Modeling Variable Dependencies between Characters in Chinese Information Retrieval

Lixin Shi and Jian-Yun Nie

DIRO, University of Montreal
CP. 6128, succursale Centre-ville, Montreal, H3C 3J7 Quebec, Canada
{shilixin,nie}@iro.umontral.ca

Abstract. Chinese IR can work on words and/or character n-grams. In previous studies, when several types of index are used, independence is usually assumed between them, which obviously is not true in reality. In this paper, we propose a model for Chinese IR that integrates different types of dependency between Chinese characters. The role of a pair of dependent characters in the matching process is variable, depending on the pair's ability to describe the underlying meaning and to retrieve relevant documents. The weight of the pair is learnt using SVM. Our experiments on TREC and NTCIR Chinese collections show that our model can significantly outperform most existing approaches. The results confirm the necessity to integrate dependent pairs of characters in Chinese IR and to use them according to their possible contribution to IR.

Keywords: Term dependency, Dependency weight, Dependency language model, Discriminative model, Chinese IR.

1 Introduction

A crucial problem in Chinese Information Retrieval (IR) is to determine the appropriate elements to serve as index. Two general families of approaches have been proposed in the literature: using characters (mainly character unigrams and bigrams) and using words. It has been found in several studies that it is beneficial to combine different types of index [5, 6, 11]. Indeed, while a word can represent precisely a meaning, the meaning can also be expressed by other words and characters. In Chinese, related words often share some characters. Therefore, character unigrams and bigrams can be used as a means to perform partial matching between them.

However, the previous approaches usually assumed that different types of index are independent. Typically, each type of index is considered as forming a distinct representation space from other types. The typical approach to Chinese IR determines a matching score according to each type of representation, and the final score is an interpolation of these scores. It is obvious that the assumption of independence between different indexes does not hold in reality. For example, the word or bigram 植树 (tree planting, forestation) is strongly dependent on the characters 植 (planting) and 树 (tree). Some studies have tried to deal with the relationships between different indexes. For example, Shi et al. [13] considers that longer and shorter words, as well as the constituent characters, are strongly

P.-J. Cheng et al. (Eds.): AIRS 2010, LNCS 6458, pp. 539–551, 2010.

dependent due to their overlapping. However, the relationship between them is simply determined by the scale of the overlap, which does not necessarily reflect the true strength of relationship. In general IR, several models have also been proposed to capture dependencies between terms [4, 8, 9]. Usually, a dependence model is defined in addition to the traditional bag-of-words or word unigram model. Each of these models is assigned a fixed weight in the final combination. However, in reality, term dependencies do not have equal importance in IR. Some dependencies (or word groups) are mandatory to consider because the constituent words without their dependency could mean different things (e.g. "hot dog") and they would retrieve off-topic documents; while some other dependencies are moderately useful (e.g. "text printing") because the constituent words separately can represent the same meaning equally well and they can retrieve a similar set of documents.

The necessity to consider dependencies in Chinese IR is even higher. Indeed, a Chinese sentence is basically constructed from characters, which act as words or part of words. Characters can be strongly dependent. The dependencies between characters can be more or less useful for IR, depending on whether the meaning can be expressed by separate characters. If the meaning can be expressed well by separate characters, then the need to consider the dependency between the characters is low. This is the case for 房屋 (house). Indeed, the characters 房 and 屋 alone can also express the same meaning. The dependency between them does not provide much additional information to IR. On the other hand, in an expression such as 京九铁路 (Beijing-Kowloon Railroad), it is important to capture the dependency between 京 (Beijing, capital) and 九 (nine) because these characters are very ambiguous and they would not mean Beijing-Kowloon when considered separately. The dependency between them in this expression helps determine a specific meaning and retrieve documents on a specific topic. Therefore, this dependency is important to capture.

The above examples illustrate the variable necessity to take into account the dependency between a pair of Chinese characters according to how strong the dependency is and how useful it is for IR. In the first example, even if there is some dependency between 房 and 屋, the dependency should play a limited role. On the other hand, the dependency between 京 and 九 in the second example should be assigned a much higher importance. This aspect has not been considered in the previous IR models. In this paper, we will define a model capable of coping with this problem. We will consider characters as forming the basic index for Chinese IR. Then dependencies between pairs of characters are incorporated with variable weights depending on the usefulness of considering the dependency. Several types of dependency will be considered: dependency between adjacent characters and dependency between co-occurring characters at different distances. For example, the characters in the word 铁路 (railway) form a dependency between adjacent characters. Co-occurring characters within a sentence at some distance such as 世 (world, age, era) and 贸 (trade) in 世界贸易 (world trade) is also important to capture. In particular, this helps account for the many abbreviations often used in Chinese, which are usually formed by co-occurring, non-adjacent, characters.

In this paper, we use SVM to determine the appropriate weight of a pair of characters according to a set of features. The experiments on several TREC and NTCIR

collections show that our model can significantly outperform the previous independent models and dependency models using fixed weights for types of index.

The remaining of the paper is organized as follows. In the next section, we review some related studies on Chinese IR and term dependency models. Then, we will describe our character-based dependency model and our parameter learning method. In Section 4, we present the experiments on TREC and NTCIR collections, and finally conclude our work in Section 5.

2 Related Work

A number of studies have investigated the effectiveness of Chinese IR using either or both characters and words. It is found that approaches using either characters (bigrams) or words can lead to comparable retrieval effectiveness [6, 11, 12]. In [14], it is further found that the retrieval effectiveness using a character-based language model is highly competitive to, and on several collections, is even higher than that using words and bigrams. This result motivates our use of characters as our basic index units in this study.

Within the language modeling framework, given a vocabulary V, the score of a document D to a query Q can be determined according to the following equation [2]:

$$Score(D,Q) = \sum_{w \in V} P(w|\theta_Q) \log P(w|\theta_D)$$

where θ_Q and θ_D are respectively the language model for the query Q and the document D. The vocabulary V in the model can be unigrams (U), bigrams (B), words (W), or a mixture of them. A general way to combine different types of index is to determine a score according to each type of index and then to combine the scores as follows:

$$Score(D,Q) = \sum_{R} \lambda_R Score_R(D,Q)$$

where R denote a type of index, which could be U, B and W.

We notice that in such a combination, no relationship between different types of index is considered. In addition, a fixed global weight is assigned to each type of index regardless to the strength of dependency between a pair of characters. As we argued earlier, this is counterintuitive: some dependencies should be attributed a higher weight than others in the matching process because of their ability to express unambiguously a meaning and to retrieve the required documents, in comparison to characters.

Term dependency is a general phenomenon present in all the languages. Several attempts have been made to define IR models capable of capturing term dependencies. Gao et al. [4] proposed a dependency model in which term dependency is captured by a biterm model. The final model combines the unigram model and the biterm model. Metzler and Croft [8] proposed Markov Random Field (MRF) models for IR, in which dependencies between terms in the same clique (a set of fully connected nodes) are considered. In the full dependence model (MRF-FD), all the terms in a sentence (query) are assumed to be dependent. This will lead quickly to the problem of complexity when

the number of terms in a clique becomes large. To limit the complexity, a sequential dependence model (MRF-SD) is used, in which only dependencies between adjacent terms are considered. In addition to unigrams, two types of dependence are considered: ordered and un-ordered. These types of dependence are assigned fixed weights $(\lambda_U, \lambda_O, \lambda_U)$ in the final score function.

As we mentioned earlier, dependencies between non adjacent characters are also important in Chinese IR, and the role of each pair of dependent characters in the matching process varies. To deal with the last problem, Bendersky et al. [1] extended recently the MRF-SD model to a weighted MRF-SD model (which we denote by WSD), in which the weight of a term and a pair of terms becomes dependent on the individual term and pair of terms. The scoring function is as follows:

$$P(D|Q) \overset{rank}{=} \sum_{q_i \in Q} \lambda(q_i) f_T(q_i, D) + \sum_{q_i q_{i+1} \in Q} \lambda(q_i q_{i+1}) [f_O(q_i q_{i+1}|D) + f_U(q_i q_{i+1}|D)]$$

in which $\lambda(q_i) = \sum_{j=1}^{k_{uni}} w_j^{uni} g_j^{uni}(q_i)$ and $\lambda(q_i q_{i+1}) = \sum_{j=1}^{k_{bi}} w_j^{bi} g_j^{bi}(q_i q_{i+1})$ are the importance of the unigram q_i and bigram $q_i q_{i+1}$ respectively, the function $g_j(\cdot)$ is a feature defined over unigrams or bigrams, and w_j is its weight, a free parameter to be estimated. This goes in the same direction as our model, i.e. to assign variable weights to unigrams and pairs of terms. However, the relationship between non-adjacent query terms is still ignored in [1] and the ordered and un-ordered pairs of terms are treated in the same way. Our model will go a step further: we will consider dependencies between non-adjacent characters as between 世 and 贸 in 世界贸易 (world trade). We will also separate ordered and unordered pairs of characters.

3 Our Method

3.1 Variable Dependency Model

MRF models are limited due to its high complexity. It is difficult to extend them to cover dependencies between distant characters. On the other hand, discriminative models have the advantage that one can selectively use a subset of useful dependencies as features rather than all the dependencies [10]. Discriminative models have been successfully used in IR [3]. A typical discriminative model corresponds to the following equation:

$$score(D, Q) = P(Rel|D, Q) = \frac{1}{Z} \exp\left(\sum_{i}^{n} \lambda_i f_i(Q, D)\right) \quad (1)$$

where $f_i(Q, D)$ is a feature function with weight λ_i and Z a normalization constant. The model we propose is a discriminative model. We limit the dependencies to pairwise dependencies, which often correspond to the strongest dependencies that we want to capture. The flexibility of discriminative models allows us to consider dependencies between more distant characters, without having to increase the complexity of the model to account for more complex and less useful dependencies.

The previous experiments show that the LM based on character unigrams works well [14]. Therefore, we use Chinese characters as our basic index units. In addition, we

consider the following types of dependencies: (1) bigram, (2) unordered co-occurring characters within some distance. Let us use C_w to denote the character co-occurrence within a window of size w in documents. To express the proximity of co-occurring characters in documents, we use a set of window sizes W (in our implementation, we use 3 window sizes: 2, 4, and 8) when we construct document models. The idea of using windows of different sizes for documents is to try to capture the strength of proximity between characters: intuitively, a pair of closer characters has a stronger relationship. The ranking function is extended from Equation (1) to the following one:

$$\begin{aligned} P(Rel|D,Q) \overset{rank}{=} & \sum_{q_i \in Q} \lambda_U(q_i|Q) f_U(q_i, D) \\ & + \sum_{q_i q_{i+1} \in Q} \lambda_B(q_i, q_{i+1}|Q) f_B(q_i q_{i+1}, D) \\ & + \sum_{w \in W} \sum_{q_i, q_j \in Q, i \neq j} \lambda_{C_w}(q_i, q_j|Q) f_{C_w}(q_i, q_j, D) \end{aligned}$$

This model contains three classes of features: unigram features $f_U(q_i, D)$, bigram features $f_B(q_i q_{i+1}, D)$ and co-occurrence features $f_{C_w}(q_i, q_j, D)$ where w is the document window size. The characters q_i and q_j used in the above ranking function are any pair of (not necessarily adjacent) characters in a query. Each feature is associated with a function λ denoting the importance of the feature for the query Q. This function allows us to take into account the variable dependencies between bigrams and co-occurring characters according to their strength and utility. The discriminative feature functions we use are simply the language model features defined as follows:

$$f_U(q_i, D) = P_U(q_i|Q) \log P_U(q_i|D)$$

$$f_B(q_i q_{i+1}, D) = P_B(q_i q_{i+1}|Q) \log P_B(q_i q_{i+1}|D)$$

$$f_{C_w}(q_i, q_j, D) = P_{C_w}(\{q_i . q_j\}_w|Q) \log P_{C_w}(\{q_i . q_j\}_w|D)$$

where $\{q_i . q_j\}_w$ denote a pair of co-occurring characters q_i and q_j in document within a window of size w. The features are defined in this way in order to make it easier to compare our model with other approaches using language modeling. However, tone can well define other features.

For the ranking purpose, we will simply fix $\lambda_U(q_i|Q)$ at the constant 1, and try to vary the other λ functions for bigrams and co-occurring terms. Putting all together, we have the following final model:

$$\begin{aligned} P(Rel|D,Q) \overset{rank}{=} & \sum_{q_i \in Q} P_U^{ml}(q_i|Q) \log P_U(q_i|D) \\ & + \sum_{q_i q_{i+1} \in Q} \lambda_B(q_i, q_{i+1}|Q) P_B^{ml}(q_i q_{i+1}|Q) \log P_B(q_i q_{i+1}|D) \\ & + \sum_{w \in W} \sum_{\substack{q_i, q_j \in Q \\ i \neq j}} \lambda_{C_w}(q_i, q_j|Q) P_C^{ml}(\{q_i . q_j\}|Q) \log P_{C_w}(\{q_i . q_j\}_w|D) \end{aligned} \quad (2)$$

This model will be called Variable Dependency Model (VDM). The fundamental difference of our model with most of previous models is that the λ functions are now dependent on the specific pair of characters. For example, for 世界贸易 (world trade), the model will capture the relations between the following character pairs: 世界, 世贸, 世易, 界贸, …, 贸易. The importance of each pair varies depending on the usefulness to consider it in IR. The weight will be learnt using SVM (see Section 3.2).

For the query models in Equation (2), we will simply use Maximum Likelihood estimation as follows, where t_R is a unigram, a bigram or a pair of co-occurring terms and $c(t_R; Q)$ its count in the query:

$$P_R^{ml}(t_R|Q) = \frac{c(t_R; Q)}{|Q|_R}, \quad R \in \{U, B, C_2, C_4, C_8\}$$

For the document model, Dirichlet smoothing is used:

$$P_R(t_R|D) = \frac{c(t_R; D) + \mu_R \cdot P_R(t_R|C)}{|D|_R + \mu_R}$$

where $c(t_R; D)$ is the count of term t_R in document D (within a window for C_w); $P_R(t_R|C) = \frac{\sum_{D \in C} c(t_R; D)}{\sum_{D \in C} |D|_R}$ is the collection language model; μ_R is a Dirichlet prior for the corresponding type of model; and $|D|_R$ is the document length in the expression of R, i.e. the total number of unigrams, bigrams or co-occurring terms within the corresponding window size.

3.2 Parameter Estimation

The λ functions are determined according to the strength and utility of bigrams or co-occurring terms in IR, in comparison to characters. We will use the epsilon Support Vector Machine Regression (ϵ-SVR) [15] method to train $\lambda_R(\cdot)$. The toolkit LIBSVM[1] is used for this purpose.

The training examples are obtained as follows. For each training query, we first find the best weights (z_i) for each bigram or pair of co-occurring characters within different windows (x_i). We use a coordinate-level ascent search algorithm [7] to find the best weight for each x_i. A set $\mathbf{x}_i$ of features is used to characterize x_i. This defines a learning example $(\mathbf{x}_i, z_i)$.

In our experiment, we use 10-fold cross-validation method, i.e., 1/10 of the data $(\mathbf{x}_i, z_i)$ will be used in turn as the test data for IR while the remaining 9/10 will be used as the training data for parameter learning. In this study, we use the following features (where x is a bigram or a co-occurring character pair, and q_i, q_j are characters in x):

- Point-wise mutual information in an independent text collection: $\mathrm{PMI\_all}(x)$. We use the concatenation of all test collections as the independent collection.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

- PMI in the current test collection: $\mathrm{PMI\_coll}(x)$
- A binary value according to the test: $\mathrm{PMI\_all}(x)$ > Threshold? (we set the[2] threshold to 0 in our experiments)
- Binary test value: $\mathrm{PMI\_coll}(x)$> Threshold?
- $\mathrm{idf}(x) - \mathrm{idf}(q_i) - \mathrm{idf}(q_j)$
- $(\mathrm{idf}(x) - \mathrm{idf}(q_i) - \mathrm{idf}(q_j))/(\mathrm{idf}(q_i) + \mathrm{idf}(q_j))$
- $\mathrm{count}(x, \mathrm{coll})/\min(\mathrm{count}(q_i, \mathrm{coll}), \mathrm{count}(q_j, \mathrm{coll})$
- $\mathrm{count}(x, \mathrm{coll})/\max(\mathrm{count}(q_i, \mathrm{coll}), \mathrm{count}(q_j, \mathrm{coll})$
- A binary value according to whether x appears in a Wikipedia Chinese title?
- The distance between q_i and q_j in the query (for co-occurring character pair)

In addition, for a bigram b_i, we also use the following additional feature, which determines the proportion of document in which one of the characters only appears in the bigram:

$$\frac{\left|\{D \left| c(b_i; D) > 1 \;\&\; c(b_i; D) = min\left(c(q_i; D), c(q_{i+1}; D)\right)\right\}\right|}{|\{D|c(b_i; D) > 1\}|}$$

We have not defined a large set of features because our primary goal of this study is to show the importance to incorporate dependencies between characters at variable degrees. The set of features could be easily extended in the future.

4 Experiments

4.1 Test Collections

We use the collections from TREC and NTCIR. The characteristics of the collections are summarized in Table 1.

Table 1. Characteristics of collections

Collection	Name	#Document	Size (MB)	Avg. Doc Length
TR56	Peoples Daily	164,788	173	158
TR9	Xinhua news agency	127,938	86	205
NT34	CIRB011&CIRB020	381,681	543	226
NT56	CIRB040r	901,446	1106	207

In our experiments, we use topic titles as our queries (see Table 2). This choice is made to better correspond to real queries on search engines.

[2] http://download.wikimedia.org/chwiki/, which includes 338,164 titles.

Table 2. Characteristics of queries

Query Set	Collection	Queries	#Query	Avg. len. (in character)
TREC5	TR56	CH1-28	28	4.7
TREC6	TR56	CH29-54	26	4.7
TREC9	TR9	CH55-79	25	3.7
NTCIR4	NT34	001-060	59	4.3
NTCIR5	NT56	001-050	50	4.6
NTCIR6	NT56	003-110	50	3.9

4.2 Pre-processing and Indexing

As the collections are in different coding schemas, we converted all the characters into GB codes. To compare to the word-based method, we use a word segmentation tool ICTCLAS[2] to segment Chinese texts to words, and use another segmentation program from LDC[3] to further segment long words into short words. For example, the long word 世界贸易组织 (World Trade Organization) will be further segmented in the second step into its constituent short words: 世界 (World), 贸易 (Trade), 组织 (Organization). The previous experiments showed that short words perform better than long words [5].

We use Indri[4] to build the index for our model. The basic index units are Chinese characters (which we denote by U). To compare to the baseline models, we also build the indexes for other index units: words (W), bigrams (B), words and bigrams combined with unigrams (WU, W+U, BU and B+U). In the combinations WU and BU, all types of index are indexed together, while in W+U and B+U, they are indexed separately and the scores from each index are combined linearly.

4.3 Experimental Results

We first provide the retrieval results of the baseline methods in Table 3. The combination parameters in W+U and B+U are tuned to their best. For a Chinese query $q_1 q_2 \ldots q_n$, we assume the word segmentation result to be $w_1, w_2, \ldots w_m$. The baseline models are listed below:

- U: We use unigrams of character, and the query is "$q_1 q_2 \ldots q_n$".
- B: We use bigrams of characters. The corresponding Indri query is "$\#1(q_1\ q_2)\ \ldots\ \#1(q_{n-1} q_n)$".
- BU: We use both bigrams and unigrams mixed up in a single query. The Indri query is "$\#1(q_1\ q_2)\ \ldots\ \#1(q_{n-1} q_n)\ q_1\ q_2 \ldots q_n$".
- B+U: of the scores using B and U are interpolated.
- W: We use segmented words. The query is "$w_1\ w_2 \ldots w_m$".
- WU: The segmented words are mixed up with character unigrams. The Indri query is "$w_1\ w_2 \ldots w_m\ q_{i1}, q_{i2} \ldots$".
- W+U: The scores using W and U are interpolated.

[2] http://ictclas.org/
[3] http://www.ldc.upenn.edu/Projects/Chinese/seg.zip
[4] http://www.lemurproject.org/indri/

Table 3. The baselines (MAP) of traditional Chinese IR models

Query	U	B	BU	B+U	W	WU	W+U
Trec5	0.3013	0.2696	0.3184	**0.3269**	0.2802	0.3265	0.3173
Trec6	0.3601	0.3610	0.3875	0.3878	0.3881	0.3983	**0.3998**
Trec9	0.2381	0.2119	0.2469	**0.2543**	0.1905	0.2283	0.2381
Ntcir4	0.2371	0.1995	0.2243	**0.2489**	0.2237	0.2396	0.2469
Ntcir5	0.3587	0.3151	0.3563	0.3681	0.3840	0.3817	**0.3998**
Ntcir6	0.2695	0.2448	0.2931	**0.3064**	0.2739	0.2863	0.3012

To see the importance of different type of index, we plot the results of the methods B+U and W+U on Trec6 and Ntcir6 collections in Figure 2. We can see that a reasonable interpolation usually leads to a higher effectiveness than using only one type of index (the two extremities of the curves). This shows that different types of index are complementary and it is useful to combine them. However, the best weight for each type of index depends on the collection and on the types of index combined. Indeed, the usefulness of different words and bigrams varies largely. The weight we assign to a type of index corresponds to a compromise among all the words and bigrams. As we will see in the experiment with our proposed model, it is better to assign a different weight to a word or a bigram depending on its usefulness.

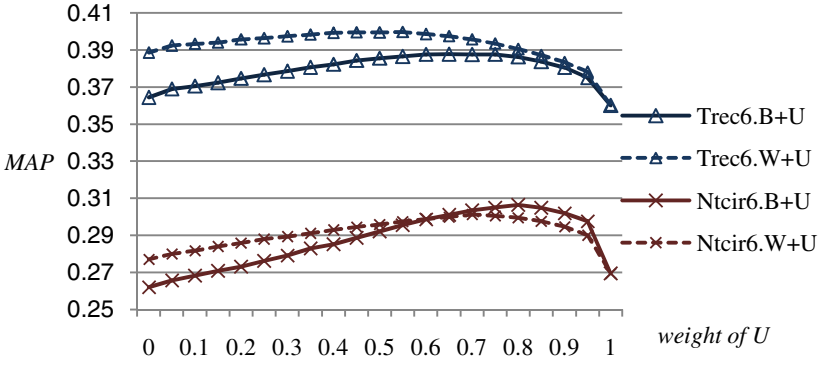

Fig. 1. Compare the MAP of B, U, W, and their interpolations on Trec6 and Ntcir6 Collections

In Table 4, we show the effectiveness with other baselines - MRF-SD and Weighted MRF-SD (WSD). For MRF-SD, we use a grid search to find the best parameters $\lambda_T, \lambda_O, \lambda_U$ so as to maximize MAP for each collection. Therefore, the effectiveness of this model is tuned to its best. The results with MRF-SD are slightly better than B+U. Indeed, if we remove the unordered part, the MRF-SD becomes identical to B+U. The difference between MRF-SD and B+U corresponds to the contribution of unordered unigram pairs. The WSD model is slight better than MRF-SD except on Trec6. However, the differences between the two models are not statistically significant.

Table 4. The baselines of dependency models: MRF-SD and WSD. † and ‡ means statistically significant difference at the level of p<0.05 and p<0.01.

Query	MRF-SD				Weighted-SD		
	MAP	**%U**	**%B+U**	**%W+U**	**MAP**	**%U**	**%SD**
Trec5	0.3271	+8.6‡	+3.1	+3.1	0.3279	+8.8‡	+0.2
Trec6	0.3899	+8.3‡	+0.6	-2.5	0.3780	+5.0	-3.1
Trec9	0.2576	+8.2	+1.3	+8.2	0.2732	+14.8†	+6.0
Ntcir4	0.2490	+5.0†	+0.0	+0.8	0.2514	+6.0‡	+1.0
Ntcir5	0.3846	+7.2 ‡	+4.5	-3.8	0.3909	+9.0†	+1.6
Ntcir6	0.3066	+13.8‡	+0.0	+1.8	0.3088	+14.6‡	+0.7

Table 5. The results (MAP) with Variable Dependency Model (VDM)

Query	VDM (best fixed)			VDM (10-fold cross-validation)							VDM (*ideal*)
	MAP	**%U**	**%SD**	**MAP**	**%U**	**% B+U**	**% W+U**	**% SD**	**% WSD**	**% fixed**	**MAP**
Trec5	0.3278	+8.8‡	+0.2	0.3501	+16.2‡	+7.1‡	+10.4‡	+7.1‡	+6.8†	+6.8‡	*0.4414*
Trec6	0.3916	+8.7‡	+0.4	0.4159	+15.5‡	+7.3‡	+4.0†	+6.7‡	+10.0‡	+6.2‡	*0.5272*
Trec9	0.2627	+10.4	+2.0	0.2713	+14.0	+6.7	+14.0	+5.3	-0.7	+3.3	*0.3896*
Ntcir4	0.2503	+5.5‡	+0.5	0.2613	+10.2‡	+5.0‡	+5.8‡	+4.9‡	+3.9‡	+4.4‡	*0.3494*
Ntcir5	0.3851	+7.4‡	+0.1	0.3949	+10.1‡	+7.3†	-1.2	+2.7	+1.0	+2.5	*0.5261*
Ntcir6	0.3070	+13.9‡	+0.1	0.3142	+16.6‡	+2.5†	4.3†	+2.5†	+1.7	+2.3†	*0.4126*

The results with our variable dependency model (VDM) are shown in Table 5. The results show that the VDM model with fixed weights is slightly better than MRF-SD. This is due to the fact that we added non-adjacent co-occurring characters.

When we vary the weights of the bigram and the pair of co-occurring characters, the result becomes much better. In general, our model outperforms all the baseline methods except in two cases. Many of the improvements are statistically significant. In comparison to B+U, W+U, MRF-SD and VDM with fixed weights, this result shows the benefit of assigning variable importance to pairs of characters. The result clearly validates the general approach we used in our model.

Notice that in the above comparison, we gave considerable advantage to the baseline models, as their parameters are tuned to their best, which is not the case for our model. In order to have an idea of the potential of our model, we also show (in the last column) the effectiveness of our model using the best parameters (best weights for each bigram and pair of characters). We can see that our model with the ideal parameters can largely outperform the existing models. This shows that the assignment of variable weights to individual pairs of characters is indeed an important aspect in Chinese IR, which our model captures. The large difference between the ideal VDM and VDM using parameters set by cross-validation also shows that the parameter learning process can be much improved. This is part of our future work.

4.4 Analysis and Discussion

In this section, we analyze the experimental results in order to understand why our model can outperform the other models. We have observed two categories of cases in which our model can increase the retrieval effectiveness.

1. By setting proper weights to character pairs (high weights to useful pairs, low weights to noisy pairs), our model can benefit from the strengths of unigram model and dependency model, and avoid the disadvantages of them.

- Unigrams (characters) are useful for matching synonyms, near-synonyms or various forms of transliterations due to the characters they share. For example, the two variants of AIDS 爱滋病 and 艾滋病 can be partly matched because they share two characters 滋 (grow, multiply) and 病 (disease). In our experiments, for the query Ch73 in Trec9: "中国的艾滋病" (AIDS in China), the average precision (AP) using words is close to 0 because the documents use a different variant of AIDS - 爱滋病. On the other hand, using unigrams, we obtain an AP of 0.3344. Using VDM, we obtain an AP of 0.4070. In VDM, we observe that except for the bigrams 艾滋 and 滋病, the weights of other bigrams and co-occurring character pairs are close to 0. This means that our model heavily relies on unigrams for this query. However, as some of the bigrams (in particular, the bigram 滋病) have a non-zero weight, they help enhance the connections between these characters. This explains the improved effectiveness of VDM over unigram model.
- On the other hand, characters that are highly ambiguous should be combined and our model can successfully make use of dependencies in these cases. For example, in the query Ch27 of Trec5 "中国 (China) 在 (in) 机器人 (robotics) 方面 (area) 的 (of) 研制 (research)", if we use unigrams, both the terms 中国 (China) and 机器人 (robot) are decomposed into very common characters 中 (China, middle), 国 (country), 机 (machine, engine), 器 (machine, utensil), 人 (human, person). These latter lead to a low effectiveness of 0.1057. When words are used the average precision is increased to 0.4079. Although our VDM model is unable to decide to rely entirely on words in this case, it still assigns a quite strong relative importance to the words, leading to an average precision of 0.3030. The highly ambiguous characters are indeed put into dependencies as follows: 中国 (with a weight of 0.64), 器人 (0.59). These strong weights help solve the ambiguity problem of separate characters.

2. Our model can capture the dependencies between non-adjacent characters.

- For the query 003 of Ntcir4 "胚胎 (embryonic) 干细胞 (stem cells)", we obtain an AP of 0.1891 using unigrams, 0.2174 using MRF-SD, and 0.2410 using WSD, while our VDM model results in an AP of 0.4096. The good performance of VDM is due to the fact that strong dependencies between non-adjacent characters are captured. In this case, we observe strong weights for the co-occurring characters 胎 and 干 (with a weight of 0.22), 胚 and 干 (0.54), 胎 and 胞 (0.27). These pairs do not correspond to legitimate words in this

query, but their combinations tend to enhance the relationship between the words 胚胎 and 干细胞. We can see that co-occurring characters can also successfully capture relationships between different words.

The above analyses show that our model has the potential of taking advantage of both groups of characters and individual characters and determine their importance in the matching process according to how useful they are. This is the very goal of our model.

5 Conclusion

Chinese is a language basically constructed from characters. Even though words can be recognized from sentences, they are not unique and invariable in form. The high flexibility in Chinese to express the meaning using different combinations of characters makes it challenging to match a query against documents using similar but different words. This characteristic of the language is the very reason why a combination of words with characters in Chinese IR has been successful. However, words and characters are not independent. The previous approaches that assume their independence fail to capture the inherent relationships between them. In this paper, we propose a model to take into account the relationships between different types of index. Pairs of characters become dependent to different degrees, and they can be useful for IR to different degrees. In our model, a pair of characters is used in the retrieval model according to its strength and usefulness for IR. The assignment of variable weights to pairs of characters has not been investigated in previous studies. It turns out that this is highly beneficial in our experiments. The model we propose in this paper points to an interesting direction for future research – the integration of dependencies according to their usefulness in IR.

The study reported in this paper has not exploited all the potential of the model. Several aspects could be further improved. For example, we have considered dependencies only between pairs of characters. It would be possible to extend them to more characters. We have used a limited set of features to train the importance of dependencies between characters. This set could be extended in the future.

References

1. Bendersky, M., Metzler, D., Croft, W.B.: Learning Concept Importance Using a Weighted Dependence Model. In: Proc. of the Third International Conference on Web Search and Web Data Mining, pp. 31–40 (2010)
2. Croft, W.B.: Language Models for Information Retrieval. In: Proc. of the 19th International Conference on Data Engineering, pp. 3–7 (2003)
3. Gao, J., Qi, H., Xia, X., Nie, J.-Y.: Linear Discriminant Model for Information Retrieval. In: Proc. of the 28th Annual International ACM SIGIR Conference, pp. 290–297 (2005)
4. Gao, J., Nie, J.-Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: Proc. of the 22nd Annual International ACM SIGIR Conference, pp. 170–177 (1999)
5. Kwok, K.L.: Comparing representations in Chinese information retrieval. In: Proc. of the 20th Annual International ACM SIGIR Conference, pp. 34–41 (1997)

6. Luk, R.W.P., Wong, K.F., Kwok, K.L.: A comparison of Chinese document indexing strategies and retrieval models. ACM Transactions on Asian Language Information Processing 1(3), 225–268 (2002)
7. Metzler, D., Croft, W.B.: Linear feature-based models for information retrieval. Information Retrieval 10(3), 257–274 (2007)
8. Metzler, D., Croft, W.B.: A Markov random field model for term dependencies. In: Proc. of the 28th Annual International ACM SIGIR Conference, pp. 472–479 (2005)
9. Nallapati, R., Allan, J.: Capturing Term Dependencies using a Sentence Tree based Language Model. In: Proc. of the 2002 ACM CIKM, pp. 383–390 (2002)
10. Nallapati, R.: Discriminative Models for Information retrieval. In: Proc. of the 27th Annual International ACM SIGIR Conference, pp. 64–71 (2004)
11. Nie, J.-Y., Gao, J., Zhang, J., Zhou, M.: On the use of words and n-grams for Chinese information retrieval. In: Proc. of the Fifth International Workshop on Information Retrieval with Asian Languages, pp. 141–148 (2000)
12. Nie, J.-Y., Brisebois, M., Ren, X.: On Chinese text retrieval. In: Proc. of the 19th Annual International ACM SIGIR Conference, pp. 225–233 (1996)
13. Shi, L., Nie, J.-Y., Cao, G.: Relating Dependent Indexes using Dempster-Shafer Theory. In: Proc. of the 2008 ACM CIKM, pp. 429–438 (2008)
14. Shi, L., Nie, J.-Y., Bai, J.: Comparing different units for query translation for Chinese cross-language information retrieval. In: Proc. of the 2nd International Conference on Scalable Information Systems, Article, No. 63 (2007)
15. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)