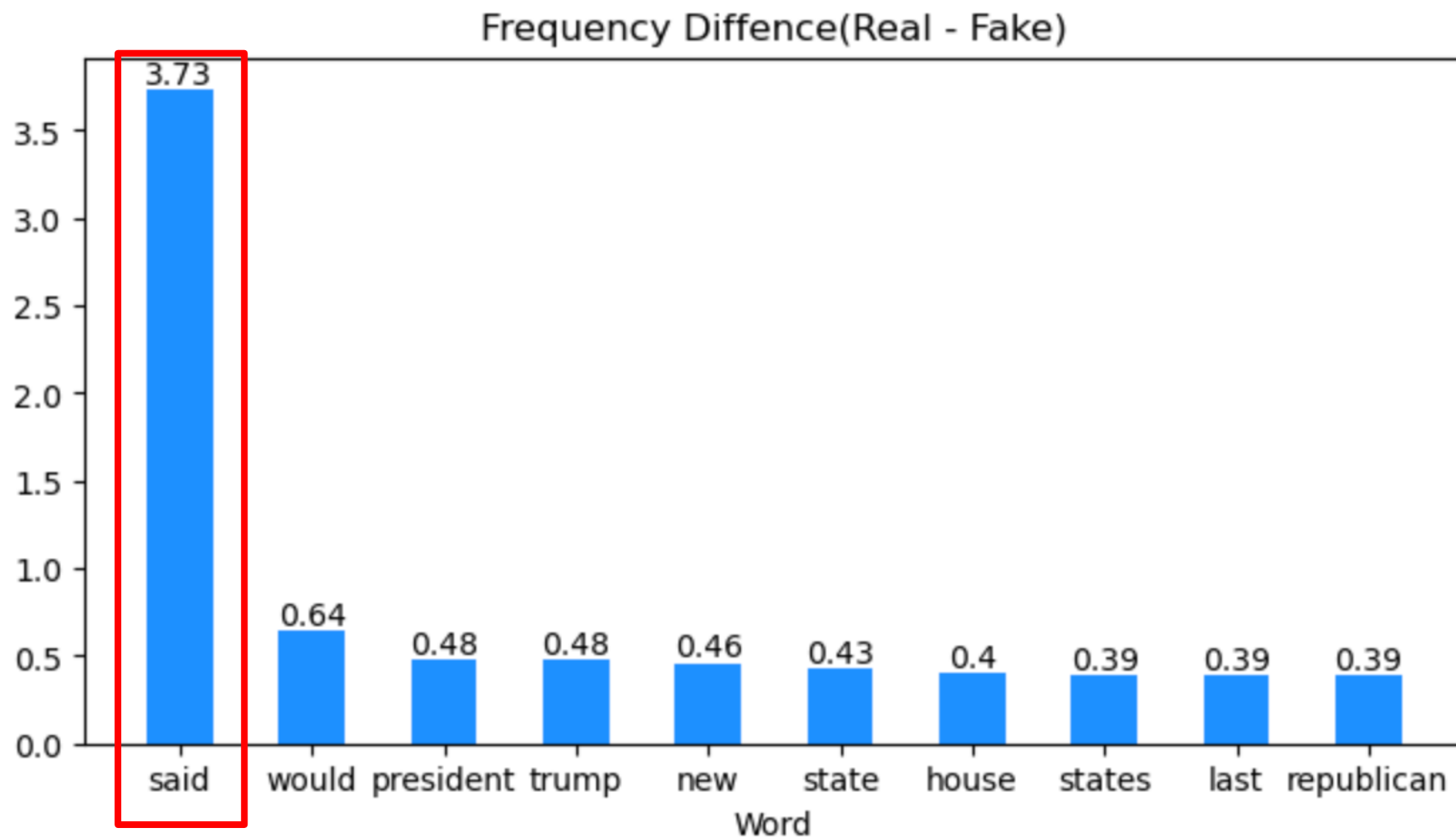
But...

真新闻：34806条      假新闻：43642条

比较词语出现频率差距可能更有意义

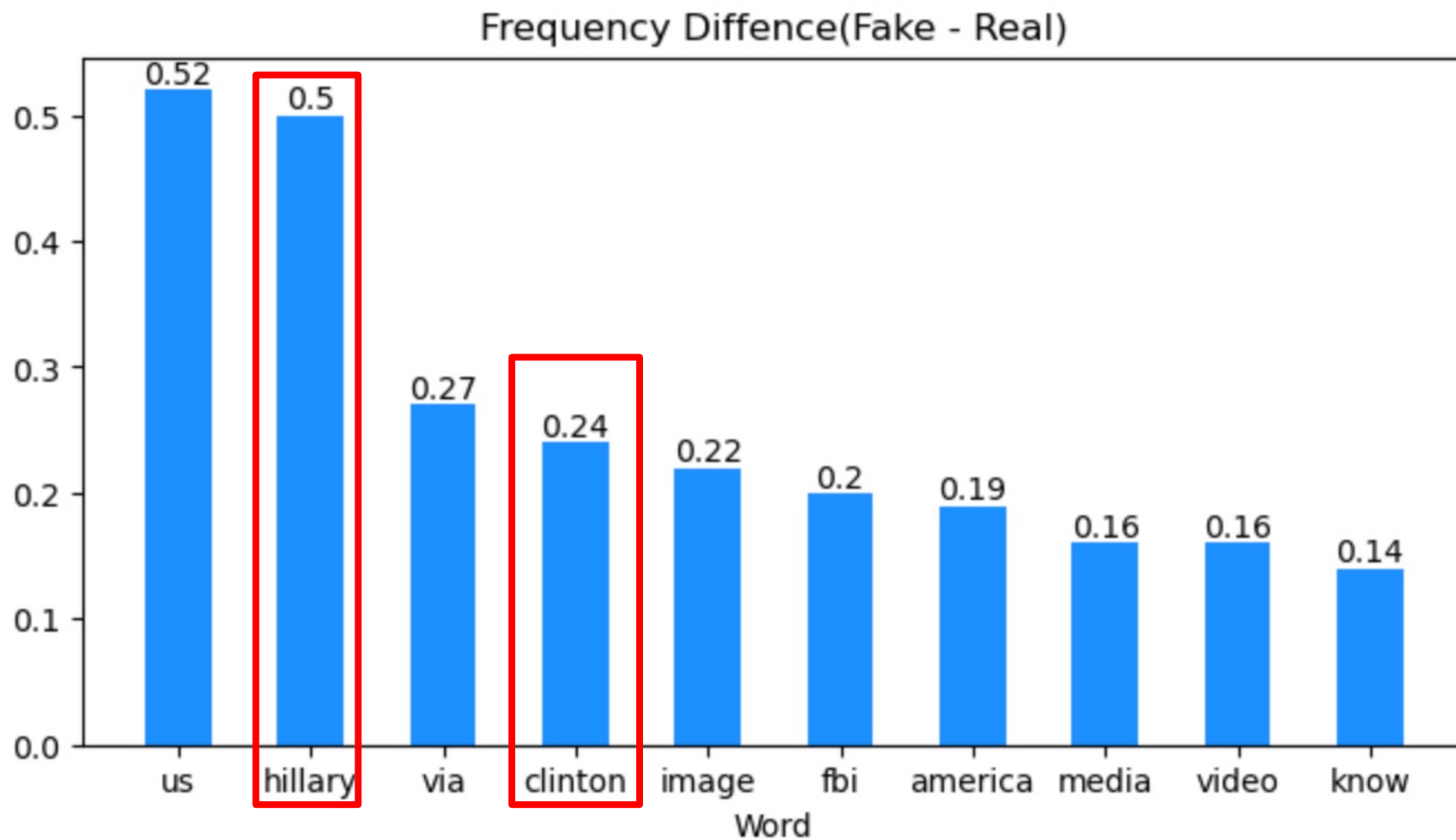
$$frequency = \frac{Word\_count}{len(category)}$$

# 词频挖掘

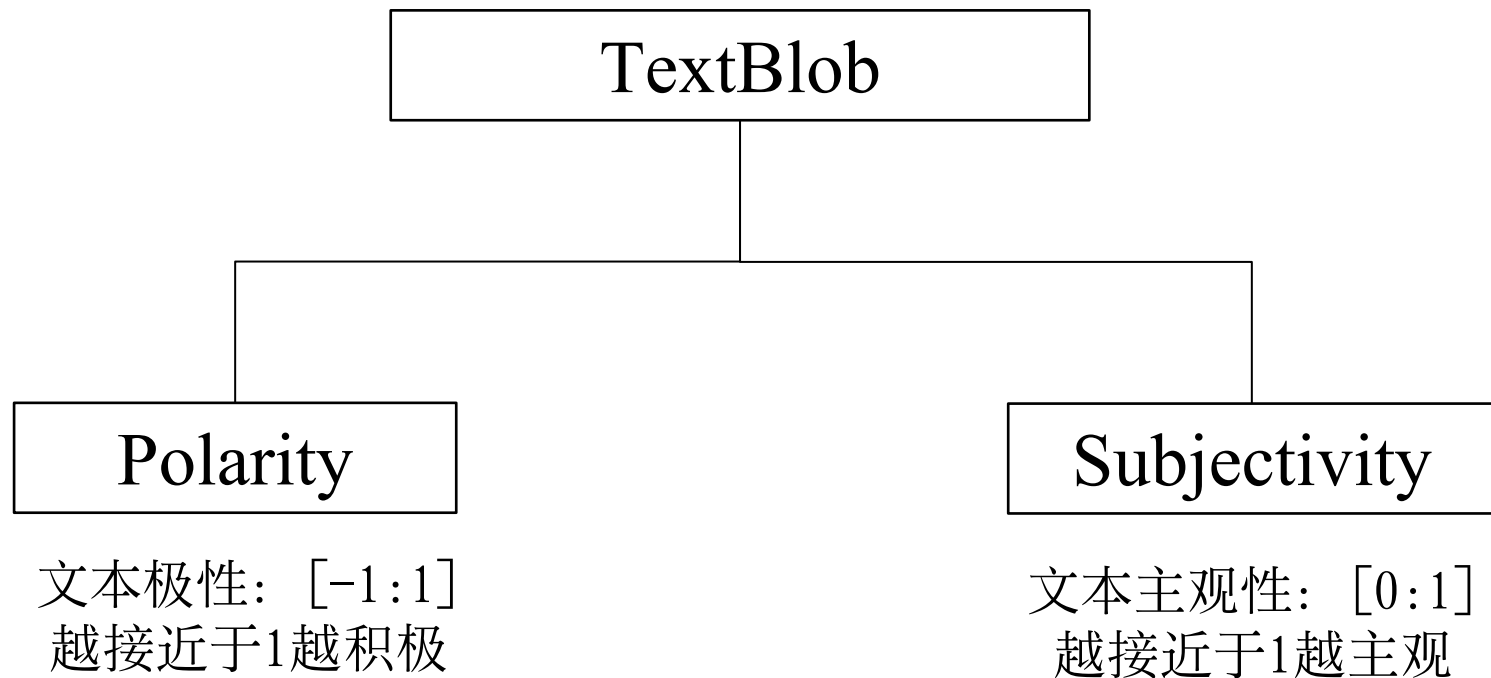




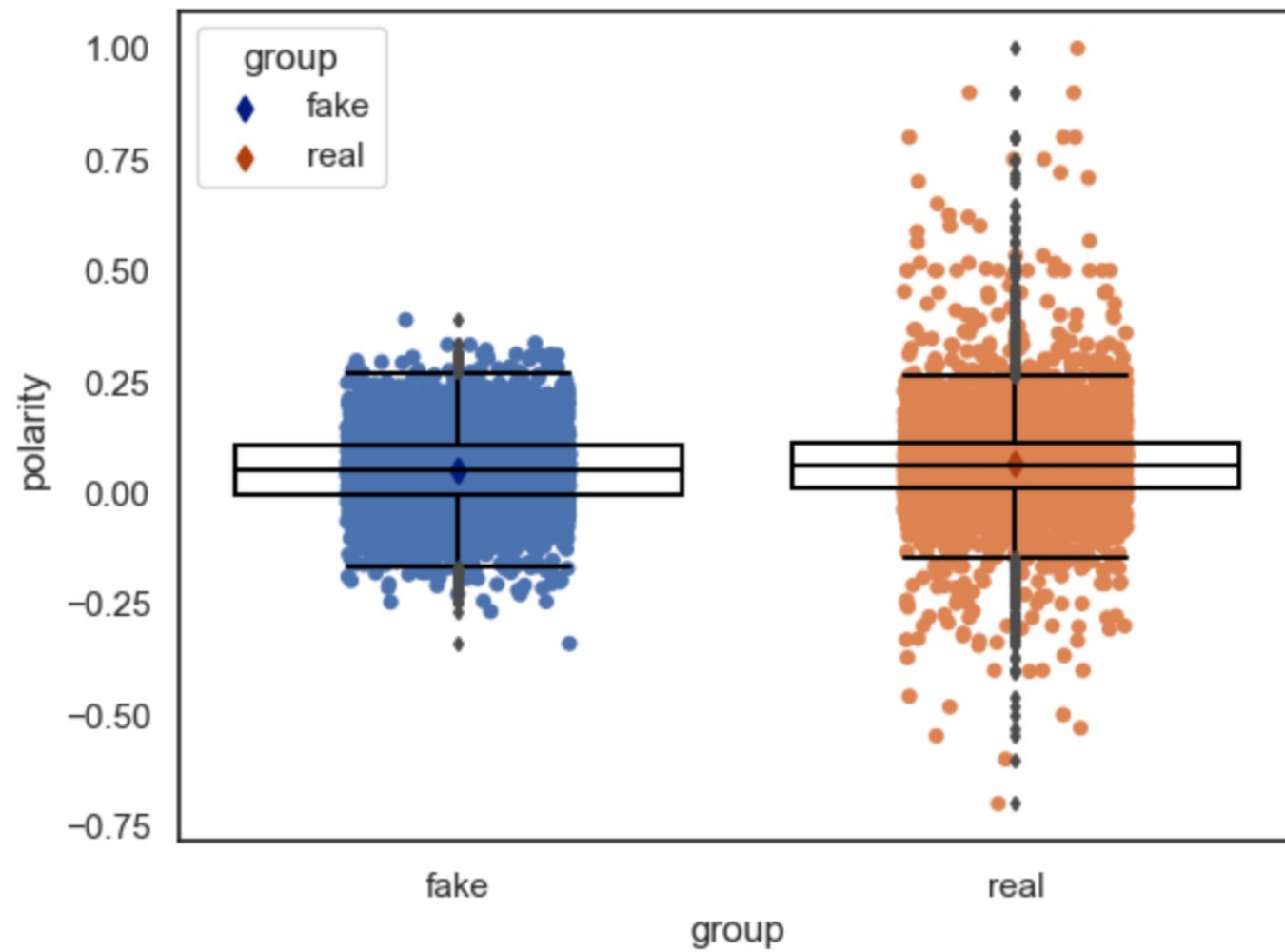
# 词频挖掘



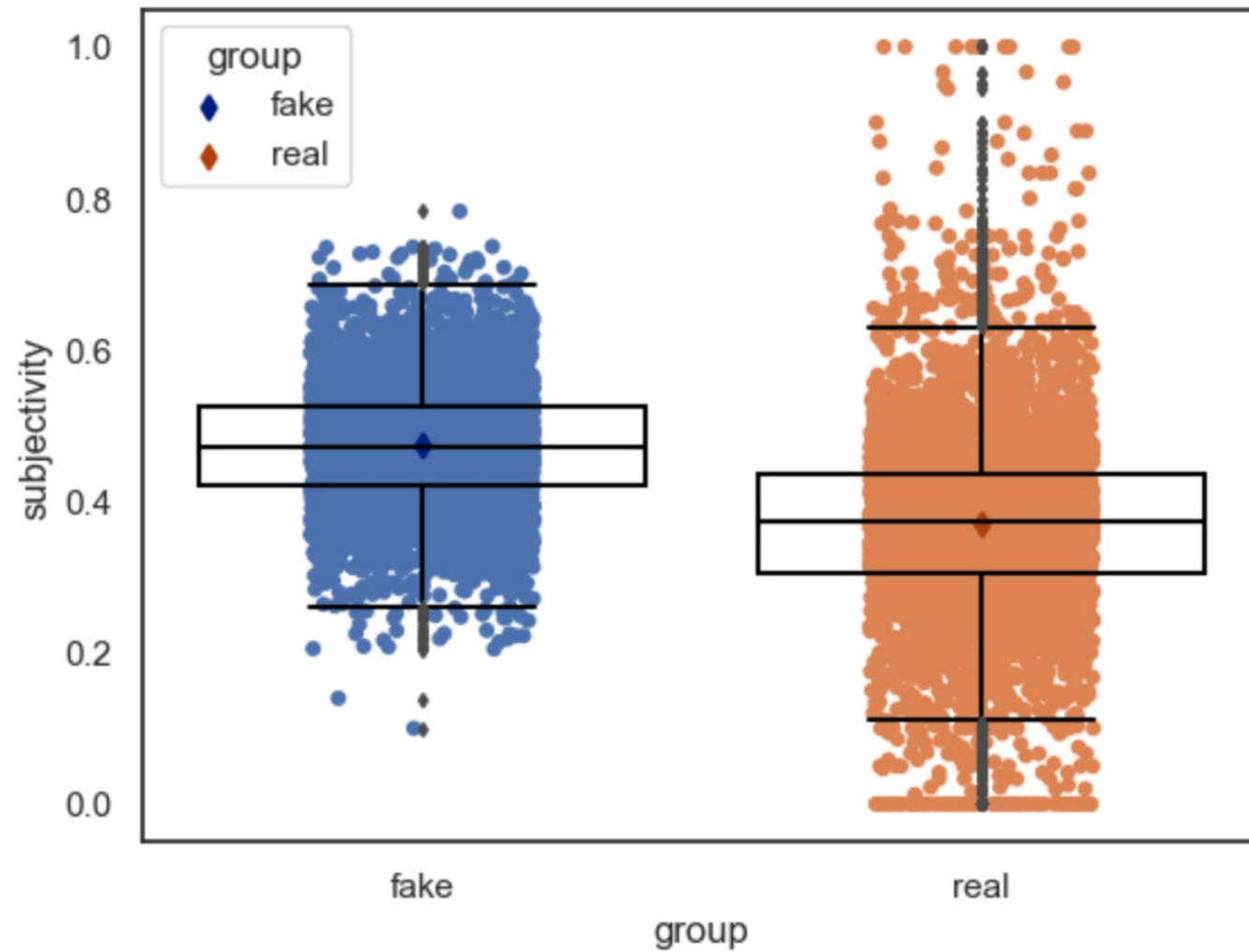
# 语义挖掘





# 语义挖掘——极性





# 语义挖掘——主观性





# 语义挖掘——主观性

 eb08c12 [TextBlob](#) / [textblob](#) / [en](#) / [en-sentiment.xml](#) 

 Go to file

 **sloria** Update English sentiment corpus 3079699 · 9 years ago

Code Blame 2932 lines (2932 loc) · 528 KB

Raw  

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--
3 SUBJECTIVITY LEXICON FOR ENGLISH ADJECTIVES.
4 Adjectives have a polarity (negative/positive, -1.0 to +1.0) and a subjectivity (objective/subjective, +0.0 to +1.0).
5 The reliability specifies if an adjective was hand-tagged (1.0) or inferred (0.7).
6 Words are tagged per sense, e.g., ridiculous (pitiful) = negative, ridiculous (humorous) = positive.
7 The Cornetto id (lexical unit id) and Cornetto synset id refer to the Cornetto lexical database for Dutch.
8 The WordNet id refers to the WordNet3 lexical database for English.
9 The part-of-speech tags (pos) use the Penn Treebank || tag set: NN = noun, JJ = adjective, ...
10 For English movie reviews (Pang & Lee polarity dataset v2.0), the accuracy is 75% (P 0.76, R 0.75, F1 0.75).
11 -->
12 <sentiment language="en" version="1.3" author="Tom De Smedt, Walter Daelemans" license="PDDL">
13 <word form="13th" wordnet_id="a-02203763" pos="JJ" sense="coming next after the twelfth in position" polarity="0.0" subjectivity="0.0" intensity="1.0" reliability="1.0">
14 <word form="20th" cornetto_synset_id="n_a-531612" wordnet_id="a-02204716" pos="JJ" sense="coming next after the nineteenth in position" polarity="0.0" subjectivity="0.0" intensity="1.0" reliability="1.0">
15 <word form="21st" wordnet_id="a-02204823" pos="JJ" sense="coming next after the twentieth in position" polarity="0.0" subjectivity="0.0" intensity="1.0" reliability="1.0">
16 <word form="2nd" wordnet_id="a-02202146" pos="JJ" sense="coming next after the first in position in space or time or degree or magnitude" polarity="0.0" subjectivity="0.0" intensity="1.0" reliability="1.0">
17 <word form="3rd" cornetto_synset_id="n_a-530634" wordnet_id="a-02202307" pos="JJ" sense="coming next after the second and just before the third" polarity="0.0" subjectivity="0.0" intensity="1.0" reliability="1.0">
18 <word form="abhorrent" wordnet_id="a-1625063" pos="JJ" sense="offensive to the mind" polarity="-0.7" subjectivity="0.8" intensity="1.0" reliability="1.0">
19 <word form="able" cornetto_synset_id="n_a-534450" wordnet_id="a-01017439" pos="JJ" sense="having a strong healthy body" polarity="0.5" subjectivity="0.0" intensity="1.0" reliability="1.0">
20 <word form="able" wordnet_id="a-00001740" pos="JJ" sense="(usually followed by 'to') having the necessary means or skill or know-how or authority" polarity="0.5" subjectivity="0.0" intensity="1.0" reliability="1.0">
```

# Summary

1. 无论是常用的特征提取+分类模型还是Bert都能够很好地分类。
2. 两类新闻在文本上有一定的特征:
  - (1) 假新闻的主观性更强, 可能是由于编写中杜撰成分更多
  - (2) 真新闻的“Said”词汇明显更多, 可能由于真新闻更敢于引用
  - (3) 假新闻的“Hilary”词汇明显更多, 可能由于数据集所属时间问题

## 后续方向

1. 基于TextBlob词表，进一步挖掘主观的词汇（假新闻特征）。
2. 试着将可视化做的更美观（比如左右柱状图）。
3. 将Bert模型在全样本上进行预测检验。

# 感谢聆听

## 敬请批评指正

赵煜东

12月22日