ICL at NTCIR-7: An Improved KNN Algorithm for Text Categorization

Wei Wang, Sujian Li
Inst. of Computational Linguistics,
Peking University

Outline

- Introduction
- Algorithm Details
- Evaluations
- Conclusion and future work
- Acknowledgment & Contact

Introduction

- Traditional KNN algorithm's computation expense is too large since there are nearly 5 million "labeled nodes" in the NTCIR-7 PAJ data for the Patent Mining Task.
- To make the computing more efficient, we employ an improved KNN algorithm which makes trade-off between effectiveness and time complexity.
  - To come up with an improved algorithm which calculates the distance between the unlabeled node and "centroid" nodes instead of all nodes. The "centroid" node represents all nodes belonging to the same category.
  - To try two distance metrics in our algorithm: cosine similarity and Euclid distance. Evaluation results on NTCIR-7 test data show that the former one is slightly better.

- The benefits of our method is two-fold:
  - Computation expense is greatly reduced, and now we have to only calculates about 30, 000 distances for an unlabeled node. (The number of different IPC codes used in NTCIR-7)
  - Data skew problem can be effectively resolved, because there is only one centroid node for each category.

- However, whether the categorization can benefit from this method still has to be further studied, which will be our future research.

Algorithm Details

- Our system uses the PAJ (Patent Abstracts of Japan) of 1993-2002 as the labeled data for our KNN algorithm.

- We merge patent files with the same IPC code. The merged file is denoted with a term vector which we call centroid vector.

- We find 1000 nearest centroid vectors (most possible IPC codes) for each topic vector using two different distance metrics: the Euclid distance metric and the cosine similarity metric.

Detailed Steps

- Step 1. Extract patent file contents from raw data, compute the IDF for each word. In order to remove noise features we discard words appearing in less than 3 patents.

- Step 2. Merge patents with the same IPC code into one file. Count the merged files' term frequency and get the "centroid" vector representation for the each merged file.

- Step 3. Get the term vector representation for each test topic. Then calculate the distance or similarity between topic vector and each centroid vector. Choose 1,000 IPC codes with minimum distance or maximum similarity as the answers.

Two distance metrics

- Suppose the term vector for topic file and the ith merged file are V_i and V_{topic} respectively. The Euclid distance and cosine similarity between the two vectors are calculated as:

$$Euclid(V_i, V_{topic}) = |V_i - V_{topic}|$$

$$Cosine(V_i, V_{topic}) = \frac{V_i . V_{topic}}{|V_i| * |V_{topic}|}$$

- The Euclid distance metric prefers long merged files because they have more words. The cosine similarity metric can avoid such problem. Evaluation results prove this.

Evaluations

- We will compare the performance of our system with other participants. Each participant can submit at most 3 results. Of the 20 results submitted by nine participants, our best result ranks 12th.

- We will compare the performance of the two distance metrics: Euclid distance and cosine similarity. Evaluation results show that the cosine similarity metric is better on average.

Comparison of our system and other systems

- There are totally 2051 answers for 879 topics. Our system retrieves 1888 of them.

Table 1: Comparison of retrieved answer number

Participant	Retrieved IPC (Relevant 2051)
NEUN1_S1	1975
xrce_e2j2e	1932
KECIR	1892
ICL07_1	1888
nttcs2	1848
BRKLY-PM-EN-02	1488
AINLP04	1455
rali1	953
PI-5b	895

- The IPC numbers showed in Table 1 are the maximum numbers retrieved by each participant. The top 5 results (including ours) show no significant difference with each other. In fact, our system’s gap with other top systems lies in the precision indicators, as shown in Table 2 and Table 3.

Table 2: Comparison of average interpolated recall precision

Interpolated Value	ICL07_1	NEUN1_S1	xrce_e2j2e	KECIR
0.00	0.2118	0.5965	0.5318	0.3973
0.10	0.2118	0.5965	0.5318	0.3973
0.20	0.2068	0.5936	0.5302	0.3949
0.30	0.1922	0.5718	0.5075	0.3721
0.40	0.1613	0.5308	0.4658	0.3300
0.50	0.1587	0.5254	0.4555	0.3201
0.60	0.1142	0.4522	0.3821	0.2507
0.70	0.1021	0.4183	0.3536	0.2212
0.80	0.0980	0.4085	0.3469	0.2113
0.90	0.0962	0.4029	0.3424	0.2062
1.00	0.0961	0.4027	0.3424	0.2062

Table 3: Comparison of micro average interpolated recall precision

Interpolated Value	ICL07_1	NEUN1_S1	xrce_e2j2e	KECIR
0.00	0.1024	0.4664	0.4107	0.2708
0.10	0.0846	0.4664	0.4107	0.2708
0.20	0.0556	0.3874	0.3305	0.1862
0.30	0.0417	0.3874	0.2704	0.1486
0.40	0.0312	0.3201	0.2392	0.1090
0.50	0.0230	0.2353	0.1669	0.0744
0.60	0.0163	0.1770	0.1007	0.0468
0.70	0.0112	0.1097	0.0609	0.0252
0.80	0.0062	0.0519	0.0293	0.0124
0.90	0.0027	0.0149	0.0075	0.0038
1.00	0.0000	0.0000	0.0000	0.0000

- Our system’s low precision indicates that our rank function is not good enough. It’s partly caused by the model we use. Our KNN model is “flat”, that is, it treats all categories equally while the IPC has a hierarchical structure.

Results Analysis

- We haven’t done much work on feature selection and term weight tuning.
  - Have about 400,000 features, most of which are perhaps noise features. We believe that noise features greatly affect our precision.
  - Use IDF as term weights, which may not be appropriate. Position can also be used to tune feature weights. for example, words in patent titles are usually more informative than words in claims, so maybe we should assign such words higher weights.

Comparison of two distance metrics

- We have submitted two results, using cosine similarity and Euclid distance as the distance metric respectively. The former one retrieves 1888 answers, while the other retrieves 1277. The precision of cosine similarity is also higher than that of Euclid distance.

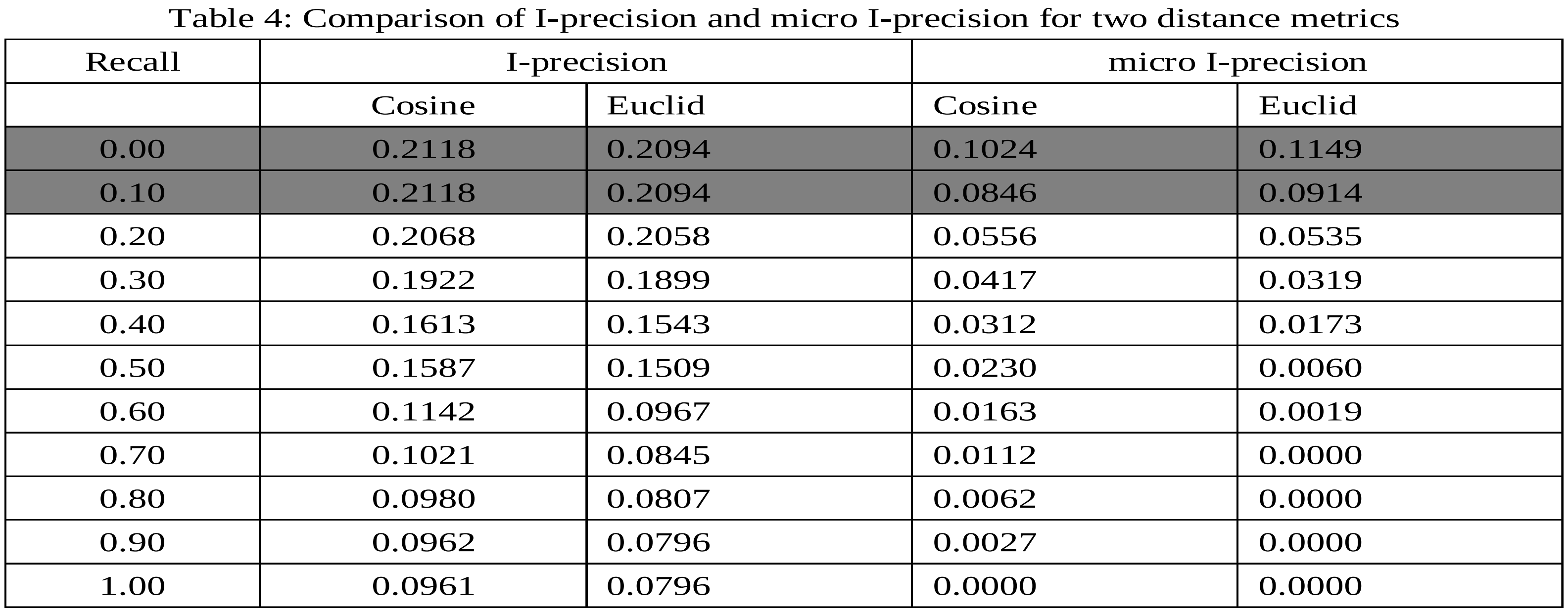

Table 4: Comparison of I-precision and micro I-precision for two distance metrics

Recall	I-precision		micro I-precision	
	Cosine	Euclid	Cosine	Euclid
0.00	0.2118	0.2094	0.1024	0.1149
0.10	0.2118	0.2094	0.0846	0.0914
0.20	0.2068	0.2058	0.0556	0.0535
0.30	0.1922	0.1899	0.0417	0.0319
0.40	0.1613	0.1543	0.0312	0.0173
0.50	0.1587	0.1509	0.0230	0.0060
0.60	0.1142	0.0967	0.0163	0.0019
0.70	0.1021	0.0845	0.0112	0.0000
0.80	0.0980	0.0807	0.0062	0.0000
0.90	0.0962	0.0796	0.0027	0.0000
1.00	0.0961	0.0796	0.0000	0.0000

Conclusion and future work

- Conclusion
  - Our current model is simple, effective, at the cost of information loss.
  - Negative features, and the hierarchical structure of IPC, are ignored by our system.
- Future work
  - Consider the hierarchical structure of this classification
  - Try feature selection methods, like IG, MI and LSI to remove noise features and redundant features.
  - Use a mixed model to improve the precision of our system, e.g., the combination of KNN and other machine learning methods. That is, we will first use the KNN model to extract top 1000 IPC codes and other models like ME or SVM to rescore each IPC code to get a more accurate ranking for them.

Acknowledgment

- This work is supported by NSFC programs (No: 60603093 and 60875042), and 973 National Basic Research Program of China (2004CB318102).

- For further information, please contact wwei@pku.edu.cn or lisujian@pku.edu.cn.