The University of Amsterdam at NTCIR-5

Jaap Kamps[1,2] Michiel van der Bruggen[1] Maarten de Rijke[2]
[1] Archives and Information Studies, University of Amsterdam
[2] Informatics Institute, University of Amsterdam
Email: kamps@uva.nl, m.j.van.der.bruggen@hva.nl, mdr@science.uva.nl

Abstract

We describe the University of Amsterdam's participation in the Cross-Lingual Information Retrieval task at NTCIR-5. We focused on Chinese monolingual retrieval, and aimed to study the effectiveness of language models and different tokenization methods for Chinese. Our main findings are the following. First, where the vector space model excels on a bigram index, the language model performs poorly. Second, on a unigram index, the language model is very effective, and even exceeds the performance of the vector space model on the bigram index. Third, and at a more technical level, in comparison to word-based langauges such as English we found that language models for Chinese require less smoothing, due to the different indexing unit.
Keywords: *CLIR, Chinese IR, Language Models*

1 Introduction

This paper details our participation in the Cross-Lingual Information Retrieval task at NTCIR-5. As a first-time participant at NTCIR, we decided to focus entirely on the Single Language IR subtask for Chinese. For details about the CLIR task and the SLIR subtask, we refer to [3]. Our main aims for participating in the Chinese monolingual retrieval subtask are as follows:

- Study the effectiveness of language models for Chinese.
- Study the effectiveness of different tokenizations.
- Study the effectiveness of blind relevance feedback on top of the various tokenizations we consider.

The textbook approach to Chinese IR [5, 7, e.g.,], is to build an index of character bigrams and use a vector-space retrieval model. Just to cite an authoritative overview [7, p.239]:

> It is obvious that, across different collections, the vector space model using character-based indexing gives the worst MAP values, but provides the best MAP values using bigram-based indexing. The difference in MAP values between character-based and bigram-based indexing is substantial, from 10% to 20%.

A look at participants of the NTCIR 4 CLIR task and their approaches to Chinese monolingual retrieval confirms this observation [4]. Although more advanced approaches were also used, indexes built on character bigrams and vector-space retrieval models are frequently applied.

The remainder of this paper is organized as follows. Next, in §2, we detail our experimental setup, and the retrieval techniques used. Then, in §3, we discuss in detail the results for our official submissions, as well as a range of post-submission experiments. Finally, in §4, we discuss our findings and draw some conclusions.

2 Experimental Setup

Our retrieval system is based on the Lucene engine with a number of home-grown extensions [2, 6].

2.1 Tokenization

Although tokenization is important for all languages, Asian languages such as Chinese present a special challenge since there are no spaces separating words. This makes it impossible to (directly) apply many of the standard approaches to tokenization, since they are all based on the unit of words [1].

We restrict our attention here to the language-independent technique of character n-gramming. Specifically, we build two different indexes:

Bigrams Our indexing unit is a pair of adjacent characters. That is, a stream of characters like `bigram` is indexed as the five tokens `bi ig gr ra am`.

Unigrams Our indexing unit is simply the individual character. That is, a stream of characters like `unigram` is indexed as the seven tokens `u n i g r a m`.

2.2 Topic fields

We look at three different queries derived from the same topic statement:

Title the short topic statement in the topic's title, i.e., the content of the ⟨title⟩ field;

Description the long topic statement in the topic's description, i.e., the content of the ⟨desc⟩ field; and the

Verbose the verbose topic statement from the topic's description and narrative, i.e., the content of the ⟨desc⟩ and ⟨N⟩ fields.

2.3 Retrieval models

For our ranking, we use either a vector-space retrieval model or a language model.

Our vector space model is the default similarity measure in Lucene [6], i.e., for a collection D, document d and query q:

$$sim(q,d) = \sum_{t \in q} \frac{tf_{t,q} \cdot idf_t}{norm_q} \cdot \frac{tf_{t,d} \cdot idf_t}{norm_d} \cdot coord_{q,d} \cdot weight_t,$$

where

$$\begin{aligned} tf_{t,X} &= \sqrt{\text{freq}(t,X)} \\ idf_t &= 1 + \log \frac{|D|}{\text{freq}(t,D)} \\ norm_q &= \sqrt{\sum_{t \in q} tf_{t,q} \cdot {idf_t}^2} \\ norm_d &= \sqrt{|d|} \\ coord_{q,d} &= \frac{|q \cap d|}{|q|} \end{aligned}$$

Our language model is an extension to Lucene [2], i.e., for a collection D, document d and query q:

$$P(d|q) = P(d) \cdot \prod_{t \in q} \left((1-\lambda) \cdot P(t|D) + \lambda \cdot P(t|d) \right),$$

where

$$\begin{aligned} P(t|d) &= \frac{tf_{t,d}}{|d|} \\ P(t|D) &= \frac{\text{doc\_freq}(t,D)}{\sum_{t' \in D} \text{doc\_freq}(t',D)} \\ P(d) &= \frac{|d|}{\sum_{d' \in D} |d|} \end{aligned}$$

The standard value for the smoothing parameter λ is 0.15.

2.4 Smoothing

In the language modeling framework, smoothing plays an important role: it helps to overcome data-sparseness, and it introduces an inverted document frequency effect, and it expresses the relative importance of query terms [9]. Since our indexing unit for Chinese is very different from collections in word-based languages such as English, it may require a different amount of smoothing. Hence, we will study the effect of smoothing on retrieval effectiveness for the various tokenization methods.

2.5 Feedback

Query expansion using pseudo-relevance (or blind) feedback is a technique that leads to improvement of the average retrieval effectiveness in almost all settings. We use a straightforward language modeling approach to feedback [8]. Our main interest is to investigate the effectiveness of blind feedback for the different retrieval models and the different approaches to tokenization.

3 Experimental Results

In this section, we discuss the results for our official submissions to NTCIR, as well as a range of post-submission experiments.

3.1 Retrieval model

Due to the late moment at which we decided to participate in NTCIR-5, we could only realize part of the envisioned experiments before the official submission deadline. We submitted the following 5 runs:

`ILPS-C-C-T-01` Vector space model using the title query.

`ILPS-C-C-D-02` Vector space model using the description query.

`ILPS-C-C-DN-03` Vector space model using the verbose (description and narrative) query.

`ILPS-C-C-T-04` Language model using the title query.

`ILPS-C-C-D-05` Language model using the description query.

All official runs use an index based on character bigrams.

Table 1 lists the results for our official submissions, with both the *relaxed* and *rigid* assessments. The first

	Relaxed		Rigid	
	MAP	P@10	MAP	P@10
ILPS-C-C-T-01	0.3247	0.5180	0.2763	0.3700
ILPS-C-C-D-02	0.2855	0.4720	0.2365	0.3420
ILPS-C-C-DN-03	0.3970	0.5940	0.3485	0.4580
ILPS-C-C-T-04†	0.1049	0.2380	0.1033	0.1880
ILPS-C-C-D-05†	0.0102	0.0200	0.0085	0.0140

Table 1. Results of our official submissions to NTCIR-5. Runs marked † were affected by a technical error; here, we list the scores of the corrected submissions.

column lists the run identifier; the second and third columns list the mean average precision and the precision at 10 for the relaxed assessments; the fourth and fifth columns list the scores for the rigid assessments. Some obvious observations present themselves. First, the performance of the title query exceeds that of the longer description query (rows 1 vs 2, and 4 vs 5). Second, the verbose query results in the best performance (row 3). Third, on the bigrams index, the vector space model is much more effective than the language model (rows 1-3 vs rows 4 and 5).

3.2 Tokenization

For our post-submission experiments, we constructed a second index, one based on character unigrams. The underlying motivation was to study the relative effectiveness of the vector space model and the language model for the bigram and the unigram tokenizations. Table 2 shows the results. Again, we make a number of observations. First, unigram tokenization leads to somewhat lower performance for the vector space model. Second, the unigram tokenization is much more effective for the language model. In fact, the language model on the unigram index outperforms the score for the vector space model on the bigram index.

Let us zoom in even further on the relative relative effectiveness of bigram and unigram tokenization for the vector space model. Table 3 shows, for the title queries, the topics with the largest gain or decrease in absolute score. An inspection of the queries reveals that bigrams may preserve meaning that is lost with unigrams. For example, the most dramatic loss of performance is for topic 23, whose title contains the name of the space station "Mir." In Chinese this translates into three characters "Hé Píng Haò" (in Pinyin). With bigrams the meaning is largely preserved by the first two characters "Hé Píng" meaning "peace." With unigrams, the first character "Hé" on its own can have many meanings, including the frequent "and." As a result, the value of the retrieval cue "Mir" is lost, and performance drops dramatically. For cases where the unigrams are more effective than the bigrams, the analysis is less straightforward. We generally see an increase in recall relative to the bigrams.

3.3 Phrases

One of the main differences between the unigram and bigram tokenization is that the adjacency of characters is lost in the unigram index. We can try to rectify this by introducing phrases in the query. For example, from a stream of characters like `unigram` we can derive six phrases of adjacent characters like "`u n`" "`n i`" "`i g`" "`g r`" "`r a`" "`a m`." We use the phrase based query either by itself, or in combination with the unigram query.

Table 4 shows the results of phrase based queries for the unigram index. We see the following: First, for the vector space model, introducing phrases in the query increases the retrieval performance. In fact, the phrase based queries on the unigram index now outperform the scores for the bigram index. A case in point is topic 23 about "Mir," discussed above, which now scores 0.4193 for a query consisting of only phrases of adjacent characters. Second, for the language model, introducing phrases in the query decreases the retrieval performance.

Topic	Bigrams	Unigrams	Difference
23	0.3989	0.0010	−0.3979
35	0.4972	0.2410	−0.2562
43	0.2576	0.0116	−0.2460
36	0.4743	0.2496	−0.2247
32	0.3174	0.1012	−0.2162
⋮	⋮	⋮	⋮
31	0.0737	0.1446	+0.0709
22	0.3455	0.4446	+0.0991
37	0.3595	0.4625	+0.1030
30	0.5135	0.6558	+0.1423
18	0.4520	0.8645	+0.4125

Table 3. Topics with largest absolute gain/decrease of MAP for two types of tokenization (bigram, unigram). Results of the vector space model on title queries using the rigid assessments.

	Bigrams		Unigrams	
	MAP	P@10	MAP	P@10
Title, Vector Space	0.2763	0.3700	0.2591	0.3440
Description, Vector Space	0.2365	0.3420	0.2179	0.3280
Title, Language Model	0.1033	0.1880	0.2817	0.4100
Description, Language Model	0.0085	0.0140	0.2501	0.3400

Table 2. Results of different tokenizations and different retrieval models using the rigid assessments.

	Adding Phrases		Only Phrases	
	MAP	P@10	MAP	P@10
Title, Vector Space	0.2827	0.3780	0.2869	0.3800
Description, Vector Space	0.2596	0.3560	0.2398	0.3400
Title, Language Model	0.2593	0.3960	0.1047	0.1940
Description, Language Model	0.2515	0.3420	0.0073	0.0140

Table 4. Results of using phrases in the query for the unigram index, using the rigid assessments.

3.4 Smoothing

There is a considerable amount of research into smoothing for word-based languages such as English [9]. Given the different nature of the indexing unit for Chinese, we want to study the appropriate amount of smoothing for the respective indexes. Table 5 lists the scores for different values of the smoothing parameter. The highest scores are obtained for λ = 0.5 (bigrams) and λ = 0.4 (unigrams). Whereas the optimal value for word-based languages is usually at the lower end, we see that for Chinese less smoothing is needed for the bigram or unigram indexes.

3.5 Query expansion

Finally, we look at the effectiveness of query expansion based on pseudo relevance feedback. Table 6(Top) lists the results for the vector space model. We see that query expansion is effective for promoting retrieval effectiveness under all conditions. The increase, however, is larger for the bigrams index. Table 6(Bottom) lists the results for the language model. Again, we see that query expansion leads to substantial improvements in retrieval effectiveness under all conditions.

	Bigrams		Unigrams	
λ	MAP	P@10	MAP	P@10
0.1	0.0932	0.1820	0.2701	0.4000
0.2	0.1082	0.1920	0.2861	0.4120
0.3	0.1114	0.2000	0.2862	0.4120
0.4	0.1144	0.2020	0.2883	0.4160
0.5	0.1157	0.2060	0.2869	0.4040
0.6	0.1151	0.2080	0.2852	0.4000
0.7	0.1140	0.2000	0.2811	0.3920
0.8	0.1125	0.1920	0.2761	0.3840
0.9	0.1107	0.1980	0.2700	0.3860

Table 5. Results of T-only queries varying the smoothing parameter, λ, using the rigid assessments.

4 Discussion and Conclusions

This paper detailed the University of Amsterdam's participation in the CLIR task of NTCIR-5. We focused on Chinese monolingual retrieval, and aimed to study the effectiveness of language models, of different tokenizations, and of query expansion.

Our main findings are as follows. First, the vector space model and bigram tokenization are an effective combination. Performance of the vector space model on the unigram index is somewhat less, but can be improved by using a phrase-based query. Second, language models and unigram tokenization are an effective combination. The language models perform unimpressively on the bigrams index. Third, due to the different indexing unit, language models for Chinese require less smoothing than the standard value for word-based languages like English. Fourth, query expansion based on pseudo relevance feedback improves retrieval effectiveness under all conditions, whether using bigram or unigram tokenization, and whether using the vector space model or language model.

Acknowledgments

Jaap Kamps was supported by grants from the Netherlands Organization for Scientific Research

	Bigrams		Unigrams	
	MAP	P@10	MAP	P@10
Title, Vector space	0.2589	0.3420	0.2869	0.3800
5 docs, 10 terms	0.3334	0.4380	0.3135	0.3920
5 docs, 20 terms	0.3529	0.4500	0.3322	0.4120
5 docs, 30 terms	0.3565	0.4600	0.3272	0.4120
5 docs, 50 terms	0.3496	0.4500	0.3093	0.4020
10 docs, 10 terms	0.3464	0.4420	0.3107	0.3800
10 docs, 20 terms	0.3612	0.4400	0.3333	0.4020
10 docs, 30 terms	0.3717	0.4600	0.3345	0.4160
10 docs, 50 terms	0.3780	0.4660	0.3227	0.3900
Title, Language model ($\lambda = 0.3$)	0.1114	0.2000	0.2862	0.4120
5 docs, 10 terms	0.1822	0.2560	0.3319	0.4200
5 docs, 20 terms	0.1914	0.2560	0.3422	0.4240
5 docs, 30 terms	0.1860	0.2640	0.3380	0.4360
5 docs, 50 terms	0.1826	0.2660	0.3286	0.4260
10 docs, 10 terms	0.1727	0.2360	0.3304	0.4260
10 docs, 20 terms	0.1827	0.2480	0.3572	0.4760
10 docs, 30 terms	0.1803	0.2420	0.3644	0.4760
10 docs, 50 terms	0.1665	0.2260	0.3553	0.4740

Table 6. Results of expanding T-only queries considering the top n docs pseudo-relevant and selecting at most m terms, using the rigid assessments. (Top): Vector space model. (Bottom): Language model.

(NWO) under project numbers 612.066.302 and 640.-001.501. Maarten de Rijke was supported by grants from NWO under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 612-13-001, 612.-000.106, 612.000.207, 612.066.302, 612.069.006, and 640.001.501.

References

[1] V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual document retrieval for European languages. *Information Retrieval*, 7(1):33–52, 2004.

[2] ILPS. The ILPS extension of the Lucene search engine, 2005. `http://ilps.science.uva.nl/Resources/`.

[3] K. Kishida, K.-H. Chen, S. Lee, K. Kuriyama, N. Kando, H.-H. Chen, and S. H. Myaeng. Overview of CLIR tasks at the fifth NTCIR workshop. In *Proceedings of the NTCIR-5*, 2005.

[4] K. Kishida, K.-H. Chen, S. Lee, K. Kuriyama, N. Kando, H.-H. Chen, S. H. Myaeng, and K. Eguchi. Overview of CLIR tasks at the fourth NTCIR workshop. In *Proceedings of NTCIR-4*, 2004.

[5] K. L. Kwok. Comparing representations in Chinese information retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference*, pages 34–41. ACM Press, New York NY, USA, 1997.

[6] Lucene. The Lucene search engine, 2005. `http://jakarta.apache.org/lucene/`.

[7] R. W. Luk and K. L. Kwok. A comparison of Chinese document indexing strategies and retrieval models. *ACM Transactions on Asian Language Information Processing*, 1:225–268, 2002.

[8] J. M. Ponte. Language models for relevance feedback. In W. B. Croft, editor, *Advances in Information Retrieval*, The Kluwer International Series in Information Retrieval, chapter 3, pages 73–95. Kluwer Academic Publishers, Boston, 2000.

[9] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.