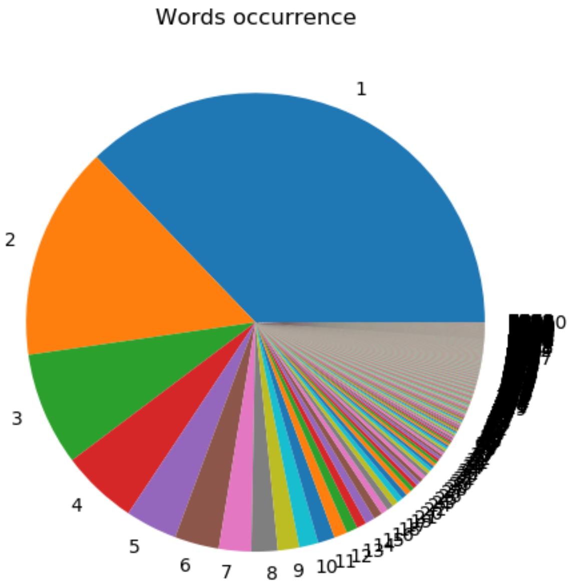
1. Exploratory data analysis:

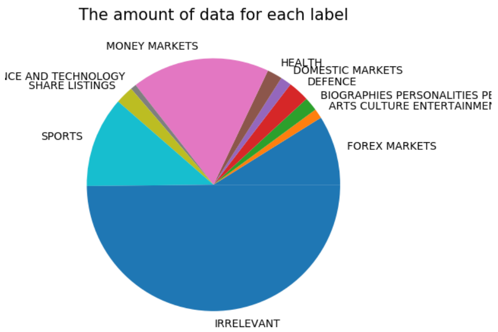
a. Words occurrence



Words occurrence	
count	857.000000
mean	41.738623
std	509.090486
min	1.000000
25%	1.000000
50%	2.000000
75%	4.000000
max	13300.000000

According to the training data, more than 50% of words only occurrence no more than twice. But the pre-train data contains most of the words, I think we should keep than rather than delete.

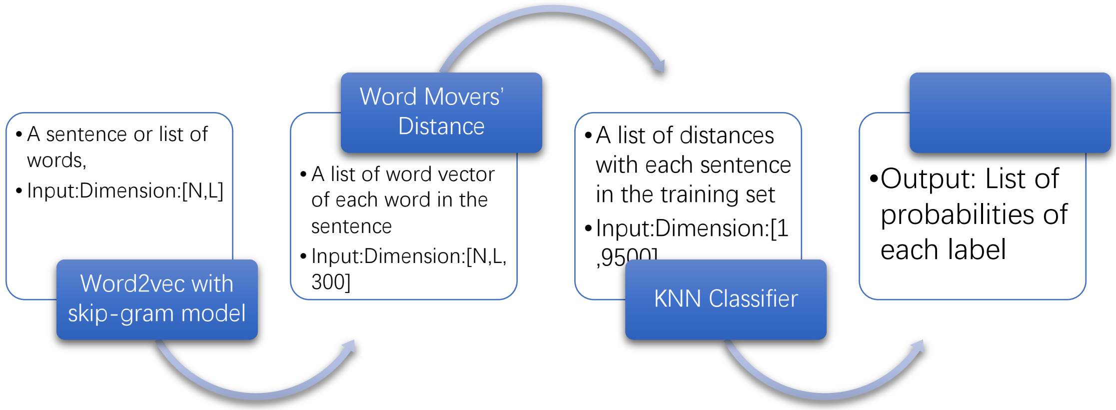
b. The amount of data for each label



The amount of data for each label	
FOREX MARKETS	845
ARTS CULTURE ENTERTAINMENT	117
BIOGRAPHIES PERSONALITIES PEOPLE	167
DEFENCE	258
DOMESTIC MARKETS	133
HEALTH	183
MONEY MARKETS	1673
SCIENCE AND TECHNOLOGY	70
SHARE LISTINGS	218
SPORTS	1102
IRRELEVANT	4734

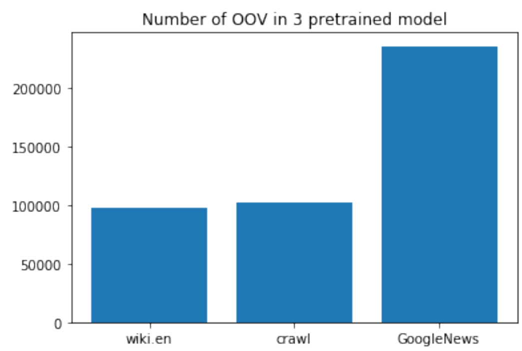
2.1 word2vec + WMDistance + KNN

a. Introduction of word2vec + WMDistance + KNN method in this project



b. Word2vec

Because the training data are not complete sentences, pre-trained word2vec data should work well in this project because it contains context relationship in it. I choose the vector data trained by Fasttext with skip-gram model[1] (which can be download here: https://fasttext.cc/docs/en/pretrained-vectors.html , English-text: wiki.en.vec) which has been trained on Wikipedia and the vector in dimension 300, and I compare 3 models, this one can cover most training-set words.

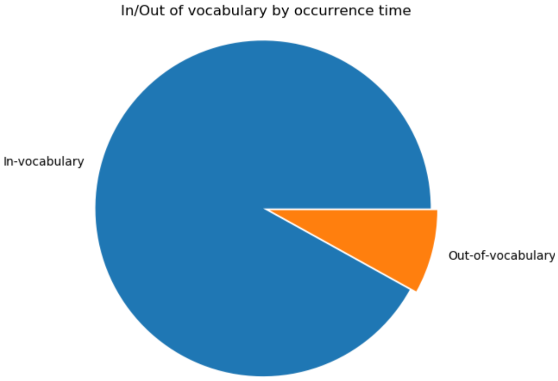


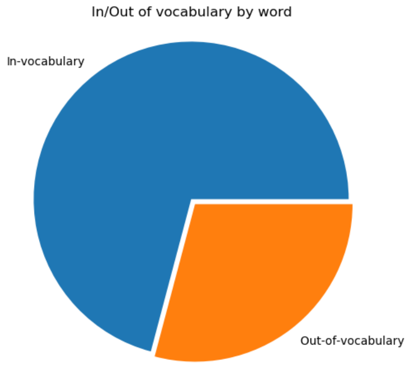
Number of OVV in 3 model (count by occurrence time)	
Wiki.en	97172
Crawl	101901
GoogleNews	235486

In word2vec, semantically similar words are very close to each other in spatial coordinates, while semantically unrelated words are far apart. This property can be used for a more generalized analysis of words and sentences.

1

By checking the words in, I found there are too many OVVs, so I just delete them in the training set, because these words can not improve the model precision and it will cause useless dimensions.

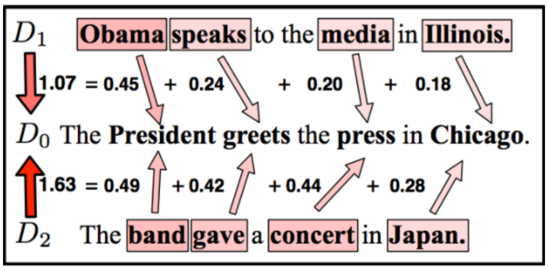




In/Out of vocabulary			
In-vocabulary by occurrence time	1110632	In-vocabulary by word	25352
Out-of-vocabulary by occurrence time	97172	Out-of-vocabulary by word	10418
Total occurrence time	1207804	Total word	35770

c. Word Mover's Distance:

Word Mover Distance is a similar text similarity method proposed in 2015[2].



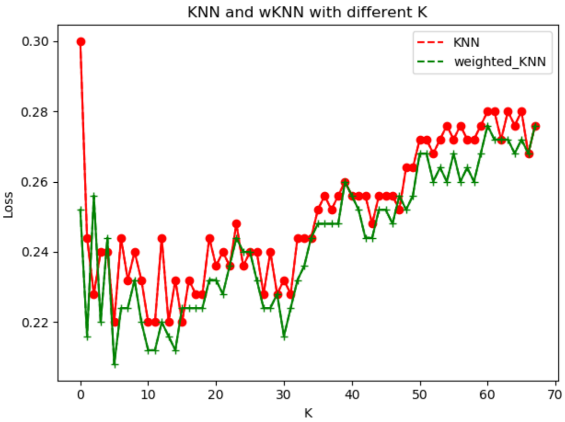
The main idea of WMD is to calculate the minimum global travel cost. For example, D_1, D_2 in the left picture, all words in D_1 travel to all words in D_2. For each word in D_1, it has a similar meaning to the words in D_2, so they can all move or move more distance (weight value); the semantic difference is greater , The moving distance is less or not moving. Multiplying the word vector distance by the moving distance is the travel cost of the two words.

d. KNN Classifier

After calculated all WMDistances between query and training data, we can the closest K training data. There are two hyper-parameter in KNN method : k and weight(Gaussian Kernel).

2

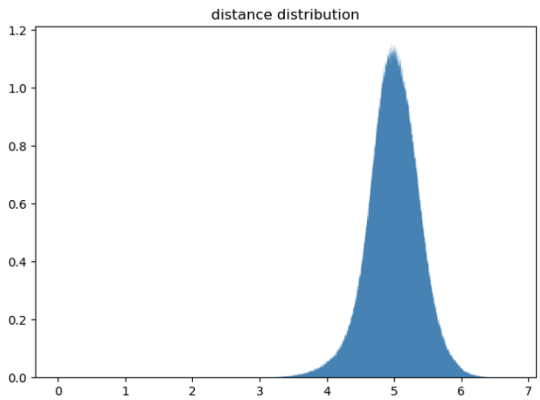
By using cross-validation, we can get Loss – K graph as below:



According to the graph, we can see that weighted KNN perform better than KNN without weights, and the chooses of k should between 3 to 15.

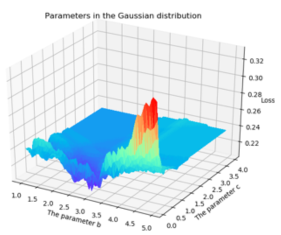
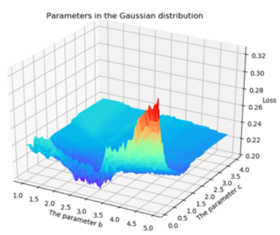
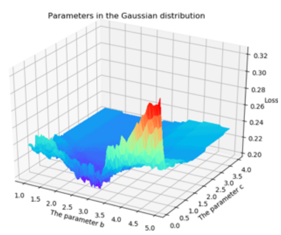
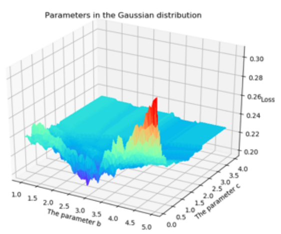
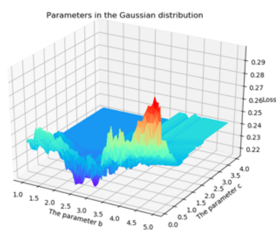
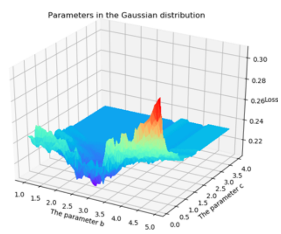
After that, we should determine the Gaussian Kernel parameters: a, b, c

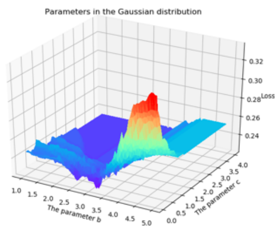
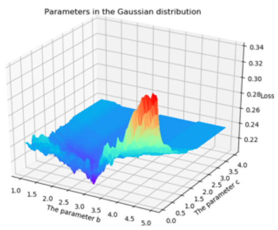
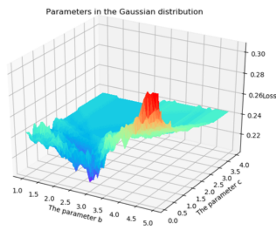
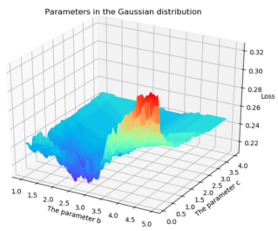
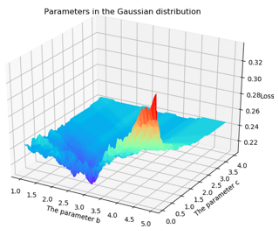
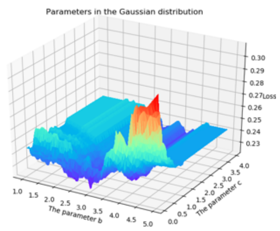
$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}, we\ assume\ that\ a = 1.$$



According to the WMDistance distribution between case in the train set, we should choose b between 1 and 4 and choose c between 0.5 and 2.5.

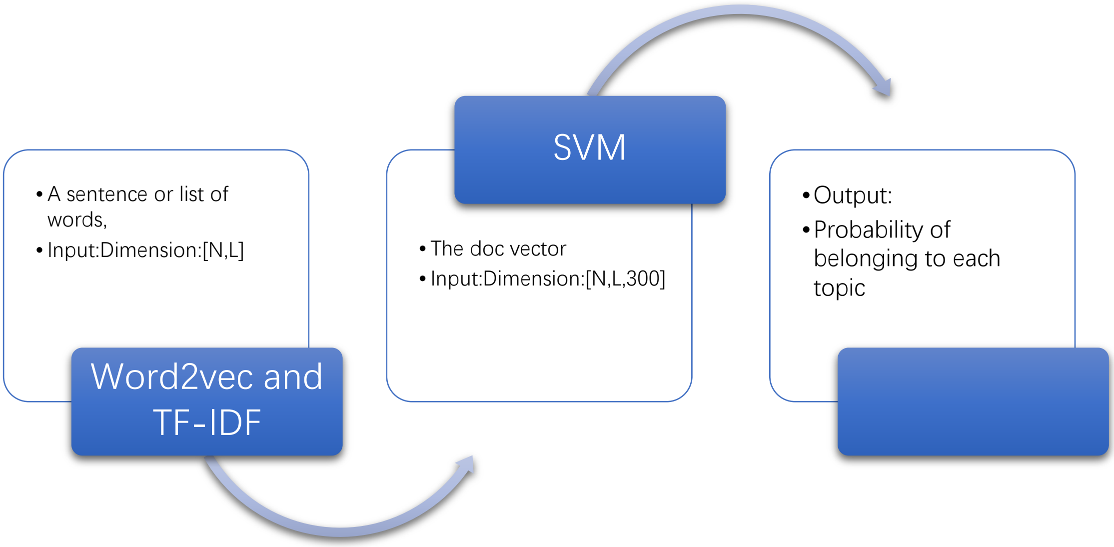
By input these parameters into KNN model, we can get loss rate like these in cross-validation:

	Fold_1	Fold_2	Fold_3
K=8			
K=9			

K=10			
K=11			

According to the cross-validation, we choose that $K = 10, b = 2.75, c = 0.66$.

2.2 word2vec + TF-IDF+SVM



a. Word2vec and TF-IDF

TF-IDF[3] (Term Frequency-Invers Document Frequency) calculates the importance of a word in the entire corpus based on the number of times the word appears in the text and the frequency of the document that appears in the entire corpus.

$$TFIDF_{i,j} = TF_{i,j} * IDF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} * log\frac{|D|}{1+|D_{t_i}|}$$

$n_{i,j}$ is the number of times feature word t_i appears in text d_j, $\sum_k n_{k,j}$ is the number of all feature words in the text d_j, $TF_{i,j}$ is the word frequency for a characteristic word.

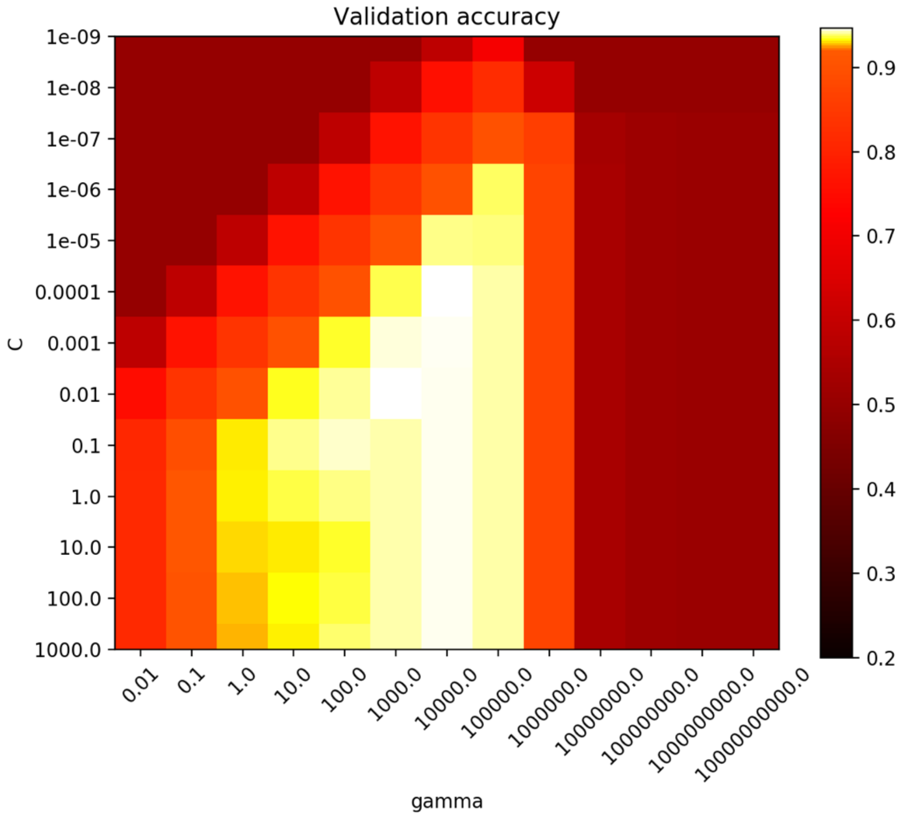
$|D|$ is the total number of texts in the corpus, $|D_{t_i}|$ indicating the number of feature words t_i in the text.

We can count each doc vector with pre trained word2vec model and TF-IDF as weights.

b. SVM

After calculated doc vector, each train or test case can be represent by 300-dim vector. The SVM theory has been learned on lecture and will not be repeated here. And two (c and gamma) parameters were tested using GCV.

3

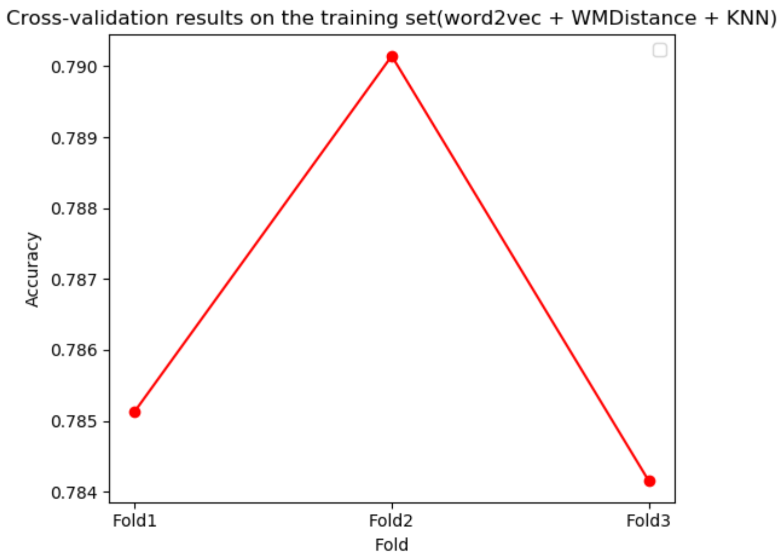


According to the result, we choose gamma = 1000 and c = 0.001, because the parameters (10000, 0.0001) and (1000, 0.1) are overfitting.

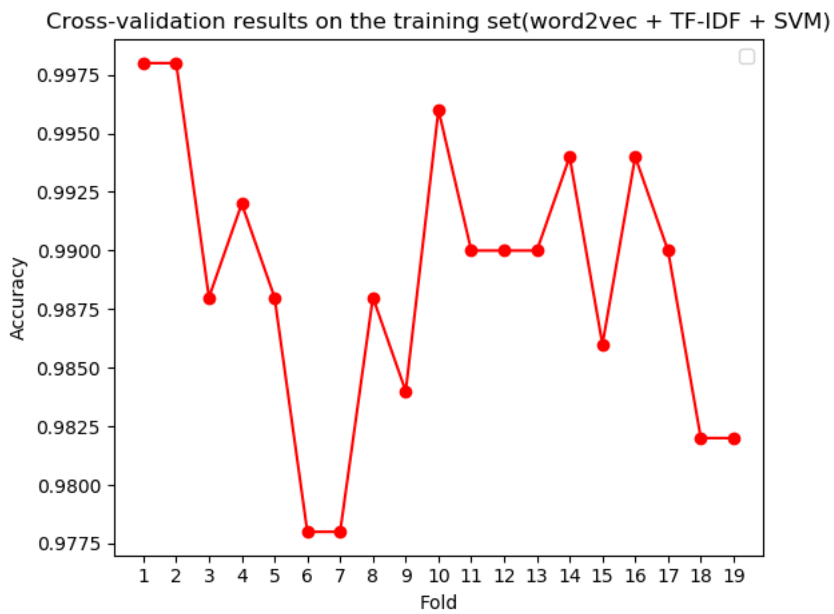
2. Results

a. Cross-validation results

a.1 word2vec + WMDistance + KNN



a.2 word2vec + TF-IDF+SVM



b. Final results

Final result of word2vec + WMDistance + KNN			
Topic name	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	0.34	0.34	0.34

BIOGRAPHIES PERSONALITIES PEOPLE	0.67	0.67	0.67
DEFENCE	0.67	0.46	0.55
DOMESTIC MARKETS	1.0	0.5	0.67
FOREX MARKETS	0.58	0.56	0.57
HEALTH	0.82	0.64	0.72
MONEY MARKETS	0.55	0.78	0.65
SCIENCE AND TECHNOLOGY	0.33	0.33	0.33
SHARE LISTINGS	0.5	0.43	0.46
SPORTS	0.95	1.0	0.98

Final result of word2vec + TF-IDF+SVM			
Topic name	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	1.00	1.00	1.00
BIOGRAPHIES PERSONALITIES PEOPLE	1.00	0.94	0.97
DEFENCE	0.92	1.00	0.96
DOMESTIC MARKETS	1.00	1.00	1.00
FOREX MARKETS	0.83	0.82	0.82
HEALTH	0.93	1.00	0.96
MONEY MARKETS	0.88	0.90	0.89
SCIENCE AND TECHNOLOGY	0.67	1.00	0.80
SHARE LISTINGS	1.00	0.78	0.88
SPORTS	1.00	1.00	1.00

c. Final article recommendations

Final result of word2vec + TF-IDF+SVM				
Topic name	**Suggested articles**	**Precision**	**Recall**	**F1**
ARTS CULTURE ENTERTAINMENT	9952 9703 9834	1.00	1.00	1.00
BIOGRAPHIES PERSONALITIES PEOPLE	9940 9758 9854 9878 9983 9768 9581 9988 9645 9526	1.00	0.94	0.97
DEFENCE	9559 9770 9987 9773 9616 9576 9706 9842 9670 9713	0.92	1.00	0.96
DOMESTIC MARKETS	9994 9796	1.00	1.00	1.00
FOREX MARKETS	9961 9965 9718 9725 9530 9588 9975 9551 9977 9837	0.83	0.82	0.82
HEALTH	9873 9661 9947 9810 9887 9621 9807 9735 9833 9978	0.93	1.00	0.96
MONEY MARKETS	9618 9516 9769 9998 9820 9828 9835 9691 9967 9860	0.88	0.90	0.89

SCIENCE AND TECHNOLOGY	9722 9929	0.67	1.00	0.80
SHARE LISTINGS	9601 9518 9562 9999 9972 9654 9667 9867 9666	1.00	0.78	0.88
SPORTS	9981 9573 9580 9657 9569 9541 9920 9663 9738 9574	1.00	1.00	1.00

3. Discussion:

If you continue this project, we can use LSTM (RNN) or CNN instead of SVM, they may get better performance in this project.

4. Reference

1. *Piotr Bojanowski, Edouard Grave : Enriching Word Vectors with Subword Information*
2. *Matt J. Kusner, Yu Sun: From Word Embeddings To Document Distances:*
3. *Martineau J, Finin T. Delta TFIDF: An Improved Feature Space for Sentiment Analysis*