IMPROVE CROSS LANGUAGE INFORMATION RETRIEVAL WITH PSEUDO-RELEVANCE FEEDBACK

Lam Tung Giang[1], Vo Trung Hung[1], Huynh Cong Phap[1]

[1] University of Danang

gianglt@gmail.com, vhung@dut.udn.vn, hcphap@gmail.com

ABSTRACT *- In dictionary-based Cross-language Information Retrieval systems, structured query translation has been shown to be an useful method for improving system performance. In this paper, we examine the effects of using pseudo relevance feedback to refine the structured query in the target language. We propose different methods for term weighting based on word distributions and the mutual information between expanded terms and original query terms. Our experimental results in a dictionary-based Vietnamese-English CLIR system show that while changing query terms weights has effects on improving precision, query expansion improves recall rates. The combination of these two techniques helps to improve system performance up to 12%, in terms of Mean Average Precision.*

Từ khóa *- CLIR, dictionary-based, structured query, Pseudo-relevance feedback, reweight query terms, query expansion.*

I. INTRODUCTION

Cross-Language Information Retrieval (CLIR) has been an important research field with the role to allow users to access documents in languages different from that of query [1][2]. A common approach in CLIR is to translate queries using dictionaries because of the simplicity and the availability of machine readable bilingual dictionaries [3][4].

In dictionary-based CLIR systems, from a given query in the source language, the translated query in the target language is built by selecting the "correct" translations from a list of candidate translations for each term in the initial query [3]. The major problem of this approach is ambiguity: a query term in the source language can have several translation options. There are two mutually exclusive techniques to address this problem. The single selection technique tries to find one best translation for each term. The multiple selection technique, on the other hand, builds a structured query in the target language.

This work is motivated by our previous results in query translation by building a structured query in the target language from a given query [5]. In this paper, we apply the pseudo-relevance feedback technique to improve the structured query translation by learning weights for query terms from top documents returned by the initial retrieval. Our experimental results in a dictionary-based Vietnamese-English CLIR system show that this method helps to improve precision up to 7%. We also examine the effects of query expansion in the target language. Different methods of calculating weights for expanded terms are examined. Our experiments show that query expansion contributes only a minor improvement in precision, however it helps to retrieve more relevant documents. The combination of using query terms re-weighting and query expansion together is shown as a good solution, when it helps to improve system performance up to 12%, in term of Mean Average Precision [6].

The article is structured as follows. Section II presents several works in pseudo-relevance feedback and review our previous work related to structured query translation. In section III, we propose and evaluate effects of query terming re-weighting and query expansion. Section IV presents and analyses the experimental results. Section V presents the conclusions of our study.

II. RELATED WORKS

A. Pseudo relevance feedback

Relevance feedback (RF) was introduced in Rocchio's work [7], where the author introduced a formula for forming a new query vector by maximizing its similarity to relevant documents and minimizing its similarity to non-relevant documents in the collection. Initially, this technique was applied for vector space model and uses user feedbacks on the relevance of documents retrieved from the initial ranking and tries to automatically refine the query.

Since real user feedbacks are hard to obtain, Pseudo-Relevance Feedback (PRF) is used as an alternative solution [8]. PRF assumes the top *n* documents from initial retrieval as being relevant and uses these pseudo-relevant documents to refine the query for the next retrieval. Due to its automatic manner and effective performance, PRF has been widely applied in different IR frameworks like vector space models [8], probabilistic IR [9] and language modeling [10].

Traditional PRF approaches recalculate query term weights based on statistics from retrieved documents and the collection such as term frequency *tf*, document frequency *df*, or term frequency-inverse document frequency *tf-idf*. For example, each term *t* in retrieved documents is assigned an expansion weight $w(t,D_r)$ as the mean of term weights in each retrieved document [9]:

$$w(t, D_r) = \frac{\sum_{d \in D_r} w(t, d)}{R} \quad (1)$$

where R is the number of retrieved documents, $w(t,d)$ is the frequency of a term t in document d. These term weights then are used to define new query by Rocchio's formula:

$$Q_{new} = \alpha \cdot Q + \beta \cdot \sum_{r \in D_r} \frac{r}{R} \quad (2)$$

Here, Q and Q_{new} represent original and new queries. Each document in the set of pseudo-relevant documents D_r is represented as a weighted term vector, annotated by r. R is the number of retrieved documents, α and β are scalar tunable parameters.

More recently, expansion techniques have been introduced within the language modeling framework [10]. In Language Modeling, the probability of a document given a query $P(d|q)$ is estimated using the Bayes' rule:

$$P(d|q) = \frac{P(q|d) \cdot P(d)}{P(q)} \propto P(q|d) \cdot P(d) \quad (3)$$

Usually, $P(d)$ is assumed to be uniform in the collection and query terms are assumed independent with each other. The probability $P(d|q)$ then is equivalent to:

$$P(d|q) \propto P(q|d) = \prod_{t \in q} P\,(t|d) \quad (4)$$

The values $P(t|d)$ are traditionally computed using Jelinek-Mercer smoothing or Dirichlet priors [11]. The basic idea of Relevance-Based (RM) approach is to estimate a better query language model by using information given by the pseudo relevant documents and query terms [12][13]. For a term w and a query q, following formulas (3) and (4), we have:

$$p(w|q) = \sum_{d} p(w|d) \cdot P(d|q) \propto \sum_{d} p(w|d) \cdot P(q|d) = \sum_{d} p(w|d) \prod_{t \in q} P(t|d) \quad (5)$$

To improve the formula (5), different extensions of Relevance-Based (RM) query model try to eliminate the irrelevant terms, which appear frequently in both returned documents and in the whole collection. The following formula interpolates the probability of a term w in the collection C [13][14]:

$$P_\lambda(w|q) = \lambda \cdot P(w|q) + (1-\lambda) \cdot P(w|C) \quad (6)$$

Here, λ is a tunable parameter for controlling irrelevant terms. Top n terms with highest $P_\lambda(w|q)$ are selected to add into the new query.

Other developments for improving query expansion in the literature include approaches using external information, such as Wikipedia [15], Wordnet [16] or query logs of web search[17]; using machine learning to find good feedback documents [18] and good expansion terms [19][20], .

B. Pseudo relevance feedback for CLIR

For cross-lingual information retrieval, PRF can be applied in different retrieval stages of pre-translation, post-translation or the combination of both with the aim of increasing retrieval performance [3][21]. The PRF strategy gives an average improvement across query topics. It works well if there are many relevant documents retrieved in the initial top n, but is less successful when the initial retrieval effectiveness is poor [22].

Hiemstra [23] builds a structured query in the target language by grouping translations for each query term in the source language with the same weight and then use relevance feedback to learn translation probabilities. Daqing and Dan[24] proposes a Translation Enhancement method to expand queries and to enhance query translation by adjusting translation probabilities. Lee and Croft [25] propose a PRF technique for informal text by defining Intra-language and Inter-language PRF models and define a set of features from query analysis to select the appropriate model for each query.

C. Building structured query in the target language

In this part, we review our previous results in query translation by building a structured query in the target language from a given query in the source language [5].

Given a Vietnamese query, we apply a hybrid method combining the use of dictionaries and a POS tagger tool for extracting keywords from a given query in Vietnamese, then propose methods based on Mutual Information to select n best translation candidates for each Vietnamese keyword from the dictionaries (we set n=5) and finally build a structured query in English. In the English structured query, the translation of each query term in the source language is a group of English words, containing the best candidate assigned weight 1 and other candidates assigned weight 0.5. Each group also is assigned a weight depending on the tag assigned to the term in the source language by a tagger tool.

Formally, given a Vietnamese query $q_v = (v_1, \ldots v_n)$, the translated query in English has the following form:

$$q_e = ((e_{1,1}, w_{1,1}) \ OR \ldots OR \ (e_{1,m1}, w_{1,m1}); w_1) \ AND \ldots AND \ ((e_{n,1}, w_{n,1}) \ OR \ldots OR, (e_{n,mn}, w_{n,mn}); w_n) \quad (7)$$

Here $v_1,...,v_n$ are query terms in the given query, values $m1,...,mn$ are numbers of translation candidates of $v_1,...,v_n$. Each pair $(e_{j,k}, w_{j,k})$ contains a translation candidate and its assigned weight for a word v_j. Each value w_i is assigned weight for the group containing translation candidates of the term v_i.

For instance, from the Vietnamese query quản_lý$_{manage}$ quy_trình$_{process}$ sản_xuất$_{production}$, we get the query translation (management^1 OR regulate^0.5 OR control^0.5)^2 (method^1 OR process^0.5 OR instruction^0.5)^4 (production ^1 OR manufacture^0.5 OR fabricate^0.5)^2, which is used in Solr search engine[1].

III. OUR PROPOSED APPROACH

In this article, we examine the effects of using pseudo-relevance feedback for refining the structured query in the target language. Given a query q in the form of formula (7) and the collection D_r of pseudo-relevant documents returned from the initial retrieval, we propose an algorithm for reweighting query terms in the target language and adding new terms to build a new query. The algorithm consists of 5 separated steps:

Step 1: Calculate query terms weights based on term distribution in pseudo-relevant documents.

Step 2: Change the query terms weight, using the new query to retrieve a new list of pseudo-relevant documents.

Step 3: Select top m popular terms contained in new retrieved documents (m=100).

Step 4: Calculate terms weights and select n terms with highest weights (n=5,10,15,20,25).

Step 5: Build the expand query by adding these n terms.

A. Calculate query term weights

Denoting D_r as the set of returned documents in the initial retrieval. The query term weight for a term t contained in query q is calculated by:

$$w(t) = \sum_{d \in D_r} score(d) \frac{count(t,d)}{length(d)} \tag{8}$$

Where $count(t,d)$ is the number of times a term t appears in a document d contained in D_r, $length(d)$ is the length of the document d, $score(d)$ is the score assigned to the document d by the search engine.

The term weights calculated by formula (8) are used to reformulate a new query. At this step, the new query has the following form:

$$q' = \left(e_{1,1}w_{1,1}\ OR \dots OR\ e_{1,n1}w_{1,n1}\right) AND \dots \left(e_{n,1}w_{n,1}\ OR \dots OR\ e_{n,nm}w_{1,nm}\right) \tag{9}$$

Here $e_{i,j}$ and $w_{i,j}$ are similar with those in formula (7). Please note that the weights assigned for translation groups in formula (7) are removed in the new query q'.

B. Select top popupar terms

With the set D_r of returned documents in the second retrieval using query q', all documents d_i in D_r are vectorized. In the result, a dictionary $Dict = \{t_1,...t_{|D|}\}$ is created, containing all terms belonging documents in D_r. Each document is represented as a vector $d_i = \{w_{i,1}, ..., w_{i,|D|}\}$, where $w_{i,j}$ is the *tf-idf* weight of the term t_i of the document d_i in the set D_r. For each term not contained in the query q, the term weight is calculated by the next formula:

$$w(\mathrm{t_j}) = \frac{\lambda}{|D_r|} \sum_{d_i \in D_r} w_{i,j} \tag{10}$$

with λ is a turnable parameter (we set λ=1 here). The formula (10) is used to select top m popular terms (m=100 in our experiments) in the retrieved documents. The expanded terms will be chosen among these terms.

C. Calculate new term weights

This part introduces 4 ways for calculating expanded term weights. The first way calculating term weights simply uses the formula (10), denoted as FW1. The n terms with highest weights then are used to add into the new query.

The second way combines the local *tf-idf* weight and the global *idf* weight of terms. For each term t_j, term weight is calculated by the formula FW2:

$$w(\mathrm{t_j}) = \frac{\lambda}{|D_r|} \sum_{d_i \in D_r} w_{i,j} \cdot \log\left(\frac{N+1}{N_{t_i}+1}\right) \tag{11}$$

Here N is the total number of documents in the collection. N_t is the number of document containing the term t, λ is a tunable parameter.

[1] http://lucene.apache.org/solr/

Based on the assumption that terms closer to the query terms are more likely to be relevant to the query topic, we propose the third way of calculating term weights uses mutual information of expanded terms and the initial query terms. First, we learn a "local" co-occurrence model of term pairs from the pseudo-relevant documents. For each term t_j, and a query term q_k, we denote $mi(t_j,q_k)$ as the number of times two these terms are in a distance of 3 in retrieved documents. The term weight then is calculated by formula FW3 as follows:

$$w(\mathrm{t_j}) = \lambda \cdot \sum_{q_k \in q} mi(t_j, q_k) \tag{12}$$

Another way of using Mutual Information is building a "global" co-occurrence model of term pairs in the whole collection, then the term weight is calculated by formula FW4:

$$w(\mathrm{t_j}) = \lambda \cdot \sum_{q_k \in q} mi(t_j, q_k) \cdot \log(\frac{N+1}{N_{q_k}+1}) \tag{13}$$

where N is the total number of documents in the collection. N_t is the number of document containing the term t.

By adding top n terms with highest term weights, the final query has the following form:

$$\begin{aligned} q_{final} &= q^{'} AND\ (expanded\ terms) \\ &= (e_{1,1}w_{1,1}\ OR \ldots OR\ e_{1,n1}w_{1,n1})\ AND\ \ldots (e_{n,1}w_{n,1}\ OR \ldots OR\ e_{n,nm}w_{1,nm})\ AND\ t_1w_1 \ldots t_nw_n \end{aligned} \tag{14}$$

Here $e_{i,j}$ and $w_{i,j}$ are similar with those in formula (7), t_i is an expanded term and w_i is the weight assigned for t_i.

IV. EXPERIMENTAL RESULTS

A. Test configuration

To evaluate presented methods, we conduct the following experiment: first, we collect 24000 English documents from Web and build an English monolingual IR system on top of the open source search tool Solr. We use 50 Vietnamese queries with an average length of 8,73 words for our experiment. At first, we apply query translation methods in [5] to build structured queries in English. After that, we follow the algorithm presented in section 3 to change query term weights. With top m 100 popular terms (m=100) from 50 retrieved documents, we use formulas FW1, FW2, FW3, FW4 defined in section 3 to calculate term weights. Top n terms (n=5, 10, 15, 20 or 25) with highest weights are selected for query expansion.

B. Results

With each formula, we examine the system performance with different values of λ: 0.0001, 0.0002, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2 and 0.5. It turns out that the tunable parameter λ is an important factor for selecting expanded terms. We conduct experiments with different values and select the final values for λ as 0.1,0.01, 0.001 and 0.001 when applying formulas FW1, FW2, FW3 and FW4 respectively.

The next two tables show the MAP scores and the number of relevant documents being retrieved for different test configurations. In each table, the first row (used as the *baseline*) and the second row (marked as the *CW* configuration) show performance when using original translated query and after changing query term weights. The next 4 rows show performance when we expand query with 5, 10, 15, 20 or 25 top terms by applying different term weighting formulas FW1, FW2, FW3, FW4.

Table 1. MAP Scores

		n=5	n=10	n=15	n=20	n=25
Baseline	0.380					
CW	0.407					
FW1		0.416	0.421	0.417	0.416	0.410
FW2		0.416	**0.425**	0.418	0.415	0.411
FW3		0.414	0.411	0.413	0.411	0.412
FW4		0.404	0.400	0.388	0.386	0.367

By changing query term weights using formula (8), the MAP score is improved 7%. With n=10, the query expansion method using the formula FW2 gives the best MAP score of 0.425, which is 112% of the baseline and 104% of CW configuration. The formula FW1 also gives a high MAP score of 0.421, which is 111% of the baseline and 103% of the CW configuration. It can be seen that query term reweighting is the main factor to contribute to the system precision.

The results in the table 2 show that the number of retrieved relevant documents is declined in the *CW* configuration. However, this number is increased when we apply query expansion. The best result of 5179 retrieved documents from total 6109 relevant documents is reached with n=15 and the formula FW4 based on global mutual information is used.

Table 2: Number of retrieved relevant documents

		n=5	n=10	n=15	n=20	n=25
Baseline	4999					
CW	4961					
FW1		5044	5047	5075	5075	5071
FW2		5010	5067	5061	5082	5099
FW3		5081	5075	5095	5070	5072
FW4		5019	5004	5179	5098	5127

The figure 1 presents the interpolated 11-point average precision for 4 configurations: *baseline*, *CW*, FW1 and FW2 with *n*=10. It shows a clear advantages of the combination of proposed algorithms with term ranking formulas FW1 and FW2 over the baseline.

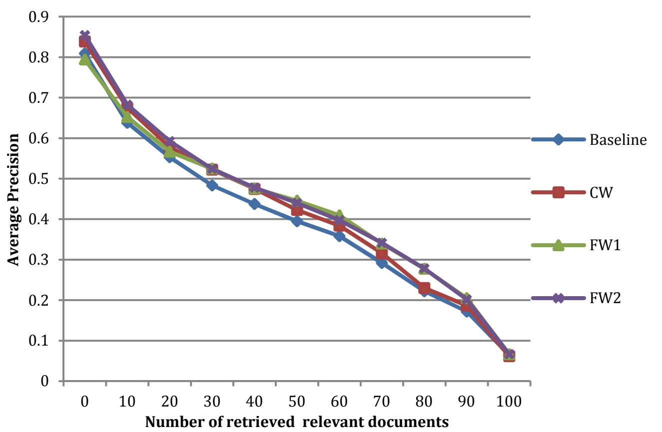

Figure 1. Interpolated 11-point average precision (n=10)

V. CONCLUSIONS

In this article, we propose an algorithm to refine the query translation in a dictionary-based CLIR system based on the use of pseudo-relevance feedback. We separate the two steps of query term re-weighting and query expansion to examine the effects of each step. We propose different variants for terms weighting using returned documents from the initial retrieval, including using local *tf-idf* term weight, combining local *tf-idf* term weight and global *idf* weight, and using mutual information of terms and initial query terms. These weighting schemes are used for re-weighting query terms and for the query expansion. Our experimental results show that the combination of the proposed algorithm and weighting functions FW1 and FW2 helps to improve the system precision and recall rates.

VI. REFERENCES

[1] Gerard Salton, Automatic processing of foreign language documents, *Journal of the American Society for Information Science*, vol. 21, no. 3, pp. 187–194, 1970.

[2] Jian-Yun Nie, *Cross-Language Information Retrieval*. Morgan & Claypool Publishers, 2010.

[3] Dong Zhou, Mark Truran, Tim Brailsford, Vincent Wade, and Helen Ashman, Translation techniques in cross-language information retrieval, *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–44, 2012.

[4] Ari Pirkola, Turid Hedlund, Heikki Keskustalo, and Kalervo Järvelin, Dictionary-Based Cross-Language Information Retrieval: Problems, Methods, and Research Findings, *Information Retrieval*, vol. 4, no. 3, pp. 209–230, 2001.

[5] Lam Tung Giang, Vo Trung Hung, and Huynh Cong Phap, Building Structured Query in Target Language for Vietnamese – English Cross Language Information Retrieval Systems, *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 04, pp. 146–151, 2015.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[7] J. J. Rocchio, Relevance Feedback in Information Retrieval, *SMART Retrieval System Experimens in Automatic Document Processing*, 1971.

[8] Gerard Salton and Chris Buckley, Improving retrieval performance by relevance feedback, *Journal of the American Society for Information Science*, vol. 41, no. 4, pp. 288–297, 1990.

[9] Zheng Ye, Ben He, Xiangji Huang, and Hongfei Lin, Revisiting Rocchio's relevance feedback algorithm for probabilistic models, *Lecture Notes in Computer Science*, vol. 6458 LNCS, pp. 151–161, 2010.

[10] John Lafferty and Chengxiang Zhai, Document language models, query models, and risk minimization for information retrieval, *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 01*, vol. 01, pp. 111–119, 2001.

[11] Chengxiang Zhai and John Lafferty, A study of smoothing methods for language models applied to Ad Hoc information retrieval, *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, 2001, pp. 334–342.

[12] Victor Lavrenko and W. Bruce Croft, Relevance-Based Language Models, *Sigir'01*, p. 8, 2001.

[13] ChengXiang Zhai and John D. Lafferty, Model-based Feedback In The Language Modeling Approach To Information Retrieval, *Cikm*, pp. 403–410, 2001.

[14] Stéphane Clinchant and Eric Gaussier, A Theoretical Analysis of Pseudo-Relevance Feedback Models, *Proceedings of the 2013 Conference on the Theory of Information Retrieval - ICTIR '13*, pp. 6–13, 2013.

[15] Yang Xu, Gareth J. F. Jones, and Bin Wang, Query Dependent Pseudo-Relevance Feedback based on Wikipedia, *Sigir*, 2009.

[16] M. Voorhees and Road Nj, Expansion using lexical-semantic relations, *SIGIR'94 Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 61–69, 1994.

[17] Ben He and Iadh Ounis, Combining fields for query expansion and adaptive query expansion, *Information Processing and Management*, vol. 43, no. 5, pp. 1294–1307, 2007.

[18] B. He and I. Ounis, Finding good feedback documents, pp. 2–5, 2009.

[19] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson, Selecting good expansion terms for pseudo-relevance feedback, *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, p. 243, 2008.

[20] Saúl Vargas, Rlt Santos, Craig Macdonald, and I. Ounis, Selecting Effective Expansion Terms for Diversity, *OAIR 2013 10th International Conference in the RIAO series*, 2013.

[21] Kazuaki Kishida, Technical issues of cross-language information retrieval: a review, *Information Processing & Management*, vol. 41, no. 3. pp. 433–455, 2005.

[22] M. Sanderson and P. Clough, Measuring pseudo relevance feedback & CLIR, pp. 484–485, 2004.

[23] Djoerd Hiemstra, Wessel Kraaij, Ren´ee Pohlmann, and Thijs Westerveld, Translation Resources , Merging Strategies , and Relevance Feedback for Cross-Language, *CrossLanguage Information Retrieval and Evaluation Workshop of CrossLanguage Evaluation Forum CLEF 2000*, pp. 102–115, 2000.

[24] Daqing He and Dan Wu, Translation enhancement: a new relevance feedback method for cross-language information retrieval, *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 729–738, 2008.

[25] Chia-Jung Lee and W. Bruc. Croft, Cross-Language Pseudo-Relevance Feedback Techniques for Informal Text, *Advances in Information Retrieval, SE - 22, 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, 2014, vol. 8416, pp. 260–272.

NÂNG CAO HIỆU QUẢ TRUY VẤN XUYÊN NGỮ BẰNG PHẢN HỒI GIẢ

Lâm Tùng Giang, Võ Trung Hùng, Huỳnh Công Pháp

TÓM TẮT - *Trong các hệ thống truy vấn xuyên ngữ dựa trên kỹ thuật dịch sử dụng từ điển, tạo lập câu truy vấn có cấu trúc được đánh giá là một phương pháp hữu hiệu nhằm nâng cao hiệu quả hệ thống. Trong bài báo này, chúng tôi kiểm tra ảnh hưởng của việc sử dụng phản hồi giả để điều chỉnh câu truy vấn trong ngôn ngữ đích. Chúng tôi đề xuất các phương pháp tính trọng số từ khóa dựa trên phân bố từ và thông tin liên quan giữa các từ khóa mở rộng và từ khóa của câu truy vấn gốc. Kết quả thử nghiệm của chúng tôi với một hệ thống truy vấn xuyên ngữ Việt-Anh cho thấy trong khi việc thay đổi trọng số từ khóa trong câu truy vấn có tác dụng nâng cao độ chính xác, việc mở rộng câu truy vấn giúp tăng độ bao phủ. Việc kết hợp hai kỹ thuật này giúp tăng điểm số MAP tới 12%.*