Berkeley Chinese Information Retrieval at TREC-5: Technical Report

Jianzhang He, Jack Xu, Aitao Chen, Jason Meggs and Fredric C. Gey
UC DATA
University of California
Berkeley, CA 94720

December 1996

1 Introduction

For Chinese track in TREC5, the collection from the People's Daily and Xinhua News articles includes 164,761 documents and the volume of the collection is about 170 MB. There are 28 queries. The task of Chinese track is to submit 1000 documents for each query ranked in the order of likelihood of relevance to the query.

It is a well known problem that there is no separator between Chinese words so that Chinese words cannot be directly used to index or search text as it is allowed in English. Therefore, people used characters, n-grams, or words as search-able tokens (character is 1-gram). In TREC5, we tried to use any meaningful string within text as indexing or search tokens. Our basic strategy is to use an exhaustive dictionary to segment document collection and queries and to use Berkeley TREC2 ad hoc retrieval algorithm to rank retrieved documents.

2 The dictionary

Our method depends heavily on the exhaustive Chinese dictionary. We obtained a public domain Chinese dictionary of 91,000 words and phrases from the World Wide Web Chinese site http://www.ifcss.org/ftp-pub/software/data. We expanded our dictionary by working on the TREC5 Chinese collection. The process of the expansion of the dictionary is as the following:

1. segment the collection with the available dictionary.
2. collect unmatched strings.
3. rank the unmatched strings by their frequencies.
4. manually select the top ranked entries.
5. add the selected entries to the dictionary.

The above process was iterated several times until another 46,659 entries were added to the dictionary. The criteria to select an entry is whether the entry is meaningful in Chinese. Therefore, the entries added to the dictionary includes words, idioms, personal names, geographical names, etc.

We also manually constructed a Chinese stopword list which includes 444 entries for which some of them are single character words.

3 The segmentation method

There are different requirements on Chinese segmentation for different applications. For example, machine translation and natural language processing require to segment Chinese text correctly according Chinese syntax. For the purpose of information retrieval, we may not have to segment Chinese text correctly since the ultimate goal is to improve retrieval performance. Although people have used mechanical generated tokens, such as single characters and n-gram, as indexing and search tokens, we believe that using meaningful tokens is better than using mechanically generated tokens. Therefore, we applied our dictionary as a primary tool to segment the collection and queries.

Segmentation based on a big dictionary can have the following three basic formats. The first is the longest match, for which text is sequentially scanned to match the dictionary. The longest matched strings are taken as indexing and search tokens and shorter tokens within it are discarded. Since longer tokens in the dictionary are more specific, longest match will generate less tokens with more specific meaning.

The second is the shortest match, for which text is scanned sequentially to match the dictionary. The first matched tokens are taken and the match process started from the next character. With the shortest match method, the segmentation process will generate more tokens with less specific meaning.

For example, from the string 数字电子计算机 (digital computer), the longest match will produce the whole string as a token but if we apply the shortest match, we will get the following tokens: 数字 (digit), 电子 (electron), 计算 (compute), and 机 (machine).

The third is the overlap match, for which tokens generated from the text can overlap each other across the matching boundary. For example, the string from query 14, 艾滋病毒 (AIDS virus) is segmented into 艾滋病 (AIDS) and 毒 (poison,toxin) with the longest match method. It is segmented into 艾滋病 (AIDS) and 病毒 (virus) with overlap match. In this case, overlap match produce better result than the regular longest match but we expect it will also generate some negative effects sometimes.

Based on the three basic match methods, we can use some combination of them. For example, we can have longest match plus overlap match, we also can have shortest match plus overlap match , or we can have all three together. For TREC-5, we used longest match plus overlap match. Since no dictionary is expected to include all the words, there are must be some unmatched strings left with dictionary based segmentation. Unmatched strings can be used as tokens for indexing and search. It is also can be used to expand the dictionary. There are several ways to treat unmatched strings:

1. take each character in the string as a token;
2. take the whole string as a token;
3. both 1) and 2);
4. discard.

We used 2) in our TREC-5 runs.

4 Manual query expansions

It is allowed to exam initial search result and then to adjust queries to achieve better performance for any manual ad hoc runs in TREC-5. We reconstructed the 28 queries by checking on our initial search results. We did three things on the queries:

1. add new words;
2. change weights (frequency) of words;
3. add negative words.

The process iterated several time before we obtained our final version of our queries. For example, our manually constructed query 14 is as the following:

Averagely, we spend about 2.8 hours per query to reconstruct the 28 queries. The retrieval result of manual queries improved 40% over our automatic run.

5 Negative terms

In some of the TREC queries, it is indicated that documents in some categories are not relevant to a particular query and sometimes a query will specify that documents in a category are relevant and otherwise are not relevant. For example, a query may only needs to retrieve documents about U.S.A and at the same time documents about other countries are not relevant. We developed the strategy to use negative terms to adjust the final weights of a document to a query. We only apply negative term weight adjust to our manual runs.

Table 1: Query 14: Cases of AIDS in China.

我国 (our country)(16)	艾滋病 (AIDS) (15)	中国 (China)(11)
监测 (control)(6)	发现 (found) (6)	感染 (infect)(6)
病毒 (virus)(6)	预防 (prevent) (5)	HIV(4)
患者 (patient)(4)	防治 (prevent and cure)(4)	
预防为主(prevention first) (3)	病例 (medical cases)(3)	
云南 (Yunnan)(3)	病人 (patient) (2)	
性病 (sexual transmitted disease)(1)	吸毒 (drug use)(1)	
注射器 (syringe)(1)	传染 (infect)(1)	

Table 2: Negative terms for query 14.

美国 (The U.S.)(2)	德国 (Germany) (2)	脊髓 (spinal cord)(2)
灰质 (grey matter)(2)	埃博拉 (2)	脑炎 (encephalitis)(2)
血吸虫 (blood fluke)(2)	丝虫 (filaria) (2)	非洲 (African)(2)
癌症 (cancer)(2)	香港 (Hong Kong)(1)	加拿大 (Canada)(1)
欧美 (The West)(1)		

The adjustment is to reduce the weight of a document if negative terms appear in the document. What we did is to divide the final weight of the document by the square root of 1 plus the sum of the negative term frequency within the document.

Our negative term experiment shows that the current strategy improves precision on the higher rank and degrades on the lower rank. The overall performance is not as good as without negative terms but all the change may be not statistically significant. We still believe that negative terms can play some role but it deserves further investigation.

6 Results

By the time of TREC-5 conference, only 19 out of 28 queries were judged. Our manual run is the best with a 46.10% average precision. We retrieved 1364 out of 1399 relevant documents. we have 10 best and 2 almost best queries out of the 19 queries. We achieved a 97.50% overall recall. Our automatic run averaged 31.92% precision with 15 queries above average. Our average precision is 13.63% better than the median average precision. The following is the detailed list prepared by Fred.

TREC-5 Chinese Manual Summary 10/28/96

Statistics computed over 5 manual runs.

	#	Relevant Retr. @ 100			Relevant Retr @ 1000			Average Precision			
Topic	Rel.	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst	BrklyCH2
1	13	11	9	7	13	13	11	0.1781	0.1350	0.0489	0.1350
2	69	38	29	27	69	63	58	0.4084	0.2993	0.2377	0.4084*
3	29	26	24	21	29	29	27	0.5791	0.3684	0.2897	0.3684
4	51	27	26	24	51	49	45	0.3130	0.2531	0.2344	0.3127+
5	28	11	9	6	28	25	22	0.0984	0.0732	0.0453	0.0942+
6	77	50	15	1	76	45	32	0.4632	0.0713	0.0144	0.4632*
7	17	12	10	6	17	14	12	0.5029	0.3345	0.2305	0.3345
8	43	41	36	15	43	42	37	0.7889	0.6032	0.1600	0.7889*
9	122	76	72	47	122	117	96	0.7689	0.6950	0.4175	0.7689*
10	49	28	22	14	43	33	29	0.2324	0.1800	0.1154	0.2324*
11	186	72	48	34	182	155	148	0.5472	0.3128	0.2299	0.5472*
12	119	47	39	28	116	113	63	0.3512	0.3100	0.1557	0.3512*
13	110	61	40	31	107	107	68	0.4875	0.3371	0.2089	0.4875*
14	57	50	34	21	56	46	38	0.6074	0.4768	0.1590	0.4768
15	71	50	41	6	71	64	33	0.5939	0.4535	0.0472	0.5939*
16	59	30	26	22	56	56	42	0.3602	0.3008	0.1519	0.3008

21	238	95	91	88	237	231	207	0.8956	0.8035	0.6138	0.8956*
22	15	15	15	11	15	15	15	0.7690	0.7481	0.2409	0.7481
23	46	30	25	18	46	45	40	0.5222	0.4521	0.2678	0.4521

1399 total relevant found Average over 19 queries 0.3794 0.4610

1364 found by BrklyCH2 at 1000

* = best

\+ = almost best

==

TREC-5 Chinese Automatic Runs with comparison to BrklyCH1 10/28/96

Statistics computed over 15 automatic runs.

	#	Relevant Retr. @ 100			Relevant Retr @ 1000			Average Precision			
Topic	Rel.	Best	Median	Worst	Best	Median	Worst	Best	Median	Worst	BrklyCH1
1	13	13	9	0	13	13	0	0.1972	0.1021	0.0000	0.1021
2	69	41	29	11	68	60	23	0.4365	0.2390	0.0244	0.3079
3	29	26	23	6	29	28	10	0.5228	0.3482	0.0178	0.4144
4	51	29	24	6	51	50	18	0.3690	0.2442	0.0247	0.3683+
5	28	13	9	6	28	22	8	0.1090	0.0612	0.0374	0.0471-
6	77	24	12	2	74	45	25	0.2116	0.0640	0.0163	0.2006+
7	17	13	9	5	16	13	8	0.4613	0.2896	0.0612	0.2096-
8	43	31	18	6	43	37	20	0.4462	0.1717	0.0264	0.3376
9	122	76	58	9	122	116	18	0.7575	0.5355	0.0117	0.3485-
10	49	26	14	3	45	32	11	0.2899	0.1293	0.0102	0.1700
11	186	47	39	15	179	155	49	0.4069	0.2643	0.0275	0.3059
12	119	40	30	17	119	105	41	0.3381	0.2348	0.0691	0.2165-
13	110	45	27	3	108	101	3	0.3767	0.1819	0.0016	0.1818
14	57	22	8	2	57	21	2	0.2562	0.0548	0.0008	0.0715
15	71	59	36	2	71	65	16	0.7172	0.4507	0.0222	0.5141
16	59	37	21	2	59	56	10	0.4506	0.2915	0.0046	0.2915
21	238	96	90	16	237	230	40	0.9083	0.7956	0.0248	0.7956
22	15	15	13	4	15	15	4	0.7754	0.4914	0.0667	0.5346
23	46	40	26	0	46	44	0	0.6658	0.3874	0.0000	0.6473

1399 Average precision over 19 queries 0.2809 0.3192

1246 relevant found by BrklyCH1

\- = below median

\+ = close to best

After TREC-5 conference, the 28 queries were all judged. Our average precision is 49.05% and we retrieved 2095 documents out of 2182 relevant documents. Our overall recall is 96.01%. We believe our manual run is still ranked first. Our automatic run averaged 35.68% precision and retrieved 1948 documents out of 2182 relevant documents with the overall recall of 89.28%.

7 Credits

Dr. Fred Gey provided the leadership and methodology advisory for the project. Jianzhang He is the coordinator of the Chinese track project. The following is the list of people and their contribution for Berkeley TREC-5 Chinese retrieval project.

1. Overall methodology development: Jianzhang He, Fred Gey.
2. Manual dictionary expansion: Jianzhang He, Jack Xu.
3. Stopword list construction: Jack Xu, Jianzhang He.
4. Chinese collection filtering: Jack Xu.
5. Segmentation software coding: Jason Meggs.
6. Retrieval software: Aitao Chen.
7. Interactive document check software: Aitao Chen.
8. Modifying SMART to handle Chinese text: Jianzhang He.

9. Negative term retrieval software implementation: Jianzhang He.
10. Manual query reconstruction: Jianzhang He, Jack Xu, Aitao Chen.

Prof. Bill Cooper also provided helpful advises during the process of the project. Berkeley TREC-5 Chinese Retrieval result was presented by Jianzhang He at NIST on 11/21/96.