EC503 Project Report: Combination of KNN and K-Means for News Classification

Cagri Yoruk, Haoqi Gu, Zhenfei Yu, Shuwei Li

INTRODUCTION

Text classification is one of the learning sciences from text mining, which is an important research area in computer science. There are several algorithms used, one of which is K-Nearest Neighbors(KNN).

The traditional KNN classification has three limitations. First, high calculation complexity. It needs to calculate all the similarities between training samples. If the number of training samples is huge, it will need much more time to finish. Second, dependence on the training set. The classification is generated only with training data, so if there is a small change in training set, it will need recalculation. Third, there is no weight difference between samples. All the samples are treated equally without weights, which does not satisfy the actual situation that the samples usually distribute unevenly.

Several approaches are proposed to solve these limitations, like Weight Adjusted KNN and Improved KNN Classification using Genetic Algorithm. Both of them focused on improving KNN algorithm itself with each step.

In our project, we focus on combining KNN and K-Means to avoid these problems, which can improve accuracy and reduce computation complexity.

METHODOLOGY

The process of our project mainly contains two parts, data train and evaluation.

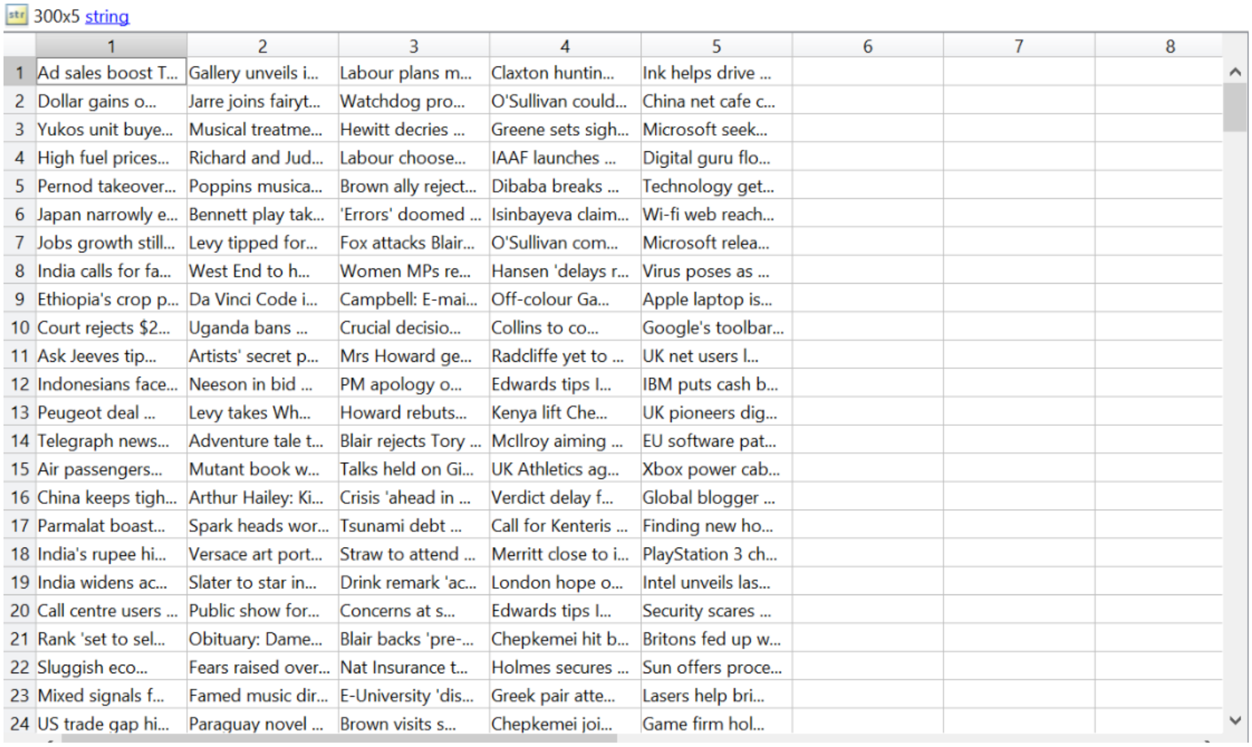
For data train, we first preprocess our dataset and get their weights as X and categorization as label. And then we apply k-fold cross validation on training data. For each fold, we use the K-Means algorithm to cluster and then apply KNN classification. And we find the optimal K, with which generates the highest training accuracy.

For evaluation, we compare the performance of applying KNN alone on test data with applying K-Means combined with KNN on test data.

DATA

This dataset is provided from BBC news. (D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006.).

Our dataset consists of 1500 news from 5 different categories. Business, entertainment, politics, sports and tech. We put all the news documents in a 300x5 matrix. Where each column represents a news category. The columns represent business, entertainment, politics, sports and tech categories respectively. We split the whole dataset by 8:2 ratio. That makes 1200 news for Train and 300 news for Test.



300x5 string

	1	2	3	4	5	6	7	8
1	Ad sales boost T...	Gallery unveils i...	Labour plans m...	Claxton huntin...	Ink helps drive ...			
2	Dollar gains o...	Jarre joins fairyt...	Watchdog pro...	O'Sullivan could...	China net cafe c...			
3	Yukos unit buye...	Musical treatme...	Hewitt decries ...	Greene sets sigh...	Microsoft seek...			
4	High fuel prices...	Richard and Jud...	Labour choose...	IAAF launches ...	Digital guru flo...			
5	Pernod takeover...	Poppins musica...	Brown ally reject...	Dibaba breaks ...	Technology get...			
6	Japan narrowly e...	Bennett play tak...	'Errors' doomed ...	Isinbayeva claim...	Wi-fi web reach...			
7	Jobs growth still...	Levy tipped for...	Fox attacks Blair...	O'Sullivan com...	Microsoft relea...			
8	India calls for fa...	West End to h...	Women MPs re...	Hansen 'delays r...	Virus poses as ...			
9	Ethiopia's crop p...	Da Vinci Code i...	Campbell: E-mai...	Off-colour Ga...	Apple laptop is...			
10	Court rejects $2...	Uganda bans ...	Crucial decisio...	Collins to co...	Google's toolbar...			
11	Ask Jeeves tip...	Artists' secret p...	Mrs Howard ge...	Radcliffe yet to ...	UK net users l...			
12	Indonesians face...	Neeson in bid ...	PM apology o...	Edwards tips l...	IBM puts cash b...			
13	Peugeot deal ...	Levy takes Wh...	Howard rebuts...	Kenya lift Che...	UK pioneers dig...			
14	Telegraph news...	Adventure tale t...	Blair rejects Tory ...	McIlroy aiming ...	EU software pat...			
15	Air passengers...	Mutant book w...	Talks held on Gi...	UK Athletics ag...	Xbox power cab...			
16	China keeps tigh...	Arthur Hailey: Ki...	Crisis 'ahead in ...	Verdict delay f...	Global blogger ...			
17	Parmalat boast...	Spark heads wor...	Tsunami debt ...	Call for Kenteris ...	Finding new ho...			
18	India's rupee hi...	Versace art port...	Straw to attend ...	Merritt close to i...	PlayStation 3 ch...			
19	India widens ac...	Slater to star in...	Drink remark 'ac...	London hope o...	Intel unveils las...			
20	Call centre users ...	Public show for...	Concerns at s...	Edwards tips l...	Security scares ...			
21	Rank 'set to sel...	Obituary: Dame...	Blair backs 'pre-...	Chepkemei hit b...	Britons fed up w...			
22	Sluggish eco...	Fears raised over...	Nat Insurance t...	Holmes secures ...	Sun offers proce...			
23	Mixed signals f...	Famed music dir...	E-University 'dis...	Greek pair atte...	Lasers help bri...			
24	US trade gap hi...	Paraguay novel ...	Brown visits s...	Chepkemei joi...	Game firm hol...			

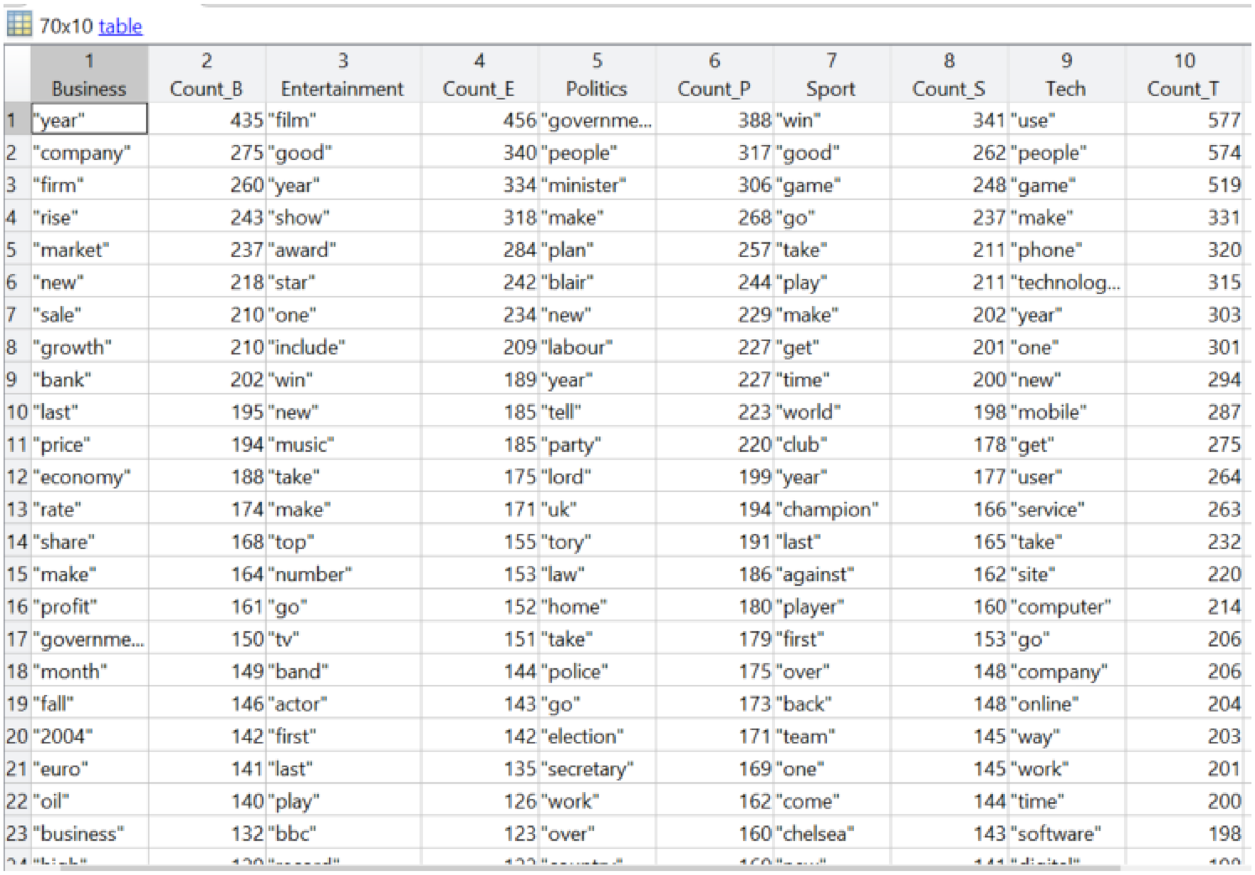
Figure 1. Dataset Matrix

DATA PREPROCESSING

In order to make use of raw text documents, we have to preprocess them. Our preprocessing steps are:

1. Case Folding: Changing letters to lowercases and getting rid of any special characters.
2. Tokenizing: The process of reducing the text into its individual.
3. Filtering: Dispose conjunction words and unnecessary words such as 'am','is','are', that doesn't give us any information about category labels.
4. Lemmatisation: This is a technique for reducing the words onto their root form.

Use a bag of words model to find the most frequent 70 words for each category, totaling 350 words. Then select each unique word to determine the feature words. The Word_list.mat file contains Most frequent words for each category that can be seen from the table:



70x10 table

	1 Business	2 Count_B	3 Entertainment	4 Count_E	5 Politics	6 Count_P	7 Sport	8 Count_S	9 Tech	10 Count_T
1	"year"	435	"film"	456	"governme...	388	"win"	341	"use"	577
2	"company"	275	"good"	340	"people"	317	"good"	262	"people"	574
3	"firm"	260	"year"	334	"minister"	306	"game"	248	"game"	519
4	"rise"	243	"show"	318	"make"	268	"go"	237	"make"	331
5	"market"	237	"award"	284	"plan"	257	"take"	211	"phone"	320
6	"new"	218	"star"	242	"blair"	244	"play"	211	"technolog...	315
7	"sale"	210	"one"	234	"new"	229	"make"	202	"year"	303
8	"growth"	210	"include"	209	"labour"	227	"get"	201	"one"	301
9	"bank"	202	"win"	189	"year"	227	"time"	200	"new"	294
10	"last"	195	"new"	185	"tell"	223	"world"	198	"mobile"	287
11	"price"	194	"music"	185	"party"	220	"club"	178	"get"	275
12	"economy"	188	"take"	175	"lord"	199	"year"	177	"user"	264
13	"rate"	174	"make"	171	"uk"	194	"champion"	166	"service"	263
14	"share"	168	"top"	155	"tory"	191	"last"	165	"take"	232
15	"make"	164	"number"	153	"law"	186	"against"	162	"site"	220
16	"profit"	161	"go"	152	"home"	180	"player"	160	"computer"	214
17	"governme...	150	"tv"	151	"take"	179	"first"	153	"go"	206
18	"month"	149	"band"	144	"police"	175	"over"	148	"company"	206
19	"fall"	146	"actor"	143	"go"	173	"back"	148	"online"	204
20	"2004"	142	"first"	142	"election"	171	"team"	145	"way"	203
21	"euro"	141	"last"	135	"secretary"	169	"one"	145	"work"	201
22	"oil"	140	"play"	126	"work"	162	"come"	144	"time"	200
23	"business"	132	"bbc"	123	"over"	160	"chelsea"	143	"software"	198
24	"high"	[illegible]	"record"	[illegible]	"country"	[illegible]	"new"	[illegible]	"digital"	[illegible]

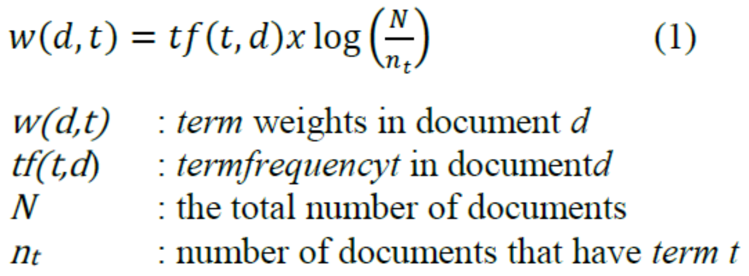
Figure 2. Word Count Table

Term-Weighting

After completing the text preprocessing, we re-weight all the news using a method called TF-IDF.

- TF: Term weighting based on the word frequency that appears in a document.
- IDF: Weighting method based on number of words that appear throughout all the documents.

You can understand the intuition behind the method with the formula given down below.



$$w(d,t) = tf(t,d)x \log\left(\frac{N}{n_t}\right) \qquad (1)$$

$w(d,t)$: *term* weights in document *d*
$tf(t,d)$: *termfrequencyt* in document*d*
N : the total number of documents
n_t : number of documents that have *term t*

Figure 3. TF-IDF Equation

After re-weighting the documents we get four matrices that are crucial to our ML model.

- Xtr: Train set that contains weights of 1200 news / 1200x228 Matrix
- Xte: Test set that contains weights of 300 news / 300x228 Matrix
- Ytr: Train labels of the weights that corresponds to the news category / 1200x1 Matrix
- Yte: Test labels of the weights that corresponds to the news category / 300x1 Matrix

COMBINATION OF KNN AND K-MEANS CLUSTERING

To find the optimal k, which is the number of nearest point calculated in KNN, we do the following steps with k equals from one to nine:

- We first use k-fold cross validation to split the data set.
- With each fold, apply K-Means to cluster the training set into K clusters and obtain K centroids, and we set the labels of centroids based on major labels.
- Run the KNN algorithm on test set with centroids and evaluate predicted labels with true labels.
- Output the mean accuracy of KNN errors with each k.
- Put the nine mean accuracy into a vector and plot k with corresponding accuracy.

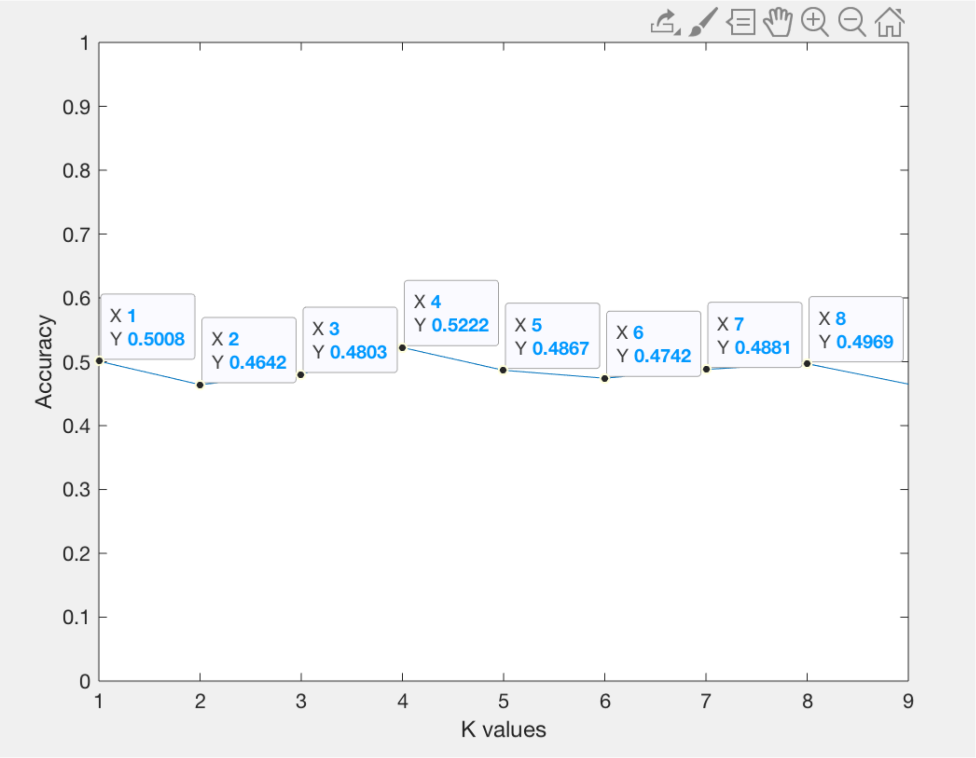


Figure 4. Accuracy-k values plot

With the plot, we can tell that when k=4, it reaches highest accuracy, 0.5222.

EXPERIMENTS AND EVALUATIONS

To evaluate text classifiers, it must consider both classification accuracy rate and recall rate. Its measurements using the F-measure with matrix confusion standard are shown in the following table.

Table 1. Matrix Confusion

Cluster by system	Cluster is actually	
	Yes	Not
Yes	*a*	*b*
Not	*c*	*d*

And Recall is a measure of the number of documents that are successfully classified of all documents that should be correct, it is calculated by:

$$recall = \frac{a}{a+c}$$

Precision is a level of accuracy from clustering results. That is, how many percent all the result cluster documents is declared properly:

$$precision = \frac{a}{a+b}$$

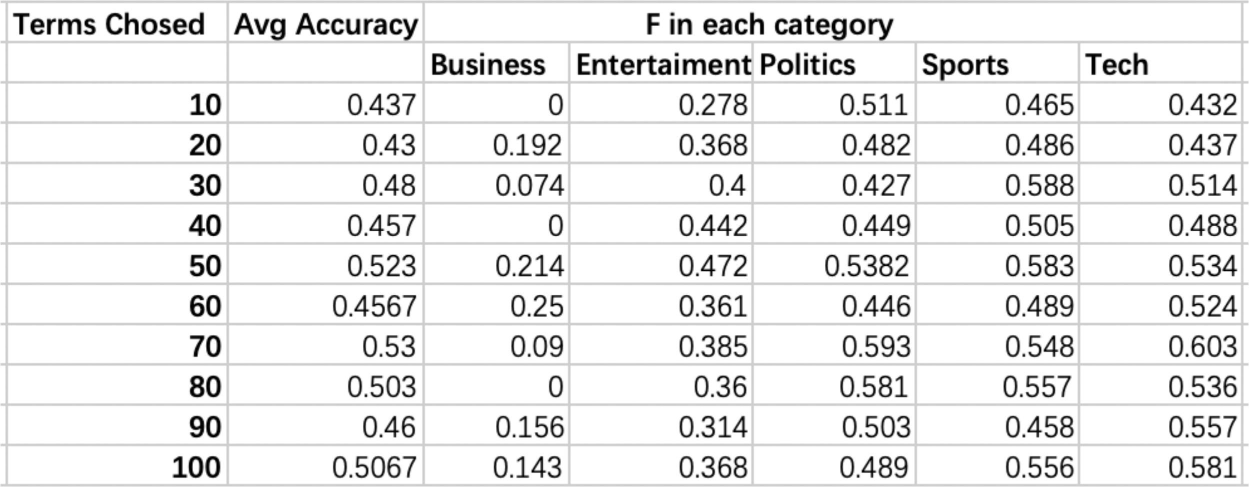
F measure is a measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Experiment 1:

The first experiment is to find the optimal word count for reweighting. We tried to compare some potential term number for all news and record the f value to see which has the best result. The table below shows the relation between the number of terms we chose and the f value of each category. The results were obtained while choosing 150 clusters in K-means and choosing k=1 in KNN algorithm.

Table 2. Accuracy with Different number of terms



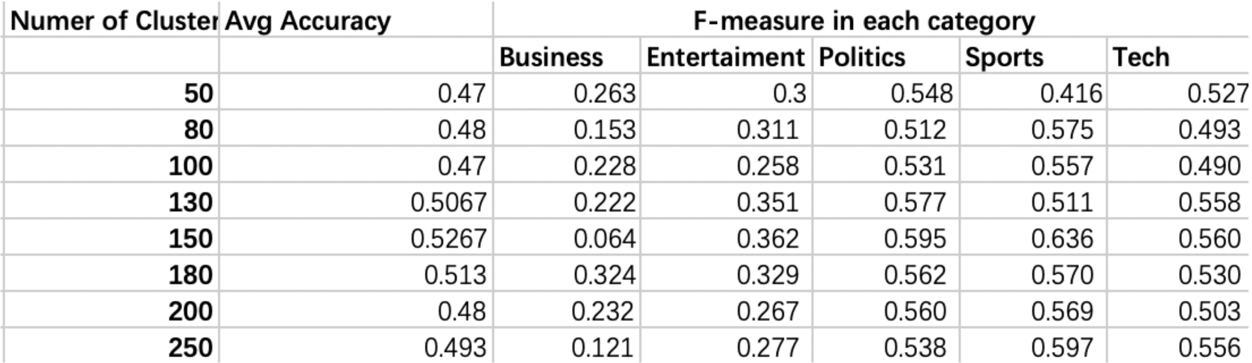
Terms Chosed	Avg Accuracy	F in each category				
		Business	Entertaiment	Politics	Sports	Tech
10	0.437	0	0.278	0.511	0.465	0.432
20	0.43	0.192	0.368	0.482	0.486	0.437
30	0.48	0.074	0.4	0.427	0.588	0.514
40	0.457	0	0.442	0.449	0.505	0.488
50	0.523	0.214	0.472	0.5382	0.583	0.534
60	0.4567	0.25	0.361	0.446	0.489	0.524
70	0.53	0.09	0.385	0.593	0.548	0.603
80	0.503	0	0.36	0.581	0.557	0.536
90	0.46	0.156	0.314	0.503	0.458	0.557
100	0.5067	0.143	0.368	0.489	0.556	0.581

It shows that there is a tradeoff between the time complexity and accuracy. Since if we increase the accuracy, one way we can do is to enlarge the group of "frequent words", but that will lead to the increase of complexity of our train data, which further expands the time consumed to classify. To choose a comparative reasonable point, we need to consider the balance, as the result, we chose 70 as our term number for each category.

Experiment 2:

The second experiment is to find the optimal clusters k in K-means. We tried to compare some potential k value for all news and record the f value to see which has the best result. The table below shows the relation between the value of k we chose and the f value of each category.

Table 3. Accuracy with different number of clusters



Numer of Cluster	Avg Accuracy	F-measure in each category				
		Business	Entertaiment	Politics	Sports	Tech
50	0.47	0.263	0.3	0.548	0.416	0.527
80	0.48	0.153	0.311	0.512	0.575	0.493
100	0.47	0.228	0.258	0.531	0.557	0.490
130	0.5067	0.222	0.351	0.577	0.511	0.558
150	0.5267	0.064	0.362	0.595	0.636	0.560
180	0.513	0.324	0.329	0.562	0.570	0.530
200	0.48	0.232	0.267	0.560	0.569	0.503
250	0.493	0.121	0.277	0.538	0.597	0.556

The table shows that when there are about 150 centers, we obtain the highest accuracy from our training data. Therefore, we set the number of clusters in the K-means algorithm to 150.

Evaluation:
From the tables stated above, we can conclude that our system classifies politics, sports and tech news better than it does on business and entertainment news. We think it is because the terms chosen from the former three types of news are more distinctive, and the terms appear in business and entertainment news are comparatively common in other kinds of news.

CONCLUSION AND FUTURE WORK

In this project, the combination of KNN and k-means clustering based on term re-weighting for classify English news is proposed. The main steps include date pre-processing, weighting term for documents, cluster with k-means algorithm, classification with KNN algorithm, and finally, the evaluation.

The results show that the optimal classification rate of news categories can be achieved to about 53% among 5 different types. It has almost the same accuracy as the traditional KNN does, but in classifying documents, a traditional KNN algorithm will require a longer execution time because it has a complex calculation.

For our future work, experiments can be performed by combining the traditional KNN with an algorithm that has a higher level of accuracy, and methods for determining the initial centroid in k-means algorithm needs to be more efficient.

ACKNOWLEDGMENTS

Our thanks goes to Dr. Francesco Orabona in the Department of Electrical & Computer Engineering, Boston University, who has helped us in our project.