KNN and re-ranking models for English patent mining at NTCIR-7

Tong Xiao, Feifei Cao, Tianning Li, Guolong Song, Ke Zhou, Jingbo Zhu and Huizhen Wang

Natural Language Processing Lab,

Northeastern University (P. R. China)

xiaotong@mail.neu.edu.cn

Outline

- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

Outline

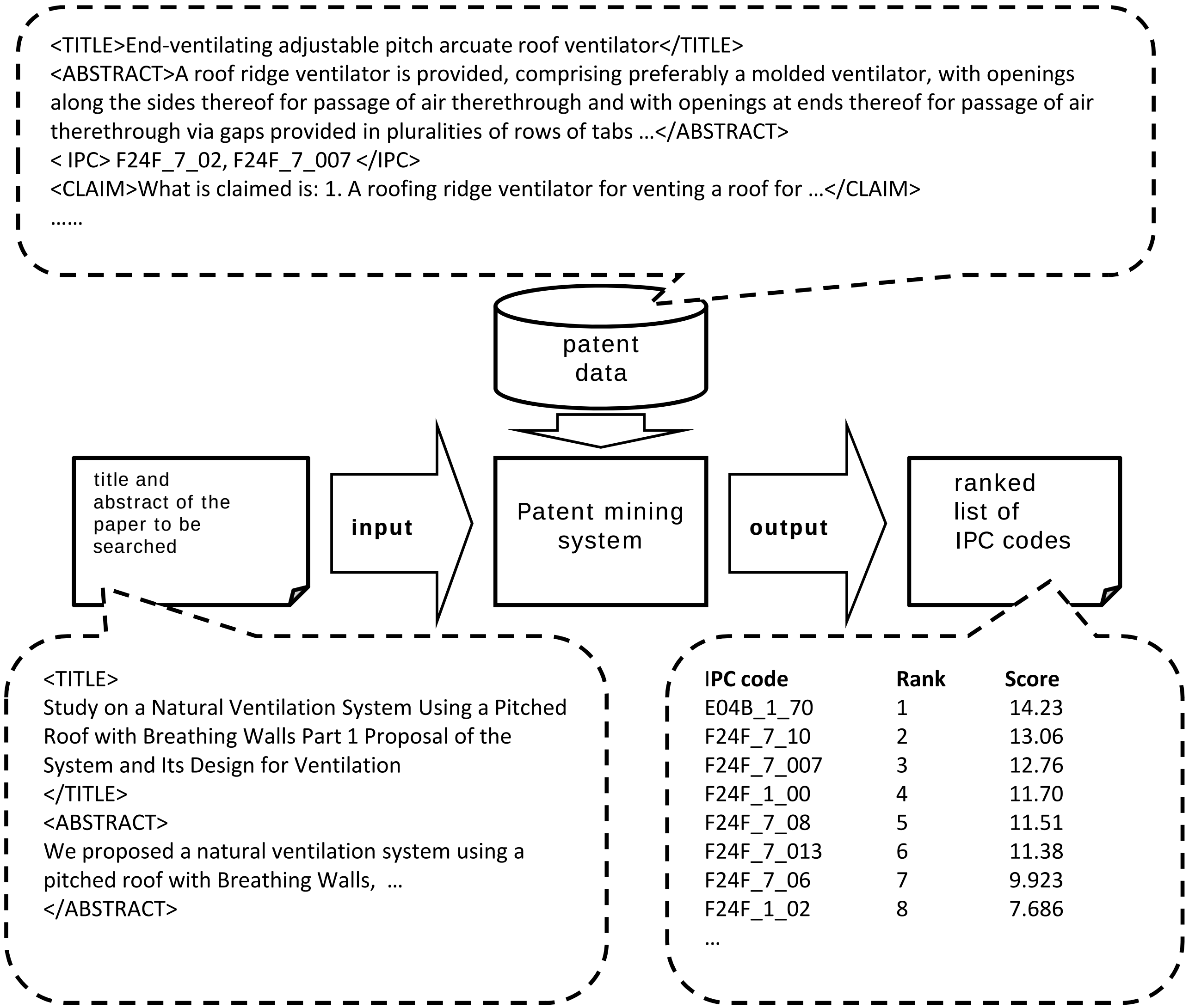
- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

Introduction of our group

- Natural Language Processing Laboratory, College of information science and engineering, Northeastern University
- Working on a variety of problems related to Natural Language Processing
  - Statistical machine translation
  - Syntactic parsing
  - Applied semantics ontology learning
  - Text mining
- Focus on patent mining from 2007
- Welcome to our homepage http://www.nlplab.com

Patent mining task at NTCIR-7

- Patent mining task
  - Mapping research papers into patent taxonomy (International Patent Classification)
- Three sub-tasks
  - English patent mining
  - Japanese patent mining
  - Cross language patent mining
  - We participated in the ***English patent mining*** sub-task

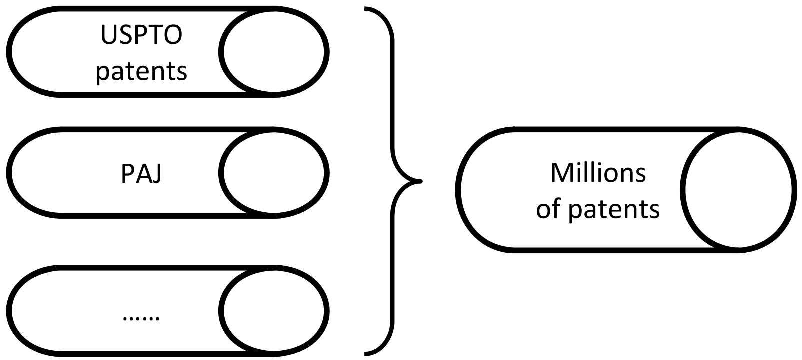


Outline

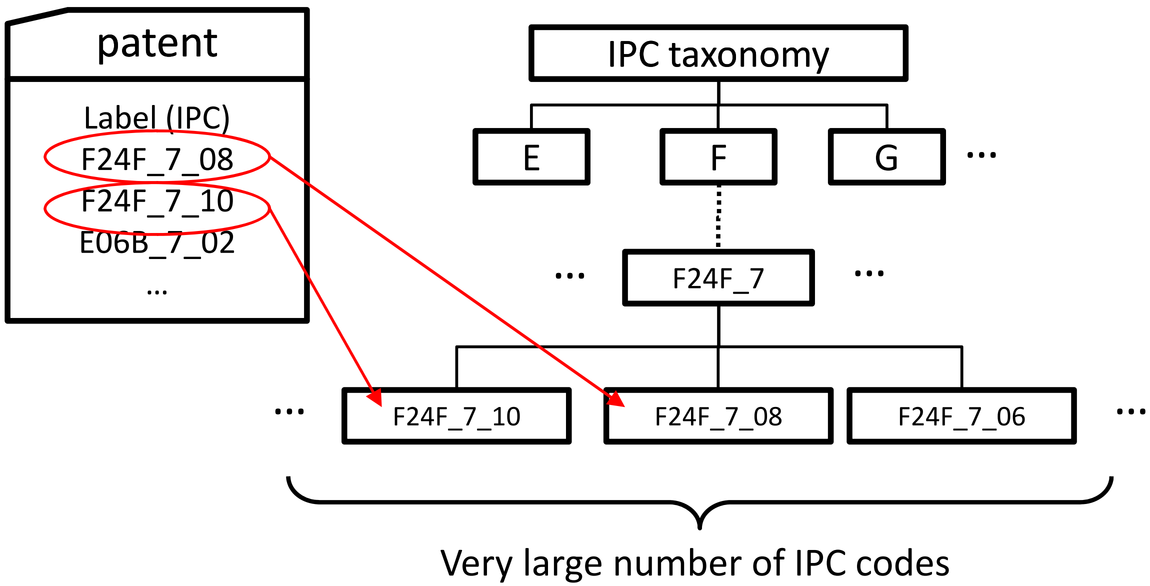
- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

Challenges

- Huge amount of training data
  - over 3 million training samples
  - how to train a supervised classifier or ranker

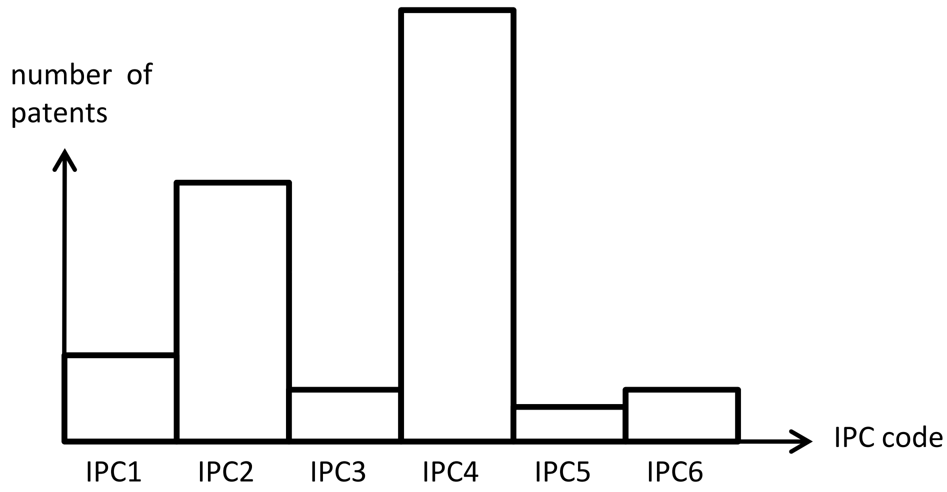


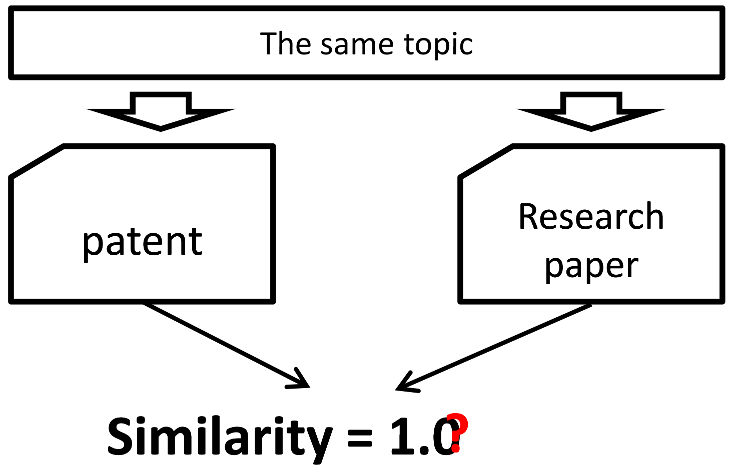
- Huge label set and multi-label
  - IPC is a hierarchical classification system which consists of more than 60,000 IPC codes.



Challenges

- Class imbalance problem of IPC
  - The distribution of IPC codes is skewed
- Different writing styles between research papers and patents
  - conflicts with the foundational hypothesis of supervised document classification theory

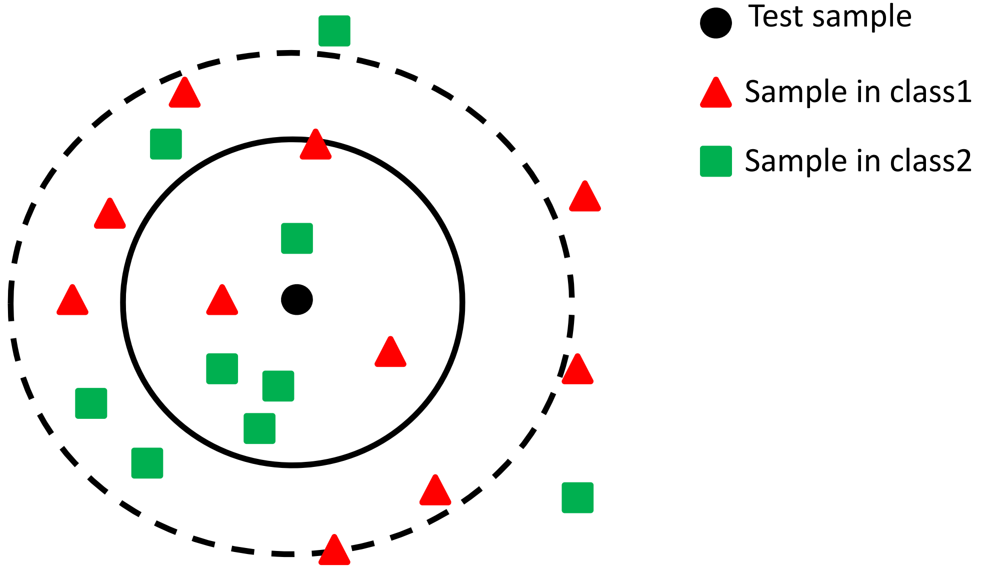




Motivation

- Difficult to apply sophisticated machine learning methods such as maximum entropy methods and support vector machines on patent mining
  - great deal of memory space and time cost is required task
  - no good solutions to multi-label classification on very large class set

- K-Nearest Neighboring (KNN) method is a comparatively easy solution
  - extracting similar examples and no training process is required
  - KNN is itself a ranking

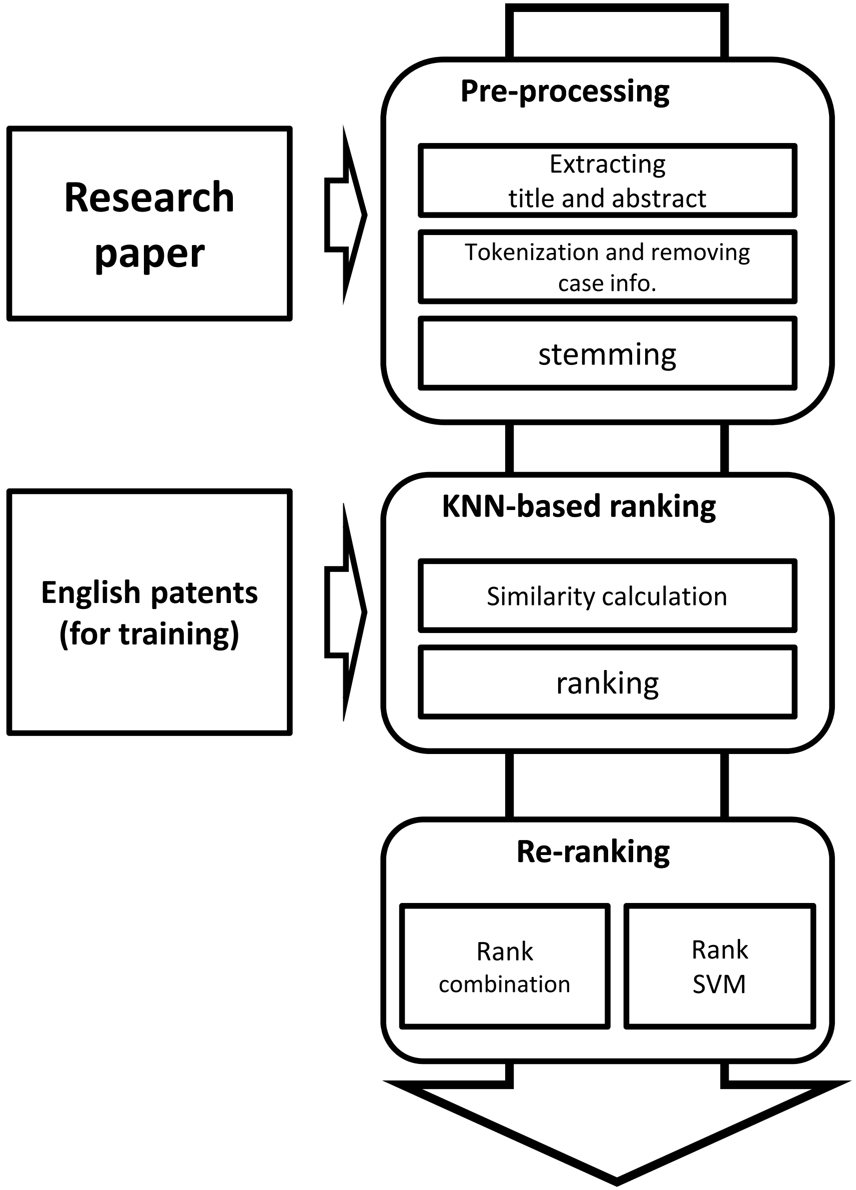


Outline

- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

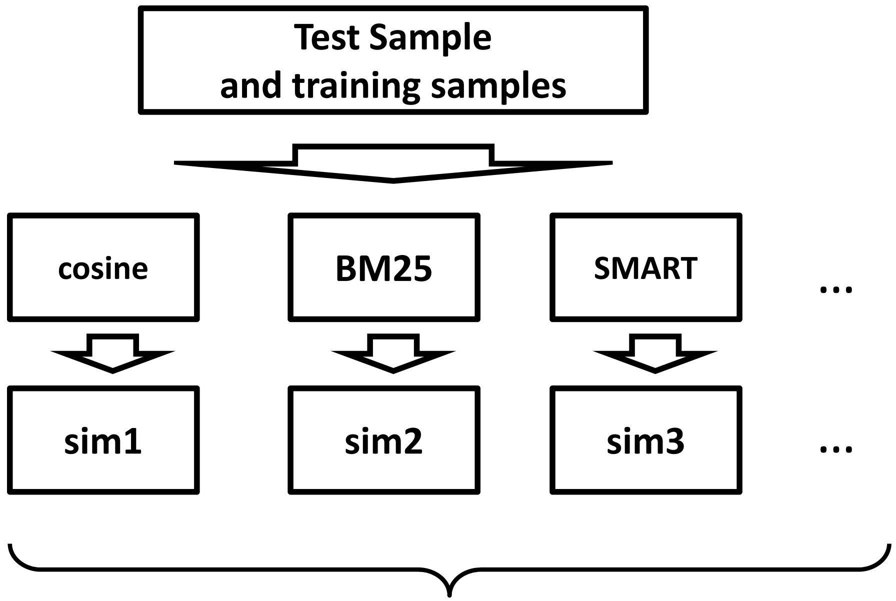
KNN-based method

- Key components
  - KNN-based ranking
  - Re-ranking
- Each document is represented as a vector in our system

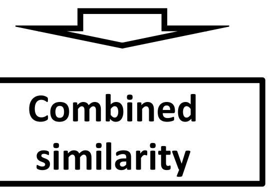


Similarity calculation

- Calculate the similarity between the test sample (research paper) and the training samples (patents)
- State-of-the-art methods
  - Cosine + tfidf
  - BM25 (Robertson et al, 1998)
  - SMART (Buckley et al, 1996)
  - PIV (Singhal et al, 1996)
  - Or some other …
- Log-linear method
  - Combine different similarities (features) to generate a refined similarity
  - Different weights to different features



$$Score_{\text{log-linear}}(c) = \frac{\exp(\sum_{m=1}^{M} \lambda_m \cdot Score_m(c))}{\sum_{c} \exp(\sum_{m=1}^{M} \lambda_m \cdot Score_m(c))}$$



Ranking

- 1. Original KNN ranking method:
  - Score each IPC code by the number of its occurrence in the extracted top-k documents
- 2. Naïve method
  - the order of IPC codes follows the order of their first occurrences in the extracted top-k documents
- 3. Sum/SumAver
  - score is calculated by summing up the similarities of all the extracted documents containing the given IPC code
  - For SumAver, we average the similarity for each sample
- 4. Listweak/ListweakAver
  - to emphasize the patents ranked in the frontier part of the list, a new factor is introduced
- 5. Weak/WeakAver
  - A drawback of KNN is the prediction of the input document tends to be dominated by the classes with the more frequent examples due to the class imbalance problem
  - Punish the classes which contain more training samples

Ranking – method 1

- 1. Original KNN ranking method:
  - Score each IPC code by the number of its occurrence in the extracted top-k documents
- 2. Naïve method
  - the order of IPC codes follows the order of their first occurrences in the extracted top-k documents
- 3. Sum/SumAver
  - score is calculated by summing up the similarities of all the extracted documents containing the given IPC code
  - For SumAver, we average the similarity for each sample

Suppose that we obtain the following list (top-5) after similarity calculation

Rank	Patent(id)	IPC	sim
1	p02	IPC1, IPC2	0.21
2	p03	IPC3, IPC4	0.11
3	p04	IPC2	0.09
4	p05	IPC2	0.09
5	p01	IPC1	0.07

Occurred 3 times

IPC	score
IPC2	3
IPC1	2
IPC3	1
IPC4	1

IPC list after ranking

Ranking – method 2

- 1. Original KNN ranking method:
  - Score each IPC code by the number of its occurrence in the extracted top-k documents
- 2. Naïve method
  - the order of IPC codes follows the order of their first occurrences in the extracted top-k documents
- 3. Sum/SumAver
  - score is calculated by summing up the similarities of all the extracted documents containing the given IPC code
  - For SumAver, we average the similarity for each sample

Suppose that we obtain the following list (top-5) after similarity calculation

Rank	Patent(id)	IPC	sim
1	p02	IPC1, IPC2	0.21
2	p03	IPC3, IPC4	0.11
3	p04	IPC2	0.09
4	p05	IPC2	0.09
5	p01	IPC1	0.07

first occurrence

second occurrence

IPC	score
IPC1	0.21
IPC2	0.21
IPC3	0.11
IPC4	0.11

IPC list after ranking

Ranking – method 3

- 1. Original KNN ranking method:
  - Score each IPC code by the number of its occurrence in the extracted top-k documents
- 2. Naïve method
  - the order of IPC codes follows the order of their first occurrences in the extracted top-k documents
- 3. Sum/SumAver
  - score is calculated by summing up the similarities of all the extracted documents containing the given IPC code
  - For SumAver, we average the similarity for each sample

Suppose that we obtain the following list (top-5) after similarity calculation

Rank	Patent(id)	IPC	sim
1	p02	IPC1, IPC2	0.21
2	p03	IPC3, IPC4	0.11
3	p04	IPC2	0.09
4	p05	IPC2	0.09
5	p01	IPC1	0.07

0.21 + 0.09 + 0.09 = 0.39

IPC	score
IPC2	0.39
IPC1	0.28
IPC3	0.11
IPC4	0.11

IPC list after ranking

Ranking – method 4

Suppose that we obtain the following list (top-5) after similarity calculation

Rank	Patent(id)	IPC	sim
1	p02	IPC1, IPC2	0.21
2	p03	IPC3, IPC4	0.11
3	p04	IPC2	0.09
4	p05	IPC2	0.09
5	p01	IPC1	0.07

Sim = 0.21 $\times$ 0.9^{1-1}
=0.21

Sim = 0.09 $\times$ 0.9^{3-1}
=0.07

Sim = 0.09 $\times$ 0.9^{4-1}
=0.06

Sim = 0.21 + 0.07 + 0.06 = 0.34

IPC	score
IPC2	0.34
IPC1	0.25
IPC3	0.10
IPC4	0.10

IPC list after ranking

- 4. Listweak/ListweakAver
  - to emphasize the patents ranked in the frontier part of the list, a new factor is introduced
- 5. Weak/WeakAver
  - A drawback of KNN is the prediction of the input document tends to be dominated by the classes with the more frequent examples due to the class imbalance problem
  - Punish the classes which contain more training samples

Ranking – method 5

Suppose that we obtain the following list (top-5) after similarity calculation

Rank	Patent(id)	IPC	sim
1	p02	IPC1, IPC2	0.21
2	p03	IPC3, IPC4	0.11
3	p04	IPC2	0.09
4	p05	IPC2	0.09
5	p01	IPC1	0.07

Suppose that there are 10 patents labeled with IPC2

Sim = $0.21 \times 0.9^{(1+10/5)}$ =0.15

Sim = $0.09 \times 0.9^{(2+10/5)}$ =0.06

Sim = $0.09 \times 0.9^{(3+10/5)}$ =0.05

Sim = 0.15 + 0.06 + 0.05 = 0.26

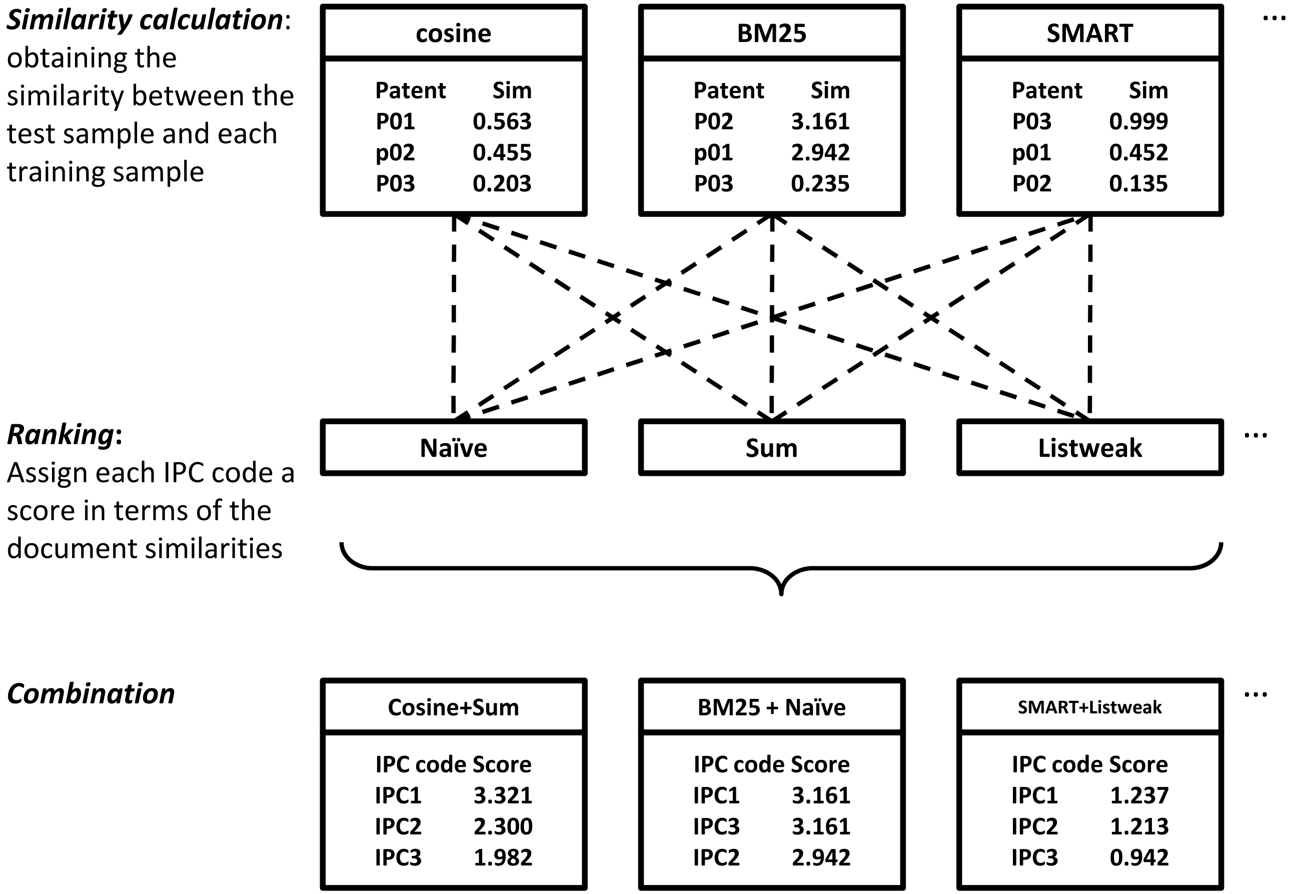
IPC	score
IPC2	0.26
IPC1	0.19
IPC3	0.07
IPC4	0.07

IPC list after ranking

- 4. Listweak/ListweakAver
  - to emphasize the patents ranked in the frontier part of the list, a new factor is introduced
- 5. Weak/WeakAver
  - A drawback of KNN is the prediction of the input document tends to be dominated by the classes with the more frequent examples due to the class imbalance problem
  - Punish the classes which contain more training samples

Re-ranking

- What have we had
  - Tens of ranked lists generated by different combinations of similarity calculation method and ranking method
- Motivation
  - Learn a better ranking from individual ranked lists (basic ranker)



Rank combination

- A linear combination of ranks in individual lists

List1

Rank	IPC code	Score
1	IPC1	3.321
2	IPC2	2.300
3	IPC3	1.982

List2

Rank	IPC code	Score
1	IPC1	3.161
2	IPC3	3.161
3	IPC2	2.942

List3

Rank	IPC code	Score
1	IPC1	1.237
2	IPC2	1.213
3	IPC3	0.942

$$1 / (\lambda_1 \times \text{rank}_1 + \lambda_2 \times \text{rank}_2 + \lambda_3 \times \text{rank}_3)$$

$$Score_{rank\text{-}combination}(c) = \frac{1}{\sum_{i=1}^{h} \lambda_i \cdot rankinlist(c, l_i)}$$

RankSVM

- Learn a ranking function
  - Each IPC is represent as a vector, in which the feature is the score in each ranked list

List1

Rank	IPC code	Score
1	IPC1	3.321
2	IPC2	2.300
3	IPC3	1.982

List2

Rank	IPC code	Score
1	IPC1	3.161
2	IPC3	3.161
3	IPC2	2.942

List3

Rank	IPC code	Score
1	IPC1	1.237
2	IPC2	1.213
3	IPC3	0.942

Feature vector of IPC3 : <1.982, 3.161, 0.942>

Outline

- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

Experiment

- Data (training)
  - Patent Abstracts of Japan (PAJ)
- Settings
  - Bag-of-words model
  - No feature selection
  - K = 100 (for KNN)
- Evaluation
  - Mean average precision (MAP)
- Re-ranking
  - Used 6 basic rankers for rank combination
  - Used 5 basic rankers for RankSVM

Experiment (cont.)

- KNN-based rankings (dry-run)

Ranking ¥ Sim	Cosine	BM25	SMART	PIV	Log-linear
Original KNN	35.16	34.79	35.78	34.51	35.05
Naïve	32.41	38.57	33.55	37.23	40.02
Sum	35.97	35.78	36.83	35.58	38.33
SumAver	35.05	35.92	36.46	34.13	38.05
Listweak	36.63	40.52	37.42	36.85	40.37
ListweakAver	34.85	40.88	37.65	36.79	41.11
Weak	36.25	36.53	37.11	35.91	38.24
WeakAver	33.42	36.15	34.90	33.01	38.38

- Re-ranking (dry-ran)

system	MAP
Rank combination	45.31
RankSVM	43.02

- Re-ranking (formal-ran)

system	MAP
Rank combination	48.86
RankSVM	47.21

- Ranking is a key factor that affects the performance of the basic KNN-based system
- Re-ranking can improve the performance of the basic KNN-based system significantly

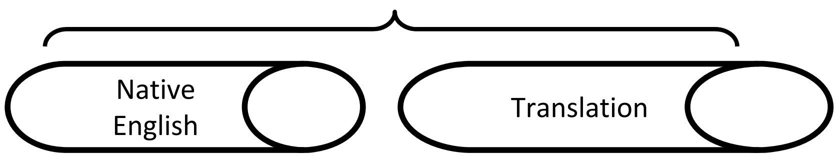
Outline

- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

Discussion – Issue 1

- Single label vs. multi-label
  - Both the training data of single label (USPTO data set) and multi-label (PAJ data set) are provided within this task.
  - However we found that the data of USPTO shows harmful to our system. The performance degrades when we trained the system on USPTO data solely or a mixed data set of "USPTO+PAJ", comparing to training on PAJ data

Another problem:
How to train a system on heterogeneous data ?



Discussion – Issue 2

- Two types of ranking techniques used
  - The first one is based on position of each candidate in the output list, such as Naïve, Rank combination.
  - The second one is based on the similarity score of each candidate, such as Sum and RankSVM.
- The first type of ranking is effective though they are simple.

Discussion – Issue 3

- Does patent structure really help ?
  - Make use of features in different sections, such as title, abstract and claim.
  - It seems not helpful
  - Need further study

<TITLE>End-ventilating adjustable pitch arcuate roof ventilator***</TITLE>***
<ABSTRACT>A roof ridge ventilator is provided, comprising preferably a molded ventilator, with openings along the sides thereof for passage of air therethrough and with openings at ends thereof for passage of air therethrough via gaps provided in pluralities of rows of tabs …**</ABSTRACT>**
< IPC> F24F_7_02, F24F_7_007 </IPC>
<CLAIM>What is claimed is: 1. A roofing ridge ventilator for venting a roof for air passage between the interior of a roof and the outside ambient through sides of the ventilator and through ends of the ventilator…**</CLAIM>**
……

Outline

- Overview
- Basic idea
- Methodology
  - KNN-based method
  - Re-ranking
- Experiment
- Discussion
- Summary

Summary

- We participated in NTCIR-7 English patent mining sub-task
  - KNN-based method
  - Re-ranking
- In future
  - Try to apply our techniques to patent mining tasks, such as patent prior art searching.

Thank you!