Using Unigram and Bigram Language Models for Monolingual and Cross-Language IR

Lixin Shi Jian-Yun Nie
Département d'informatique et de recherche opérationnelle, Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal, Québec, H3C 3J7 Canada
{shilixin, nie}@iro.umontreal.ca

Abstract

Due to the lack of explicit word boundaries in Chinese, and Japanese, and to some extent in Korean, an additional problem in IR in these languages is to determine the appropriate indexing units. For CLIR with these languages, we also need to determine translation units. Both words and n-grams of characters have been used in IR in these languages; however, only words have been used as translation units in previous studies. In this paper, we compare the utilization of words and n-grams for both monolingual and cross-lingual IR in these languages. Our experiments show that Chinese character n-grams are reasonable alternative indexing and translation units to words, and they lead to retrieval effectiveness comparable to or higher than words. For Japanese and Korean IR, bi-grams or a combination of bigrams and unigrams produce the highest effectiveness.

Keywords: *Bigram, Unigram, Language Model, Cross-Language IR.*

1 Introduction

The common problem in Chinese, Japanese and Korean processing is the lack of natural word boundaries. Even though some spaces are added in Korean sentences, they often separate a sentence into phrases instead of words. For all these languages, we have to determine the indexing units by an additional process – either using word segmentation or by cutting the sentence into n-grams (usually unigrams and bigrams) [4]. The latter is a simple method that does not require any linguistic resource.

The utilization of unigrams and bigrams for Chinese, Japanese and Korean IR has been investigated in several previous studies. The investigations have been carried out for monolingual retrieval only. It has been found that using a combination of unigrams and bigrams for IR in these languages can be as effective as using a word segmenter [6] [8]. However, there is no investigation on using n-grams for cross-language information retrieval (CLIR) with these languages. We do not know if the utilization of n-grams as translation units can be as effective as words in CLIR. In our experiments in NTCIR6, we focus on this problem. We have compared several alternatives for monolingual IR: using words (with the help of a segmenter), using n-grams (unigrams and bigrams), and using some combinations of them. For CLIR, we have only tested for English-Chinese CLIR due to our limited linguistic resources. We also focused on the comparison between n-grams and words as translation units in query translation. Our results show that using n-grams in CLIR, we can also achieve effectiveness equal to or better than words.

In this report, we will describe the general approach we used. Then we will describe our experimental results.

2 Background

Our general retrieval model is based on language modeling (LM). So let us first describe the general LM approach we used.

The basic approach of language modeling to IR is to build a statistical language model for each document, and then determine the likelihood that the document model generates the query as the score function [10]. An alternative is to build a language model for each document as well as for the query. A score of document is determined by the divergence between them. A common score function is defined based on KL-divergence as follows:

$$\begin{aligned} Score(D,Q) &= -KL(\theta_Q \,||\, \theta_D) \\ &= -\sum_{w\in V} p(w|\theta_Q)\log\frac{p(w|\theta_Q)}{p(w|\theta_D)} \\ &\propto \sum_{w\in V} p(w|\theta_Q)\log p(w|\theta_D) \end{aligned} \quad (1)$$

where θ_D and θ_Q are respectively the language models for the document *D* and the query *Q*.

In CLIR, words in Q and D are in different languages. Query translation can be integrated into the query model $p(w|\ \theta_Q)$ as follows:

$$\begin{aligned} p(t_i \mid \theta_{Q_s}) &= \sum_j p(s_j, t_i \mid \theta_{Q_s}) \\ &= \sum_j t(t_i \mid s_j, \theta_{Q_s}) p(s_j \mid \theta_{Q_s}) \\ &\approx \sum_j t(t_i \mid s_j) p(s_j \mid \theta_{Q_s}) \end{aligned} \tag{2}$$

where s_j is a word in source language (language of the query), t_i is a word in target language (language of the documents), and $t(t_i|s_j)$ is a translation probability between s_j and t_i. This probability can be obtained from a translation model trained on a parallel corpus. In our case, we use IBM model 1 [1] trained using GIZA++[1]. A similar approach has been used in [5] for CLIR between European languages, in which s_j and t_i are words. For Chinese, Japanese and Korean languages, an additional question is what units to use for query translation and document indexing.

3 Our Approaches

3.1 Monolingual IR

Firstly, we re-examine the problem of monolingual IR in Chinese, Japanese and Korean. Several studies have compared the utilization of words and character n-grams as indexing units for Chinese IR [6] [8]. Most of them have been done in models other than language modeling. Here, we re-examine the impact of different indexing units within the language modeling framework.

Previous studies on Chinese word segmentation showed that segmentation accuracy in Chinese is usually higher than 90%. This accuracy is shown to be satisfactory for IR [7].

We notice that many similar Chinese words share some common characters. Therefore, a natural extension to word-based indexing of documents and queries is to add characters as additional indexing units. [7] showed that this approach is effective for Chinese IR. Following the same principle, we can create several possible indexing units for Chinese documents: word, unigram, bigram, word+character, and bigram+character. In the last two cases, we cut each Chinese sentence into both words and characters or into bigrams and characters (unigrams). For example, the original Chinese sentence "企业科研投资" (company's investment in R&D) can be transformed respectively into:

企业/科研/投资/企/业/科/研/投/资

企业/业科/科研/研投/投资/企/业/科/研/投/资

Then every separate unit (word, bigram or unigram) are considered as an index. In particular, we can have the following possible basic indexing strategies:

- W (Word): Sentences are segmented into words, and words are used as indexing units.
- U (Unigram): Sentences are cut into single characters or unigrams, which are used as indexing units.
- B (Bigram): Sentences are cut into overlapping bigrams of characters.
- WU (Word and Unigram): Sentences are segmented into both words and single characters. Both words and unigrams are indexing units.
- BU (Bigram and Unigram): Sentences are cut into both overlapping character bigrams and single characters.

Previous studies have shown that the last two approaches are more effective than the others [8]. However, these approaches do not have much room for setting the relative importance between words, bigrams and unigrams.

Another possible approach to combine different index units is as follows: we can create several indexes for the same document, using words, unigrams and bigrams separately. Then during the retrieval process, these indexes are combined to produce a single ranking function. In LM framework, this means that we build several language models for the same document and query. Each type of the model determines a score function $Score_i$. The final score is a combination of the scores. So, in general, we define the final score as follows:

$$Score(D,Q) = \sum_i \alpha_i Score_i(D,Q) \tag{3}$$

where $Score_i$ is the score determined by a type of model (in our case, either unigram, bigram or word model) and α_i its importance in the combination (with $\sum_i \alpha_i = 1$). In this way, we can set an appropriate relative importance to each type of index. In particular, we can have the following additional indexing strategy:

- B+U: Interpolate bigram and unigram models, defined as Formula (4).

$$Score_{B+U}(D,Q) = \lambda Score_B(D,Q) + (1-\lambda) Score_U(D,Q) \tag{4}$$

Our experiments on Chinese IR (next section) show that we can obtain good effectiveness using BU and B+U. Therefore, for Japanese and Korean, we only test the bigram, unigram, and their combinations instead of using a word segmentation method.

3.2 English to Chinese cross-language IR

For CLIR, we use a translation model, namely IBM model 1, to translate query Q_s from source language to target language.

[1] http://www.fjoch.com/GIZA++.html

Here, we use maximum likelihood estimation to estimate the source terms in the query, that is $P(s_j \mid \theta_Q) = \dfrac{c(s_j, Q_s)}{|Q_s|}$. The query model in Formula (2) becomes:

$$p(t_i \mid \theta_{Q_s}) = \sum_{s_j \in Q_s} t(t_i \mid s_j) \frac{c(s_j, Q_s)}{|Q_s|} \qquad (5)$$

where $c(s_j, Q_s)$ is occurrence of term s_j in query Q_s, and $|Q_s|$ is the number of terms in Q_s.

The simplest TM is English-Chinese word-to-word translation model, which can be trained from an English-Chinese parallel corpus (in which Chinese sentences are segmented into words). We denote this translation approach by W, which uses the probability in the translation table from English words into Chinese words.

The monolingual IR results show that using bigram and unigram model can achieve the same performance as word, and combining the bigram and unigram can perform even better.

Now let us describe the training process of bigram and unigram TMs and their combination in the retrieval process. Similarly to monolingual Chinese IR, there are two ways to combine bigram and unigram translation models:

(1) One can process the parallel sentence (Chinese part) so as to transform it into both bigrams and unigrams. Then a standard training process is used to train a translation model, which will contain translations of English words to both Chinese bigrams and unigrams. This strategy is similar to BU in monolingual IR. So we will denote the translation method by BU, too.

(2) We can also create two separate translation models, one for bigrams and another for unigrams. Then the two translation models can be combined linearly by setting a relative importance to each of them, in a similar way to Formula (4). We denote this approach by B+U.

Now we show how these TM are used for our CLIR. Firstly, the translation candidates with low probabilities usually are not strongly related to the query. They are more noise than useful terms. So, we remove them by setting a threshold δ (set at 0.001 in our experiments): we filter out the items t_i with $t(t_i \mid s_j) < \delta$. Then, the probabilities of the remaining translation candidates are re-normalized so that $\sum_{t_i} t(t_i \mid s_j) = 1$.

Then, we calculate the query model by Formula (5). To further reduce noise, we use one of the following two methods to select translations:

(1) For each source term s_j, we only use the top N best translations, and use all of resulting t_i (together with their probabilities) as query.

(2) We can also rely on $p(t_i \mid \theta_{Q_s})$ and select the top N*$|Q_s|$ translation candidates t_i as query translation, where N is a fixed parameter we can tune manually, and $|Q_s|$ the length of the query.

3.3 Combine bilingual dictionary and parallel corpus

Combining a translation model with a bilingual dictionary can greatly improve the CLIR effectiveness [9]. Our experiments show that treating a bilingual dictionary as a parallel text and using the trained TM from it is much better than directly using the items of a dictionary. Therefore, we regard each entry in a bilingual dictionary as a "parallel sentence", and a bilingual dictionary is then considered as another "parallel corpus". Then, we train a series of TMs from the dictionary: U, B, W, and BU.

In this way, as a dictionary is transformed into a TM, it can be combined with a parallel corpus by combining their TMs. An equivalent way is to combine the retrieval scores resulted from using different TMs, as follows:

$$Score_{MC+MD}(D,Q) = \lambda Score_{MC}(D,Q) + (1-\lambda) Score_{MD}(D,Q) \qquad (6)$$

where MC and MD are TMs trained from the parallel corpus and the bilingual dictionary respectively.

4 Result and Experiment

4.1 Monolingual IR

In NTCIR 6, both documents and topics are from previous NTCIR experiments. The characteristics of documents and queries are described in Table 1 and Table 2.

Tabel 1: Description of collections of NTCIR

	NTCIR3/4		NTCIR5/6	
	Collections	#doc (K)	Collections	#doc (K)
Cn	CIRB011 CIRB020	381	CIRB040r	901
Jp	Mainichi98/99 Yomiuri98+99	594	Mainichi00/01r Yomiuri00+01	858
Kr	Chosunilbo98/99 Hankookilbo	254	Chosunilbo00/01 Hankookilbo00/01	220

Tabel 2: Topic numbers

For all languages (Chinese, Japanese, Korean, and English)				
	NTCIR3	NTCIR4	NTCIR5	NTCIR6
#topics	50	60	50	50

As our basic retrieval tool, we use Lemur toolkit[1] with KL-divergence and Dirichlet prior smoothing

[1] http://www.lemurproject.org

method (with the default setting in Lemur). We evaluated the results of IR by TREC_EVAL.

Tables 3 give the Chinese, Japanese, and Korean monolingual retrieval results for previous NTCIR queries. All of our results are measured in MAP (Means Average Precision). We use bold to indicate the best results. We tested the parameter λ of Formula (4) from 0.1 to 0.9. The results show that λ=0.3 gives the best performance. We notice that interpolating unigram and bigram (B+U) has the best performance for Chinese and Japanese. However, BU and B are best for Korean.

Table 3: The results of using different index units for C/J/K monolingual IR on NTCIR4/5 data

Run-id	Means Average Precision (MAP)											
	U		B		W		BU		WU		0.3B+0.7U	
	Rigid	Relax	Rigid	Relax	Rigid	Relax	Rigid	Relax	Rigid	Relax	Rigid	Relax
C-C-T-N4	.1929	.2370	.1670	.2065	.1679	.2131	.1928	.2363	.1817	.2269	**.1979**	**.2455**
C-C-D-N4	.1491	.1943	.1367	.1789	.1503	.1925	.1555	.1983	.1515	.1953	**.1735**	**.2228**
C-C-T-N5	**.3302**	.3589	.2713	.3300	.2676	.3315	.2974	.3554	.3017	.3537	.3300	**.3766**
C-C-D-N5	.2608	.3114	.2156	.2779	.2339	.2899	.2492	.3093	.2516	3008	**.2811**	**.3369**
J-J-T-N4	.2377	.2899	.2768	.3670	–	–	.2807	**.3722**	–	–	**.2873**	.3664
J-J-D-N4	.2089	.2632	.2216	.3022	–	–	.2287	.3113	–	–	**.2521**	**.3301**
J-J-T-N5	.2376	.2730	.2471	.3273	–	–	.2705	.3458	–	–	**.2900**	**.3495**
J-J-D-N5	.2106	.2649	.1900	.2534	–	–	.1906	.2563	–	–	**.2380**	**.2989**
K-K-T-N4	.2004	.2147	.3873	.4195	–	–	**.4084**	**.4396**	–	–	.3608	.3889
K-K-D-N4	.1652	.1745	.3244	**.3498**	–	–	**.3248**	.3488	–	–	.3213	.3441
K-K-T-N5	.2603	.2777	.3699	.3996	–	–	**.3865**	**.4178**	–	–	.3800	.4001
K-K-D-N5	.2211	.2463	**.3501**	**.3903**	–	–	.3340	.3726	–	–	.3396	.3787

Table 4: The results of CJK monolingual IR in NTCIR6
(We use model 0.3B+0.7U for C-C and K-K, UB for K-K-T, B for K-K-D.
The columns shadowed are our officially submittions at Stage 1)

Run-id	RALI without pseudo feedback (MAP)		RALI with pseudo feedback (MAP)		Average MAP of all NTCIR6 runs	
	Rigid	Relax	Rigid	Relax	Rigid	Relax
C-C-T	.2139	.3022	.2330	.3303	.2269	.3141
C-C-D	.1671	.2376	.2031	.2907	.2354	.3294
J-J-T	.2426	.3171	.2576	.3343	.2707	.3427
J-J-D	.1877	.2485	.2292	.3052	.2480	.3214
K-K-T	.3332	.3939	.3460	.4130	.3833	.4644
K-K-D	.2623	.2970	.3287	.3945	.3892	.4678

Table 5: English to Chinese CLIR result on NTCIR 3/4/5 data
(The cells shadowed are our officially submittions at Stage 2)

Run-id		Means Average Precision (MAP)									
		U		B		W		BU		0.3B+0.7U	
		Rigid	Relax	Rigid	Relax	Rigid	Relax	Rigid	Relax	Rigid	Relax
E-C-T-N3	Corpus	.0910	.1079	.0798	.0860	.0881	.1086	.0874	.0983	.1020	.1132
	Dict.	.0691	.0768	.0449	.0530	.0434	.0509	.0652	.0776	.0669	.0767
RALI-E-C-T01-N3	Int(C,D)	.0928	.1106	.0805	.0985	.0898	.1080	.0938	.1102	**.1021**	**.1170**
E-C-D-N3	Corpus	.0894	.1091	.0969	.1147	.1099	.1248	.1099	.1249	**.1276**	.1370
	Dict.	.0526	.0680	.0354	.0482	.0625	.0693	.0553	.0775	.0628	.0829
RALI-E-C-D01-N3	Int(C,D)	.0900	.1149	.1037	.1333	.1163	.1315	.1116	.1370	.1226	**.1439**
E-C-T-N4	Corpus	.0854	.0962	.0698	.0804	.0569	.0703	.0907	.1066	.0918	.1063
	Dict.	.0733	.0862	.0488	.0548	.0587	.0712	.0748	.0870	.0805	.0955
RALI-E-C-T01-N4	Int(C,D)	.0935	.1060	.0872	.1004	.0746	.0897	**.1042**	**.1194**	.1018	.1180
E-C-D-N4	Corpus	.0771	.0898	.0576	.0749	.0574	.0735	.0806	.0981	.0849	.1034
	Dict.	.0700	.0739	.0448	.0482	.0560	.0647	.0683	.0723	.0764	.0827
RALI-E-C-D01-N4	Int(C,D)	.0921	.1021	.0774	.0897	.0727	.0893	.0935	.1076	**.1017**	**.1173**
E-C-T-N5	Corpus	.1500	.1686	.1072	.1306	.1134	.1305	.1527	.1852	.1473	.1751
	Dict.	.1161	.1187	.0613	.0694	.0830	.0898	.0932	.0910	.1092	.1090
RALI-E-C-T01-N5	Int(C,D)	.1533	.1727	.1245	.1512	.1317	.1566	.1632	**.1970**	**.1655**	.1916
E-C-D-N5	Corpus	.1456	.1558	.0983	.1172	.0818	.0943	.1506	.1743	.1492	.1655
	Dict.	.0994	.1140	.0472	.0550	.0929	.1114	.0800	.0920	.1139	.1253
RALI-E-C-D01-N5	Int(C,D)	.1676	.1792	.1158	.1369	.1254	.1492	.1629	.1844	**.1776**	**.1946**

At Stage 1 of NTCIR6, we submitted the monolingual IR results of Chinese, Japanese, and Korean. For Chinese and Japanese, we submitted the results produced by interpolating unigrams and bigrams (B+U); for Korean, we submitted the result using BU index (RALI-K-K-T) for title-only queries; and the results using bigrams for description and title+description queries (RALI-K-K-D, RALI-K-K-TD). In Table 4, we compare the result we submitted to NTCIR6. We noticed that our results are lower than average MAPs of NTCIR6. This is due to the fact that we only tried to compare index units and used the basic IR technique. After apply a simple pseudo relevance feedback (by our experiment, we set the number of document and term to 20 and 80 respectively), the results become more comparable to average MAPs.

4.2 English to Chinese CLIR experiments

Our model requires a set of parallel texts to train a TM. We have implemented an automatic mining tool to mine Chinese-English parallel texts from Web. It uses a similar approach to [2]. The parallel texts are from six websites, which are located in United Nations, Hong Kong, Taiwan, and Mainland China (Chinese pages encode in GB2312, Big5, and Unicode). It contains about 4 000 pairs of pages.

After converting the HTML texts to plaint texts and mark the paragraph and sentence boundaries, we use a sentence alignment algorithm to align the parallel texts to sentence pairs. Our sentence alignment algorithm is an extension of the length-based method [3], which also considers the lexical-translation according to a bilingual dictionary. The idea is that if a pair of sentences contains many words that are mutual translations in the dictionary, then their alignment score should be high. Here we use CEDICT[1], which contains 28,000 Chinese items, to recognize the word translations between parallel sentences. After sentence alignment, we obtain 281,000 parallel sentence pairs.

For query translation, we use two resources: the parallel corpus and a bilingual dictionary. Similarly to parallel corpus, we train a series of TMs (B, U, W, and BU) from a bilingual dictionary, which merges Chinese-to-English Wordlist and English-to-Chinese Wordlist (version 1.0) from LDC [2]. The final dictionary contains 42,000 entries, which are treated as parallel sentences in TM training.

To translate English topic to Chinese, we select the top 6 translations of TM for each English source term. This number produced good results in our previous tests.

We tested different values for λ to combine parallel corpus and dictionary, ranging from 0.1 to 0.9. The result show *Int(C,D) = 0. *Corpus+0.3*Dict* is the best.

In Table 5, *Int(C,D)* is the interpolating of TMs from Corpus and Dictionary (by Formula (6)). The last columns of *Int(C,D)* are result of we submitted, which are from two step interpolations:

(1) Interpolating the retrieval scores obtained with bigram and unigram models trained on the parallel corpus using Formula (4); the same for the models trained on the bilingual dictionary.
(2) Interpolating the scores obtained in step 1 by Formula (6). In both interpolations, λ is set at 0.3.

For Chinese, the experiments of monolingual IR show that indexing by unigrams is comparable to using words, and usually better then bigram. The CLIR results show that using bigram and unigram as translation units is a reasonable alternative to words. Combinations of bigram and unigram usually produce higher effectiveness for both monolingual IR and CLIR. In particular, the best combination seems to be 0.3B+0.7U, i.e. the final ranking score is produced by Formula (4) with λ=0.3. Similar to Stage 1, the results we submitted at Stage 2 without using pseudo relevance feedback.

Due to lack of Japanese and Korean resources, we could not compare the results of n-gram to word for these languages. Between unigrams and bigrams, our experiments show that bigram is slightly better than unigrams for Japanese, and bigram is much better for Korean. The combination of bigram and unigram is also more effective for Japanese, but not necessarily for Korean.

5 Conclusion and Future Work

In our experiments in NTCIR6, we have focused on the comparison between words and n-grams, as well as their combinations, both for indexing and for query translation. Our experimental results with previous NTCIR queries show that n-grams as generally as effective as words for monolingual IR in Chinese. Different types of n-gram can be combined in different ways: they can be segmented simultaneously in the Chinese, Japanese and Korean texts, and used at the same time as indexes of the documents. This mixture approach has been used in several previous experiments [8]. In addition to this approach, we also tested the alternative of creating different types of index separately, then grouping them during the retrieval process. We found that this second approach is slightly more effective for Chinese and Japanese. This approach also has the flexibility of setting appropriate relative importance between different types of index, which the first mixture approach does not have.

For query translation, we have tested the utilization of Chinese n-grams as translation units. Similarly to monolingual IR, unigrams and bigrams can be

[1] http://www.mandarintools.com/cedict.html
[2] http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm

simultaneously segmented in Chinese sentences, so they are mixed up the same translation model; or we can create two separate translation models for unigrams and bigrams. Our experiments also show that the second approach is slightly better.

The overall conclusion of our experiments is that n-grams can be interesting alternative indexing and translation units to words. For the purpose of IR, we do not necessarily have to segment Chinese, Japanese and Korean texts into words. It is possible to use n-grams to represent them.

References

[1] P. F. Brown, S. A.D. Pietra, V. J.D. Pietra, and R. L. Mercer. "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, 19(2):263-311, 1993.

[2] J. Chen and J.Y. Nie. "Automatic construction of parallel English-Chinese corpus for cross-language information retrieval," *ANLP*, Seattle, Washington, 2000, pp.21-28.

[3] W. A. Gale, and K. W. Church. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics,* 19(1):75-102, 1993.

[4] K.L. Kwok. "Comparing representations in Chinese information retrieval," *SIGIR 199* : 34-41.

[5] W. Kraaij, J.Y. Nie, and M. Simard. "Embedding Web-based statistical translation models in cross-language information retrieval," *Computational Linguistics*, 29(3):381-419, 2003.

[6] R.W.P. Luk, K.F. Wong, and K.L. Kwok. "A comparison of Chinese document indexing strategies and retrieval models", *ACM Trans. Asian Lang. Inf. Process*, 1(3): 225-268, 2002.

[7] J.Y. Nie, M. Brisebois, and X. Ren. "On Chinese text retrieval," *SIGIR 1996*, pp.225-233.

[8] J.Y. Nie, J. Gao, J. Zhang, and Zhou, M. "On the use of words and n-grams for Chinese information retrieval," *In Fifth International Workshop on Information Retrieval with Asian Languages, IRAL-2000*, pp.141-148.

[9] J.Y. Nie, M. Simard. 2001. Using Statistical Translation Models for Bilingual IR. *CLEF 2001*, pp. 137-150.

[10] J. Ponte and W.B. Croft. "A language modeling approach to information retrieval," *SIGIR 1998*, pp.275-281.