Lexicon Effects on Chinese Information Retrieval

K.L. Kwok
Computer Science Dept., Queens College, City University of NY,
Flushing, NY 11367, USA.
kklqc@cunyvm.cuny.edu

Abstract

We investigate the effects of lexicon size and stopwords on Chinese information retrieval using our method of short-word segmentation based on simple language usage rules and statistics. These rules allow us to employ a small lexicon of only 2,175 entries and provide quite admirable retrieval results. It is noticed that accurate segmentation is not essential for good retrieval. Larger lexicons can lead to incremental improvements. The presence of stopwords do not contribute much noise to IR. Their removal risks elimination of crucial words in a query and adversely affect retrieval, especially when the queries are short. Short queries of a few words perform more than 10% worse than paragraph-size queries.

1 Introduction

It is well known that a sentence in Chinese (or several other oriental languages) consists of a continuous string of 'characters' without delimiting white spaces to identify words. In Chinese, the characters are called ideographs. This makes it difficult to do machine studies on these languages since isolated words are needed for many purposes, such as linguistic analysis, machine translation, etc. Automatic methods for correctly isolating words in a sentence -- a process called word segmentation -- is therefore an important and necessary first step to be taken before other analysis can begin. Many researchers have proposed practical methods to resolve this problem such as (Nie et al., 1995, Wu and Tsang, 1995, Jin & Chen, 1996, Ponte & Croft, 1996, Sproat et al., 1996, Sun et al., 1997).

Information retrieval (IR) deals with the problem of selecting relevant documents for a user need that is expressed in free text. The document collection is usually huge, of gigabyte size, and both queries and documents are domain unrestricted and unpredictable. When one does IR in the Chinese language with its peculiar property, then one would assume that accurate word segmentation is also a crucial first step before other processing can begin.

However, in the recent 5th Text REtrieval Conference (TREC-5) where a fairly large scale Chinese IR experiment was performed [Kwok and Grunfeld, 199x], we have demonstrated that a simple word segmentation method, couple with a powerful retrieval algorithm, is sufficient to provide quite good retrieval results. Moreover, experiments by others using even simpler bigram representation of text (i.e. all consecutive overlapping two characters), both within and outside the TREC environment, also produce good results [Ballerini et al., 199x, Buckley et al., 199x, Chien, 1995, Liang et al., 1996]. This is a bit counter-intuitive because the bigram method leads to three times as large an indexing feature space compared with our segmentation (approximately 1.5 million vs 0.5 million), and one would expect that there are many random, non-content matchings between queries and documents that may adversely affect precision. Apparently, this is not so. Based on this observation, we made some adjustments to our lexicon, and provide some experimental results of the lexicon effects on retrieval effectiveness.

2 Short-Word Segmentation

While word segmentation for linguistic analysis may aim at the longest string that carry a specific semantic content, this may not be ideal for IR because one then has to deal with the problem of partial string matching when a query term matches only part of a document term or vice versa. Instead, we aim at segmenting texts into short words of one to three characters long that function like English content terms. Our process is based on the following four steps A to D:

A) facts - lookup on a manually created 2175-entry lexicon called L0. This is small, consisting of commonly used words of 1 to 3 characters, with some proper nouns of size 4. Each entry is tagged as 0 (useful: total 1337), 1 (stopword: 671), s (symbol: 88), 6 (numeric: 37), 4 (punctuation: 9), and 2 or 3 for the rules below. Other researchers have used lexicons of hundreds of thousands. We do not have such a large resource; besides, maintenance of such a list is not trivial. We try to remedy this via rules.

Given an input string, we scan left to right and

perform longest matching when searching on the lexicon. Any match will result in breaking a sentence into smaller chunks of texts. Fig.1b shows the result of processing an original TREC query (Fig.1a) after our lexicon lookup process.

B) rules - for performing further segmentation on chunks. Words in any language are dynamic and one can never capture 'all' Chinese words in a lexicon for segmentation purposes. We attempt to identify some common language usage ad-hoc rules that can be employed to further split the chunks into short words. The rules that we use, together with their rationale and examples and counter-examples are described below:

Rule D (for double): any two adjacent similar characters xx are considered stopwords -- this identifies double same characters that are often used as adjectives or adverbs that do not carry much content (see ex.1-3 below). However, some Chinese names do use double same characters (ex.4) and we would 'stop' them wrong. Other cases such as 'Japan Honshu' (ex.5), 'U.S. Congress' (ex.6) requires splitting between the same two characters. In these cases we rely on 'Japan' or 'U.S.' being on the lexicon and identified first before applying this rule.

Rule D

Examples:		
(1)	天天	daily
(2)	慢慢	slowly
(3)	处处	every where
Counter-Examples:		
(4)	何中中	person name
(5)	日本本州	Japan Honshu
(6)	美国国会	U.S. Congress

Rule 2: Px, where P is a small set of 31 special characters, are stopwords for any x -- these characters are tagged '2' in our lexicon and examples are shown

Rule 2

P = {一, 这, 那, 认, .. }

Examples:		
(7)	一枝	a branch/stick of
(8)	一早	early
(9)	一齐	together
(10)	(这,那)种	(this, that) kind
(11)	(这,那)次	(this, that) time
(12)	认为	consider to be
(13)	认真	in earnest
Counter-Examples:		
(14)	一国	one country
(15)	认错	admit mistake

below (ex.7-13). When character p is tagged '2', we also try to identify common words where p is used as a word in the construct yp, and these are entered into the lexicon. yp may or may not be a stopword. This way a string like ..ypx.. would be split 'yp x' rather than 'y px', dictionary entries being of higher precedence. This rule works in many cases, but we believe that our list may be too long, and many words that have content (such as ex.14-15) are stopped.

Rule 3: xQ, where Q currently has only 2 special characters, are stopwords for any x -- these are tagged '3' and is a complement to Rule 2 (see ex.16-19 and counter-examples ex.20-21).

Rule 3

Q = { 们, 些 }

Examples:		
(16)	(我, 咱)们	we
(17)	他们	they
(18)	那些	those
(19)	多些	more
Counter-Examples:		
(20)	老师们	teachers
(21)	勤力些	more diligent

Rule E (for even): any remaining sequence of even number of characters are segmented two by two -- this arises from the observation that 70-80% of Chinese words are 2-characters long, and the rhythm of Chinese are often bi-syllable punctuated with mono-syllables and tri-syllables. If one can identify where the single character words occur, the rest of the string quite often can be split as such when it is even. These single characters are often stopwords that hopefully are in our lexicon. Examples 22 to 26 below show chunks that are even, being surrounded by punctuation signs or stopwords. They will be segmented correctly. Examples 27 to 29 show counter-examples with even number of characters that do not obey this rule.

In addition, numeric entries are also removed as stopwords although one can often detect a sequence of them and have it identified as a number.

C) frequency filter - after a first pass through the test corpus via steps A and B, a list of candidate short-words will be generated with their frequency of occurrence. A threshold is used to extract the most commonly occurring ones. These are our new short-words that are 'data-mined' from the corpus itself.

D) iteration - using the newly identified short-words of Step C all tagged useful for segmentation purposes, we expand our initial lexicon in step A and re-process the corpus. In theory, we could continue to iterate, but we have only done one round. With a frequency threshold value in Step C of 30, a final lexicon size of 15,234 called L01 was obtained.

We believe the rules we use for Step B, though

```
Rule E
------
Examples:
(22) .. 最 ! 惠国待遇分离.
        1
(23) .. 对 ! 南海诸岛 ! 的..
        1              1
(24) .. 的 ! 财产损失数目.
        1
(25) .. 在 ! 战火蹂躏 ! 的..
        1              1
(26) .. 代表 ! 讨论双边经贸 ! 关系
        0                    1
Counter-examples:
(27) .. 邓小平 ! 说:
(28)    据 ! 新华社 ! 报导,
(29) .. 共产党 ! 员.
```

simple, are useful. They naturally do not work always, but may work correctly often enough for IR purposes. Fig.1c shows the results of processing the TREC-5 query #28 based on these rules after Step A. Comparison with a manual short word segmentation of the set of 28 TREC-5 queries shows that we achieve 91.3% recall and 83% precision on average. It is possible that these queries are easy to segment. Our method of segmentation is certainly too approximate for other applications such as linguistic analysis, text-to-speech, etc. For IR, where the purpose is to detect documents with high probability of relevance rather than exact matching of meaning and is a more forgiving environment, it may be adequate. Besides, one also has other tools in IR to remedy the situation. These are discussed below.

3 The Retrieval Environment

Our investigations are based on the TREC-5 Chinese collection of 24,988 Xinhua and 139,801 People's Daily news articles totaling about 170 MB. To guard against very long documents which can lead to outlier in frequency estimates, these are divided into subdocuments of about 475 characters in size ending on a paragraph boundary. This produces a total of 247,685 subdocuments which are segmented into short-words as described in Section 2. In addition, the single characters from each word of length two or greater are also used for indexing purposes to guard against wrong segmentation.

Provided with the TREC-5 collection are 28 very long and rich Chinese topics, mostly on current affairs. They are processed like documents into queries. These topics representing user needs have also been manually judged with respect to the (most fruitful part of the) collection at NIST so that a set of relevant documents for each query is known. This allows retrieval results to be evaluated against known answers.

For retrieval, we use our PIRCS (acronym for Probabilistic Indexing and Retrieval - Components - System) engine that has been documented elsewhere [Kwok 1990,1995] and has participated in the past five TREC experiments with admirable results [see for example Kwok & Grunfeld 1996]. PIRCS is an automatic, learning-based IR system that is conceptualized as a 3-layer network and operates via activation spreading. It combines different probabilistic methods of retrieval that can account for local as well as global term usage evidence. Our strategy for ad-hoc retrieval involves two stages. The first is the initial retrieval where a raw query is used directly. The d best-ranked documents from this retrieval are then regarded as relevant without user judgment, and employed as feedback data to train the initial query term weights and to add new terms to the query - query expansion. This process has been called pseudo-feedback. This expanded query retrieval then provides the final result. This second retrieval in general can provide substantially better results than the initial if the initial retrieval is reasonable and has some relevants within the d best-ranked documents. The process is like having a dynamic thesaurus bringing in synonymous or related terms to enrich the raw query.

As an example of a retrieval, we have shown in Table 1 comparing the TREC-5 Chinese experiment using bigram representation with our method of text segmentation in the PIRCS system. The table is a standard for the TREC evaluation. Precision is defined as the proportion of retrieved documents which are relevant, and recall that of relevant documents which are retrieved. In general when more documents are retrieved, precision falls as recall increases. It can be

Represent'n:	**Bigram**	**Short-Word Segm**
Total number of documents over all queries		
Retrieved:	28000	28000
Relevant:	2182	2182
Rel_ret:	2125	2015
Interpolated Recall - Precision Averages:		
at 0.10	0.6978	0.6521
at 0.30	0.5428	0.5650
at 0.50	0.4477	0.4716
at 0.70	0.3688	0.3616
at 0.90	0.2592	0.2493
Average precision (non-interpolated) over all rel docs		
	0.4477	0.4516
Precision At:		
5 docs:	0.6429	0.6643
10 docs:	0.6036	0.6000
20 docs:	0.5625	0.5482
30 docs:	0.5214	0.5321
100 docs:	0.3796	0.3693
Exact:	0.4557	0.4522

Table 1: Bigram and Short-Word Segmentation Retrieval Results Averaged over 28 Queries

seen that the two methods provide quite similar performance - bigram method ranks 2125 of the 2182 known relevant documents within the first 1000 retrieved for the 28 queries while the short-word method has about 5% less, at 2015. The latter has a slight edge in average precision (0.4516 vs 0.4477). Average precision is often used as a standard for comparison.

The precision at different number of documents retrieved, a user-oriented measure, are also comparable in both cases.

4 Lexicon Effects on Retrieval

In bigram representation of text, no lexicon is used and many meaningless bigrams as well as many that are true stopwords are included. Yet they do not seem to affect retrieval effectiveness. We take this as a clue that stopword removal may not play an important role in Chinese IR and lead us to investigate its effect. We also like to see how lexicon size can affect retrieval. Usually one needs as large a dictionary as possible so that many segmentation patterns are available for the system to select the correct one.

An entry in our lexicon list can serve the purpose of a segmentation marker or, in addition, for detection of stopwords. In our system stopwords can be determined in three ways based on: lexicon, rule or frequency threshold (statistical). The last category arises from Zipfian behavior of terms and is standard for IR processing: features with frequencies that are too high or too low have adverse effects on retrieval effectiveness and efficiency. This is done as a default, and is also performed for bigrams.

Our lexicon-based stopwords consists of 671 entries in our list tagged as '1'. The major rule-based stopword removal is Rule 2, while others have minor effects because they occur much less often. A run through the collection shows that the number of times tag 1 and Rule 2 were exercised are about 1.9m and 2.1m.

We have enabled Rules D and E, tags 0,3 and 4 to be effective for segmentation as a default, and perform experiments where the lexicon (tag 1 & 6) and rule-based (Rule 2) stopword removal (and segmentation) can be activated or deactivated as follows:

	tag 1,6	**Rule 2**
ExpTyp.1		
segment	yes	yes
ExpTyp.2		
stop & segm		yes
segment	yes	
ExpTyp.3		
stop & segm	yes	
segment		yes
ExpTyp.4		
stop & segm	yes	yes

Lexicon:	<----		**L0**		---->	<----		**L01**		---->
ExpTyp:	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
Total number of documents over all queries										
Retrieved:	<-----				28000					----->
Relevant:	<-----				2182					----->
Rel_ret:	**2059**	**2062**	**2047**	**2046**	**2013**	**2058**	**2060**	**2041**	**2040**	**2012**
Interpolated Recall - Precision Averages:										
at .1	.688	.682	.699	.689	.671	.673	.676	.678	.675	.655
at .3	.557	.557	.555	.555	.549	.564	.563	.564	.568	.560
at .5	.467	.473	.470	.473	.466	.474	.481	.475	.483	.469
at .7	.375	.374	.373	.367	.356	.378	.376	.380	.376	.365
at .9	.249	.253	.246	.239	.233	.252	.257	.250	.254	.246
Average precision (non-interpolated) over all rel docs										
	.455	**.457**	**.456**	**.457**	**.448**	**.461**	**.462**	**.460**	**.460**	**.451**
Precision At:										
5 docs:	.650	.657	.664	.650	.650	.664	.657	.664	.643	.686
10 docs:	**.596**	**.589**	**.611**	**.611**	**.596**	**.593**	**.596**	**.621**	**.614**	**.607**
20 docs:	.564	.559	.557	.561	.566	.555	.552	.554	.558	.552
30 docs:	.526	.533	.535	.537	.537	.531	.533	.525	.536	.535
100 docs:	.373	.376	.372	.373	.368	.380	.377	.370	.371	.368
Exact:	.455	.465	.460	.463	.455	.453	.457	.462	.462	.452

Table 2: Effect of Lexicon-based and Rule-based Stopwords on Long Query Retrieval using L0 & L01

For example, ExpTyp.2 means lexicon entries with tags 1,6 are used for segmentation only, while those obeying Rule 2 serve to segment and removed as well. An ExpTyp.5 will be explained later. Retrievals using lexicons of four different sizes with long and short versions of the TREC-5 queries were performed and evaluated.

5 Results and Discussion

5.1 Long Queries

Table 2 tabulates the precision and recall values averaged over 28 long queries using L0, the 2175-entry and L01, the 15234-entry lexicons. In ExpTyp.1 under L0 for example, where tags 1 & 6 as well as Rule 2 are in effect for segmentation only, an average precision of 0.455 and recall of relevants (at 1000 retrieved) of 2059 out of 2182 are achieved. On average close to 5.96 out of the first 10 retrieved documents are relevant. This is very good performance for a purely statistical retrieval system. It is also interesting to see that the small lexicon is sufficient to yield this good result. Indirectly, it shows that our rule-based segmentation (Rule D, E, 2) can define sufficiently good features for retrieval, and remedies our deficiency in lexicon size. When both tag 1,6 entries and Rule 2 are used for stopword removal (ExpTyp.4, L0), average precision remains practically the same at 0.457. Similarly for ExpTyps.2 & 3 L0, where either Rule2 or tag 1,6 are used for stopword removal, effectiveness does not seem to alter much. Removal of tag 1,6 words however decreases the number of relevants slightly from 2060 to around 2040. It appears that the presence of stopwords have little effect on Chinese IR, just as noticed for bigrams.

ExpTyp.5 L0 in Table 2 is included as a demonstration of the perils associated with stopword removal. It shows about a 2% drop in average precision as well as in relevants retrieved compared with ExpTyp.4 L0 due to bad result of one single query. Query #19 asks for documents on 'Project Hope', and the Chinese query is shown below. The

TREC-5 Chinese Query #19:

希望(= hope)工程 (= project).

中国，希望工程，文化程度，教育
相关文件应提到希望工程是什么，它的
目标为何，实施成果如何，有关改进
教师待遇，文化扶贫工作与捐款等文件
亦属相关文件．不相关文件包括听众
信箱之问题，或文件提到教育法但未
提具体法案内容，或仅提希望工程之名
但没有具体数据以及推行办法者．

word 'hope' is often used in the context of 'We hope to/that..' or 'My hope is ..' and quite non-content bearing. It is not unreasonable to regard it as a stopword in both English and Chinese. However, for this query it is crucial. ExpTyp.5 L0 is done under the same circumstances as ExpTyp.4 L0 except that the word 'hope' is changed to be a stopword (tag 1). This query then practically accounts for all the adverse effect. Since the presence of stopwords has been shown to have a benign effect on Chinese retrieval, it appears advisable to keep them as indexing terms to guard against such unexpected results.

In Table 2 under L01, we repeat the same experiments using our larger lexicon which is derived from the collection using L0 as the basis. It is seen that the larger lexicon improves average precision by about 1%, from around 0.456 to about 0.461. Otherwise, the two sets of experiments are qualitatively similar. Since retrieval is crucially dependent on how well the queries are processed, it appears that the 28 are well-prepared for retrieval using the original 2175-entry lexicon.

Recently, we further augment our L0 to a larger initial lexicon L1 with 27,147 entries. This derives L11, a 42,822-entry lexicon from the collection based on our segmentation procedure. Results of repeating the retrieval experiments using these two larger lexicons are shown in Table 3. There is incremental improvements in average precision by using the larger lexicon: e.g. for ExpTyp.1, from 0.455 (L0) to 0.463 (L11), about 2%. The removal of stopwords for L11 (ExpTyp.4 vs 1) does not lead to much difference,

Lexicon:	**<- L1 ->**		**<- L11 ->**		
ExpTyp.:	**1**	**4**	**1**	**4**	**5**
Total number of documents over all queries					
Retrieved:	<------		28000		------->
Relevant:	<------		2182		------->
Rel_ret:	**2062**	**2056**	**2061**	**2056**	**2008**
Interpolated Recall - Precision Averages:					
at .1	.684	.673	.696	.695	.688
at .3	.555	.553	.558	.567	.558
at .5	.478	.475	.478	.479	.465
at .7	.381	.375	.384	.379	.358
at .9	.254	.262	.256	.262	.247
Average precision (non-interpolated) over all rel docs					
	.460	.459	.463	.464	.451
Precision At:					
5 docs:	.650	.643	.671	.693	.693
10 docs:	**.614**	**.604**	**.604**	**.611**	**.607**
20 docs:	.561	.555	.563	.550	.543
30 docs:	.529	.524	.525	.521	.516
100 docs:	.373	.373	.373	.374	.366
Exact:	.460	.466	.461	.468	.458

Table 3: Effect of Lexicon-based and Rule-based Stopwords on Long Query Retrieval using L1 and L11

but the peril of accidentally removing a crucial word remains, leading again to about 2% drop in effectiveness (ExpTyp.5 vs 4 L11).

5.2 Short Queries

It has been pointed out that the paragraph-size TREC queries are long and unrealistic because real-life queries are usually very short, like one or two words. One or two words, on the other hand, often do not supply sufficient clues to a retrieval engine. To study the effects of lexicons on short queries, we further perform retrievals using only the first sentence of each query that belongs to the 'title' section of an original topic. They average to a few short-words and we hope to see more pronounced effects. These results are shown in Table 4.

As expected, retrieval effectiveness decreases substantially over 10% compared to the full length queries: from around 0.463 to 0.409 (ExpTyp.1 L11, Tables 3&4). The larger lexicon L11 also has an edge over L0 (average precision 0.409 vs 0.398 Table 4), and the use stopwords (ExpTyp.4 vs 1 L11) can improve precision as for long queries, but the accidental removal of a crucial word can lead to a much bigger adverse effect of 6% drop in average precision (ExpTyp.5 vs ExpTyp.4). Especially hard hit is the number of relevants at 1000 retrieved, which decreases by 11% (1962 vs 1732). The reason for this pronounced effect is that when a query is short (like two words 'Project Hope') and a crucial word ('Hope') is removed, what is left for retrieval is practically useless. In long queries however, many other terms are still available to remedy the removed crucial word,

Lexicon:	**<- L0 ->**		**<- L01 ->**			**<- L1 ->**		**<-- L11 -->**		
ExpTyp:	**1**	**4**	**1**	**4**	**5**	**1**	**4**	**1**	**4**	**5**
Total number of documents over all queries										
Retrieved:	<-----				28000					----->
Relevant:	<-----				2182					----->
Rel_ret:	**1958**	**1929**	**1961**	**1914**	**1684**	**1970**	**1952**	**1975**	**1962**	**1732**
Interpolated Recall - Precision Averages:										
at .1	.608	.596	.609	.614	.579	.579	.578	.586	.605	.569
at .3	.502	.498	.500	.486	.456	.496	.495	.492	.493	.458
at .5	.410	.409	.410	.415	.383	.420	.426	.427	.434	.402
at .7	.336	.346	.345	.344	.321	.348	.349	.351	.355	.333
at .9	.217	.223	.227	.233	.232	.234	.235	.234	.241	.241
Average precision (non-interpolated) over all rel docs										
	.398	**.405**	**.408**	**.407**	**.382**	**.405**	**.409**	**.409**	**.417**	**.391**
Precision At:										
5 docs:	.579	.550	.579	.564	.529	.586	.586	.593	.607	.571
10 docs:	**.534**	**.532**	**.568**	**.554**	**.518**	**.550**	**.554**	**.564**	**.571**	**.536**
20 docs:	.495	.496	.516	.502	.466	.488	.489	.488	.495	.459
30 docs:	.466	.474	.481	.473	.437	.467	.464	.469	.475	.439
100 docs:	.334	.336	.339	.335	.301	.330	.329	.333	.335	.301
Exact:	.403	.404	.406	.406	.381	.399	.405	.398	.409	.385

Table 4: Effect of Lexicon-based and Rule-based Stopwords on Short Query Retrieval using L00, L01, L1 & L11.

and the effect is less pronounced.

6 Conclusion

For the TREC-5 Chinese collection of documents and queries, it is found that a small 2175-lexicon coupled with some simple linguistic rules is sufficient to provide indexing features for good retrieval results. Larger lexicons can give incremental improvements. Lexicon or rule-based stopword removal have negligible effect on retrieval with long queries. For short queries with a large lexicon, stopword elimination can lead to some improvements, but runs the risks of accidentally deleting a crucial word in a query that can adversely affect retrieval significantly. It appears advisable to keep all stopwords and use them for segmentation purposes. One needs only retain high and low frequency thresholds to screen out frequency-based statistical stopwords. Experimentation with more varied queries is needed to verify these findings.

7 Acknowledgments

This work is partially supported by a Tipster grant from the U.S. Department of Defense. Xianlin Zhang and Jing Yan helped prepare the lexicons.

References

Buckley, C., Singhal, A & Mandar, M. 199x. Using query zoning and correlation within SMART: TREC 5. In: The Fifth Text REtrieval Conference (TREC-5). Harman, D.K. (Ed.). To appear.

Chien, L.F. 1995. Fast and quasi-natural language search for gigabytes of Chinese texts. In: Proc. 18th ACM SIGIR Conf. on R&D in IR. Fox, E., Ingwersen, P. & Fidel, R. (eds.) ACM:NY, NY. pp.112-120.

Jin W. & Chen, L. 1995. Identify unknown words in Chinese corpus. In: Proc. of 3rd NLP Pacific-Rim Symposium (NLPRS'95). Seoul, Korea. Vol.1, pp.234-9.

Kwok, K.L. & Grunfeld, L. 199x. TREC-5 English and Chinese retrieval experiments using PIRCS. In: The Fifth Text REtrieval Conference (TREC-5). Harman, D.K. (ed.). To appear.

Kwok, K.L. 1990. Experiments with a component theory of probabilistic information retrieval based on single terms as document components. ACM Transactions on Office Information Systems, 8:363-386.

Kwok, K.L. 1995. A network approach to probabilistic information retrieval. ACM Transactions on Office Information Systems, 13:325-353.

Liang, T, Lee, S.Y & Yang W.P. 1996. Optimal weight assignment for a Chinese signature file. Information Processing & Management, 2:227-237.

[NiBR95] Nie, J.Y, Hannan, ML & Jin, WY (1995). Unknown word detection and segmentation of Chinese using statistical and heuristic knowledge

[PoCr96] Ponte, J & Croft, W.B (1996). USeg: a retargetable word segmentation procedure for information retrieval. In: Symposium on document analysis and information retrieval (SDAIR '96).

Sproat, R., Shih, C., Gale W. & Chang, N. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. Computational Linguistics, 22:377-404.

Sun, M., Shen, D. & Huang, C. 1997. CSeg&Tag1.0: A practical word segmenter & POS tagger for Chinese texts. In: Proc. 5th Conference on Applied Natural Language Processing, Mar 31 - Apr 3, 1997. pp.119-124.

Wu, Z & Tseng, G. 1995. ACTS: An automatic Chinese text segmentation system for full text retrieval. Journal of the American Society of Information Science, 46:83-96.

(a) TREC-5 Chinese Query #28: The Spread of Cellular Phones in China

---

移动电话在中国的成长 .
数字,蜂窝式,移动电话,网络,自动漫游

相关文件应包括下列信息: 中国移动电话用户数,
覆盖地区, 中国如何以数据分组交换网覆盖
全国移动电话的通讯. 不相关文件则包括 有关
制造移动电话厂商的报道, 以及移动电话的厂牌.

(b) Initial Segmentation using Lexicon L0 only:

```
移动电话 ¦ 在 ¦ 中国 ¦ 的 ¦ 成长 ¦
          1        1       4
数字 ¦ 蜂窝式 ¦  移动电话 ¦ 网络 ¦ 自动 ¦
     4       4          4     4  2
漫游相关文件 ¦ 应 ¦ 包括 ¦ 下列信息 ¦
              1    1          4
中国 ¦ 移动电话 ¦ 用户 ¦ 数 ¦ 覆盖 ¦ 地区 ¦
 0               0       4        0   4
中国 ¦ 如何 ¦ 以 ¦ 数 ¦ 据 ¦ 分组 ¦ 交换 ¦
 0     1     1        1           0
网覆盖 ¦ 全国 ¦ 移动电话 ¦ 的 ¦ 通讯 ¦ 不 ¦
          0             1       4 1
相关文件 ¦ 则 ¦ 包括 ¦ 有关 ¦ 制造移动电话厂商 ¦
          1     1     1
的 ¦ 报道 ¦ 以及 ¦ 移动电话 ¦ 的 ¦ 厂牌 ¦
1        4  1              1        4
```

(c) Further Segmentation Result using Rule E:

```
移动 ¦ 电话 ¦ 在 ¦ 中国 ¦ 的 ¦ 成长 ¦
    E
数字 ¦ 蜂窝式 ¦ 移动 ¦ 电话 ¦ 网络 ¦ 自动 ¦
              E
漫游 ¦ 相关 ¦ 文件 ¦ 应 ¦ 包括 ¦ 下列 ¦ 信息 ¦
    E      E                    E
中国 ¦ 移动 ¦ 电话 ¦ 用户 ¦ 数 ¦ 覆盖 ¦ 地区 ¦
          E
中国 ¦ 如何 ¦ 以 ¦ 数 ¦ 据 ¦ 分组 ¦ 交换 ¦

网覆盖 ¦ 全国 ¦ 移动 ¦ 电话 ¦ 的 ¦ 通讯 ¦ 不 ¦
                   E
相关 ¦ 文件 ¦ 则 ¦ 包括 ¦ 有关 ¦ 制造 ¦ 移动 ¦ 电话 ¦ 厂商 ¦
    E                          E      E      E
的 ¦ 报道 ¦ 以及 ¦ 移动 ¦ 电话 ¦ 的 ¦ 厂牌 ¦
                        E
```

Fig.1(a-c): A TREC-5 Query and its Processing by Lexicon and Rules