Building Structured Query in Target Language for Vietnamese – English Cross Language Information Retrieval Systems

Lam Tung Giang
Danang People Committee
Vietnam

Vo Trung Hung
University of Danang
Vietnam

Huynh Cong Phap
University of Danang
Vietnam

***Abstract*—Query translation is the most important component in Cross Language Information Retrieval systems using dictionary-based approach. In this paper, we present a method to build structured query in target language from a given query in source language. The method is based on constructing bi-lingual dictionaries, keyword extraction from source query, getting translation candidates for each keyword using mutual information and finally building structured query in target language. By combining several translations for each query term in target language, we overcome shortcomings of selecting only one translation and therefore improve system performance. Results for Vietnamese-English cross-lingual retrieval show improvements of building structured query over other methods using dictionary to produce query translation in target language by combining single selection translation for each keyword.**

Keywords — CLIR; dictionary-based; Vietnamese; keyword extraction; structured query; disambiguation; mutual information

I. INTRODUCTION

With the growing amount of web documents available in different languages, Cross-language Information Retrieval (CLIR), a subfield of Information Retrieval, becomes increasingly important with the role to allow users overcome language barrier to access documents in languages different from that of query [1]. The common approach in CLIR is to translate queries from source language into target language, and then search documents in this target language. The query translation can be executed by different methods: using dictionary, using parallel corpora, machine translation, or ontology-based. In particular, dictionary-based translation is widely used method because of its simplicity and the availability of machine readable bilingual dictionaries [2].

In this article, we review the process of creating the query in target language from a given query in source language, analyse problems in each step and suggest improvements. First, we construct two dictionaries: *wiki dictionary* is built by extracting data from from Wikipedia intert-language links, *normal dictionary* is built by reorganizing the data of Free Vietnamese Dictionary Project[1]. Next, we propose an algorithm for extracting keywords which are needed to translate from a given query. Our algorithm includes an error-correction step, which allows to choose correct keywords from the query text in case extracted keywords overlapping. In the third step, we apply a method using Mutual Information score to find best translations for each extracted keywords. Finally, a structured query in the target language is built by combining translations.

The article is structured as follows. In section 2, we review the process of creating the query in target language. Section 3 describes our proposed methods and the implementation. Section 4 presents and analyses experimental results and section 5 concludes our study.

II. RELATED WORKS

A. *Translation methods used in CLIR*

The most natural approach to CLIR is to replace each query term with most appropriate translations extracted automatically from bilingual dictionaries, which have become increasingly available. The three main problems in this dictionary-based approach include query segmentation, dictionary coverage and lexical ambiguity [3][4]. These problems cause low performance of this approach. Cross-language effectiveness using dictionary-based approach is less than 60% in comparison with mono-lingual retrieval (in terms of average precision).

Another popular query translation method applied in CLIR recently is using machine translation systems (MT systems) to translate queries into target language. In practice, Google Translate[2] is widely used in CLIR systems and provides high-quality translations. In CLEF 2009 campaign, this tool overcame other approaches and achieved the equivalent of 90%-99% of the monolingual baseline on English, French and German collections [5]. The down side of this method is it produces only one single output and therefore misses other translation options. Using commercial MT as a black box, developers are also depended on the tool and can not improve translation quality [2].

The third translation method is based on parallel corpora. Parallel corpora are collections where texts in one language are aligned with their translations in another language. In general, these corpora are built by mining parallel text from the Web and then are used for building statistical models, containing translation probabilities of words in target language being the translation of words in source language. The key disadvantage of this method is the difficulty in obtaining suitable document collections. One possible solution for this involves the use of comparable corpora, which are texts that are not translations, but share similar topics.

The fourth method is based on ontologies, which are semantic networks consisting of multiple-level concepts and semantic relations between them. Documents and queries are then annotated by mapping inside terms to concepts in an ontology and then conceptual distances between nodes are used to measure similarities between queries and documents. This method exploits multilingual ontologies to bridge the gap between surface linguistic form and meaning. However, it is hard to build semantic networks. A small semantic network also can cause a low performance.

B. *Translation process analysis*

Typically, there are three main steps in dictionary-based approach. The first step involves extracting keywords from the given query. These keywords will be looked up in a dictionary in the second step for choosing appropriate translations (each word in query can have some possible translations). Finally, a new query in the target language is created from these translations. These three steps are matched with the three steps in the conceptual model of CLIR described in [2]: pre-translation, translation, post-translation steps.

The first step is responsible for analyzing the given query in source language to identify elements for translation and can consist of several tasks: tokenization, stop-word removal, stemming and keyword expansion. Different techniques are applied for query tokenization. For English, French, it is natural to separate words by white spaces. Certain languages (e.g., German, Dutch, Russian) are rich in compound words and thus the query is needed to de-compound. Other problems arise with query segmentation for Asian languages such as Vietnamese and Chinese due too the lack of word boundary. Most of Vietnamese words have multiple syllables [6] and need to be extracted correctly from the query to get right translations. There are several word segmentation methods for Vietnamese based on algorithms such as maximal matching, longest matching, dictionary-based, transition graph, tagger-based or using MI-score [7][8][9]. It is empirically shown that heuristic methods outperform other methods using only single algorithm. Despite reported high scores of these algorithms, they are not without errors. In particular, most of published algorithms are limited in produce single words, for example *tàu (train or ship)*, or *sân bay (airport)* and can not produce complex words like tàu *sân bay (aircraft carrier)* and thus lead to imprecise translation.

Expansion occurs when additional terms are added to the query to improve its quality, expressiveness. This can be done by adding synonyms from a dictionary. Another technique is known as pseudo-relevance feedback (PRF) [10] by searching in source language and extracting high weighted terms from the top *n* documents returned by this search and adding to the query.

In the second step, each keyword extracted in step 1 is looked up to find its translations in the dictionary. For Vietnamese, the main obstacle is the dictionary coverage. The original VDict open source dictionary is limited with 23,000 entries in Vietnamese-English version and about 100,000 entries in English-Vietnamese version.

For the disambiguation, various approaches have been proposed, such as using the first term listed in the dictionary, using relevance feedback, using a parallel or a comparable corpus. Recently, mutual information (MI), which is calculated from co-occurrence frequency of terms in a monolingual corpus become widely used to select correct combinations of translations [11] [12][13]. The hypothesis grounding the use of term co-occurrence data in this context states that the correct translations of query terms will tend to co-occur more than incorrect translations.

There are two ways in step 3 to build a query in the target language. The first one joins best translations for each term in source query to build a single sentence. Another way is select best translations for each term and and build a complex, structured query with the syntax of the monolingual search engine being used. With this approach, term weights, calculated in disambiguation process also can be added.

In our experiment, we follow the second way to examine the performance of using structured query over single sentence produced by other methods.

III. Building structured query

In this part, we describe our approach to build a structured query in English from a given Vietnamese query. At the keyword extraction step, we apply Vietnamese tagger tool vnTagger to identify keywords. Besides, we also extract Vietnamese words contained in the query having at least one translation in the dictionaries. The two sets are merged to get a list of keywords. For each keyword, we follow the statistical approach based on Mutual Information and propose two methods to find best translations. Finally, a structured query in English is built by joining 3 best translations of each Vietnamese extracted keyword and adding weight for each keyword depending its type in source language. For example, from the Vietnamese query *quản lý quy trình sản xuất*, we produce the English query *(management OR regulate OR control)^2 (method OR process OR instruction)^4 (production OR manufacture OR fabricate)^2*. Here each noun is assigned weight 4 and each verb is assigned weight 2.

A. *Dictionaries*

The dictionary size plays crucial role for the two steps keyword extraction and translation. In this article, we download the new data of Vietnamese-English and English-Vietnamese dictionaries of Free Vietnamese Dictionary Project. Each item in dictionaries is analyzed to extract translation pairs consisting of one Vietnamese word and one of its possible translations in English. At the result, we get a new version of Vietnamese English dictionary with more than 600,000 entries. In addition, we build the *wiki dictionary* from Vietnamese-English word pairs in Wikipedia language links database ^{3}. The special feature of the *wiki dictionary* is each word has only one translation. This dictionary is very helpful for translation of Named entities, which are not available in normal dictionaries.

B. Monolingual search

In our experiment, we use the open source search tool *Solr* version 4.3.10[4] to build monolingual search systems for Vietnamese and English. Our crawler uses article titles in online newspapers as queries and send them to Google search engine to get address and then download web pages in search result lists. For the Vietnamese search system, we add 200,000 Vietnamese documents downloaded from website *http://vietnamplus.vn*. In the result, we have built two document collections for search engines: more than 200,000 Vietnamese articles and about 12,000 English articles for our experiment.

C. Extracting keywords and possible translations

Our first step to extract keywords from a given Vietnamese query consists of 5 mini steps, described in the following algorithm:

```
Algorithm Extract keywords from a given query
Input: a Vietnamese query
Output: a list of item, each contains a Vietnamese word and its  translation candidates
Begin
Step 1:
Process POS tagging for the query, extract words in the query with their tags
list_keywords = empty
for each word:
  look up in the normal dictionary
    if found
      extract all English translations
    else:
      if tag = 'Np':
        remove Vietnamese accents
        take the word as its translation
    add item (word,translations) to list_keyword
extract from the normal dictionary all words that are contained in the query and their possible translations
for each word in this list:
  add item (word,translations) to list_keyword if the word not in  list_keyword
Extract from the wiki-dictionary all words that are contained in the query and their possible translations.
For each word in the wiki list:
  if word exist in the list_keyword:
    replace the item of existing word by the item associated with the word in wiki list
Step 2:
for each item in list_keyword:
  if the Vietnamese word in POS tag list:
    assign tag to the item
  else
    create the list of tagged word contained in the pair's word
    assign the tag with highest level to the  item
Step 3
Remove items that have Vietnamese word contained in the word of another item in  list_keyword
Step 4
Send the query to Vietnamese IR system to get a sample text
For each Vietnamese word in items:
  calculated the word weight calculated by formula (1) below
Compare items with overlapped words, remove the one having lower weights to get the list of good items
Step 5
Create a fuzzy text by remove all word of list_keyword's items from the query text
Check the list of items of tagged words, add items with word that is contained in the fuzzy text to the list of good items
return the list of good items
End
```

At first, we use the tool *vnTagger* for tagging the Vietnamese query with the result is a list of Vietnamese words and their part-of-speeches. Each them is looked up in dictionaries to get possible translations. For words tagged as *Np* (noun phrase), if there is no translation, we consider it as a foreign word, a proper name, a technical term, or an acronym. For these words, we remove Vietnamese accents and treat the words as their translations.

Besides, we extract all items of Vietnamese words and their English translation from the Vietnamese-English dictionary and the wiki-dictionary, in which the Vietnamese word is fully contained in the query. In the result, we get a list of items, each consists of a Vietnamese word and its possible translations.

Detecting word tag plays an important role in query analysis. Noun phrases (*Np* tag) are normally location names, organization names, person names and if a noun phrase appears in the query, it should be assigned a high weight. We also give higher importance level for nouns than for verbs, adjectives and other types. These importance level values will be used to build the structured query.

In the third step, we want to remove *bad* keywords, which can poorly affect the system performance. First, we remove words tagged as preposition, numeral, conjunction, determiner. For remaining keywords, we remove those, which are fully contained in another keywords in the dictionary list. Our argument is if one compound word exists in the bilingual dictionary and contains other words (called internal words), its translation tends to be better than the combinations of translations for internal words.

At this step, there still can be overlapped words, and we execute the correction in step 4: the Vietnamese query is sent to the search system. With the set of *n* top returned documents (*n*=10 in our experiment), we join all text into a string *bigtext* and calculate keywords weights by the formula:

$$weight(w) = nq(w) * \log\left(\frac{1+nc(w)}{nc(w)}\right) + \log(1+n \qquad (1)$$

where *nq(w)* is number of times the word *w* occurs in *bigtext*, and *nc(w)* is number of documents in collection containing word *w*. All keywords are sorted by their weights and between overlapped words, we remove those with lower weights.

Next, we create a fuzzy text by replacing all extracted keywords in the query by empty spaces and review the list of tagged words by *vnTagger*. If a word is contained in this fuzzy text, it is added to the list of *good* keywords. Finally, we get the list of *good* keywords and their translations.

D. Best translations selection

In our research, we follow the approach using Mutual Information score to find best translations for each extracted keywords. First, we define 2 formulas to calculate Mutual Information score. Next, we propose 2 methods for finding best translations.

1) Calculating Mutual Information score

Here we apply two formulas to calculate the mutual information (*MI* value) of two words. For the first formula, we train a monolingual text corpus to build a word co-occurrence model for calculating words similarities. For two words *x* and *y*, the value *MI(x,y)* is calculated as follow:

$$MI_{cooc}(x,y) = \log\left(\frac{p(xy)}{p(x)p(y)}\right) - C \qquad (2)$$

with *p(x)* and *p(y)* are frequencies of words *x* and *y* in the text corpus, and *p(x,y)* is the number of times two words co-exists in a same sentence. Value of *C* is based on the text corpus size. In our experiment, we define C = $\log_2(12000)$.

The second formula is based on the monolingual English IR system. For two words *x* and *y*, the value *MI(x,y)* is calculated as follow: we send strings *x*, *y* and *"x AND y"* as queries to the English IR system and get *n(x)*, *n(y)*, *n(x,y)* as numbers of documents containing string *x*, *y* and *x AND y*. If *n(x)* or *n(y)* equals 0, the return value is 0, if not, we use the following formula:

$$MI_{ir}(x,y) = \frac{n(x,y)}{n(x)n(y)} \qquad (3)$$

The advantage of this formula is the calculation is executed directly on the document collection being searched and can help to eliminate word pairs not existing in the document collection. However it requires heavy calculation.

2) Selecting best translations by coherence score

The first algorithm of selecting best translations for each Vietnamese keyword is similar to [11]. From the result of the previous step, each Vietnamese query q_v is represented as a set $((w_1,L_1),(w_2,L_2), \ldots((w_n,L_n))$, in which each w_i is a Vietnamese word and L_i is the list containing translation candidates of w_i.

For a word *e* and a list of words $L = (t_1,t_2,..,t_n)$,we define:

$$MI(e,L) = \underset{t \in L}{sum}\, MI(e,t) \qquad (4)$$

For each Vietnamese keyword w_i, we get best translations by sorting L_i by a *cohesion score* for each word *e* in L_i, which is calculated by the following formulas:

$$cohesion\ (e) = \sum_{j \neq i} MI\ (e, L_j) \qquad (5)$$

Finally, the three English words with highest *cohesion scores* are selected as possible translations of keyword w_i for building English query in the next step.

3) Extracting best translations sequentially

The idea of the second algorithm is based on constructing a set of columns, each contains a Vietnamese word and its translation, created in the step extraction keywords, then extracting translations sequentially. We first select best translations from pairs of adjacent columns, then look at columns standing immediately before and after the set of selected columns to find the best translation by *cohesion score* in the following formula (6) below. The process continues until we examine all columns. Finally, we get the list of best translations for Vietnamese words. This algorithm is similar to [13], but here we look at only pairs of adjacent words to limit side effects of choosing words appearing far to each other in the query string.

The algorithm is described in details as follow:

```
Algorithm SQ: Extracting best translations sequentially
Input: a list of items, each contains a Vietnamese word and its translation candidates
Output: a list of items, each contains a Vietnamese word and its best translations
Begin
Step 1:
create set AllColumns from index of all input items
create a list of pairs from adjacent input items
  for each pair $(w_i,w_{i+1})$
    for each translation $t_i^k$ of $w_i$ and each translation $t_{i+1}^l$ of $w_{i+1}$:
      calculate value mi( $t_i^k$ , $t_{i+1}^l$ )
  choose the pair ( $t_i^{best}$ , $t_{i+1}^{best}$ ) with highest MI value as translations of Vietnamese word $w_i$ and $w_{i+1}$
TranslationSet = [($w_i$, $t_i^{best}$ ), ($w_{i+1}$, $t_{i+1}^{best}$ )]
Create the set GoodColumns containing value i and i+1
Step 2:
while GoodColumns <> AllColumns
```

```
        $cohesion\left(t_i^k\right) = \sum_{c \in GoodColumns} MI\left(t_i^k, t_c^{best}\right)$        (6)

        choose $t_i^{best}$ with highest cohesion score as the translation of Vietnamese word
        add $(w_i, t_i^{best})$ to TranslationSet
        add value i to the set GoodColumns
    return TranslationSet
    End
```

E. Building structured query

There are several ways to build a structured query to process the search in English. The first way is simply use the one best translation for each Vietnamese word. In the second way, we join possible translations by operator *OR* to create a group. For each group, keyword tag assigned by *vnTagger* in source language is verified and a weight is assigned to group (8 for noun phrase (tag *Np*), 4 for normal noun (tag *N*), 2 for verb (tag *V*) and 1 for other tags in our experiment). The final query is created by joining these groups by operator *AND.*

IV. EXPERIMENTAL RESULTS

A. Test configuration

To measure the effectiveness of the proposed methods, we conduct the following experiment: With the Vietnamese and English monolingual IR systems built as described, we create 25 Vietnamese queries with average length at 9.52 words. We test and compare following configurations:

top_one_ch: use formula MI_{ir} and the *cohesion score* as formula (5), select one best translation for each Vietnamese word to build the query.

top_three_ch: use formula MI_{cooc} and the *cohesion score* as formula (5), build structured query by joining 3 best translations of each word.

top_one_sq: use formula MI_{cooc}, select one best translation by the algorithm *SQ*.

top_three_sq: use formula MI_{cooc}, select the best translations by the algorithm *SQ*, then add more 2 candidates, which are most similar to the best translation (also measured by *MI score*).

top_three_all: follow the same algorithm as *top_three_sq*, but use a special MI formula:

$$MI(x,y) = \alpha MI_{cooc}(x,y) + (1-\alpha) MI_{ir}(x,y)$$

Here we choose $\alpha = 0.02$

google: using Google Translate tool for query translation.

base_line: manually translate the Vietnamese query.

For the evaluation of proposed methods, we use the standard retrieval performance evaluation measures introduced in [14] . For each query, the precision at position *k*, value *P(k)*, is defined as the fraction from top *k* documents returned from IR system that are relevant. The average precision is calculated by the following formula:

$$AP = \frac{\sum_{k=1}^{n} P(k) * rel(k)}{N} \qquad (7)$$

where *n* is the number of retrieved documents, *N* is the number of relevant documents, *rel(k)* is an indicator function equaling 1 if the item at rank *k* is a relevant document, zero otherwise. Finally, Mean Average Precision (MAP) for a set of queries is the mean of the average precision scores for each query.

$$MAP = \frac{\sum_{1=1}^{Q} AP(q)}{Q} \qquad (8)$$

where *Q* is number of queries, *AP(q)* is the average precision of query *q* calculated by the formula (6).

B. Test results

The next table describes our test results. For each method, we show average values of *P(k)* (with k = 1, 5, 10) and the *MAP* value. In the last column (column *Perf.*), we compare MAP values of proposed algorithms with the baseline method using manual translation.

TABLE I. TEST RESULT

	Configuration	P1	P5	P10	MAP	Perf.
1	top_one_ch	0.64	0.48	0.444	0.275	71.24%
2	top_one_sq	0.52	0.472	0.46	0.291	75.39%
3	top_three_ch	0.68	0.528	0.524	0.316	81.87%
4	top_three_sq	0.64	0.552	0.532	0.323	84.55%
5	top_three_all	0.76	0.576	0.54	0.364	94.30%
6	Google	0.64	0.568	0.536	0.349	90.41%
7	Baseline	0.76	0.648	0.696	0.386	100%

C. Result Analysis

All of propose configurations have performance higher 70% in comparison with the baseline using manual translation. Structured queries created by *top_three_all, top_three_ch* and *top_three_sq* give much better results in comparison with the two configurations *top_one_ch* and *top_one_sq*, which select only one best translation for each Vietnamese word.

In some queries, our translations do not match expressions in documents. For instance, from "*kiểm soát hoạt động trên biển*" we get "*(supervise OR to oversee OR monitor)^2 (action OR activity OR operation)^4 afloat*", which is a good structured query. However we get a low score for this query because the expression *afloat* translated from *"trên biển"* by the dictionary is not common used in documents. At the same time, other words are too common for finding relevant documents. The query should be translated as "*monitor activities on sea*". We think we can overcome this problem in future with a deeper grammar structure analysis on noun phrases in Vietnamese queries.

The configuration *top_three_all* gives highest MAP score at *0.364*, which is *94.30%* in comparison with manual translation. This score is better than score of *0.349* given by Google Translate tool. Regarding configuration *top_three_sq*, the deep analysis shows this configuration produces better results in 9 queries in the comparison with Google Translate and in 7 queries in the comparison with manual translation, when documents contain various word expressions for same contents. We hope that we can use our methods and existing methods as complements to improve Vietnamese-English CLIR systems.

V. Conclusion

In the article, we propose an approach for building a structured English query from a given Vietnamese query. Our method of keyword extracting works effectively with tested queries. The best tested configuration of building structured query based on using multiple translations and keyword weighting reaches 94.30% performance of the search using manual translation.

In near future, we plan to extend our research on deep grammar analysis in source language and using relevance feedback to reweight query terms in target language for a better performance.

References

[1] Douglas W. Oard and Bonnie J. Dorr, A Survey of Multilingual Text Retrieval, *Electr. Eng.*, pp. 1–31, 1996.

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

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

[4] Jianfeng Gao, Jian-Yun Nie, Endong Xun, Jian Zhang, Ming Zhou, and Changning Huang, Improving query translation for cross-language information retrieval using statistical models, *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 94–104.

[5] Nicola Ferro and Carol Peters, CLEF 2009 Ad Hoc Track Overview : TEL & Persian Tasks, *Proceedings of the 10th Cross-language Evaluation Forum Conference on Multilingual Information Access Evaluation: Text Retrieval Experiments (CLEF'09)*, 2009, pp. 13–35.

[6] Nguyen Han Doan, Vietnamese-English Cross-language information retrieval (CLIR) using bilingual dictionary, *International Workshop on Advanced Computing and Applications Ho Chi Minh City*, 2007.

[7] Oanh Thi Tran, Cuong Anh Le, and Thuy Quang Ha, Improving Vietnamese Word Segmentation and POS Tagging using MEM with Various Kinds of Resources, *J. Nat. Lang. Process.*, vol. 17, no. 3, pp. 41–60, 2010.

[8] Nguyen Thi Uyen and Tran Xuan Sang, Dynamic Programming Method Applied in Vietnamese Word Segmentation Based on Mutual Information among Syllables, 2014, vol. 3, no. 9, pp. 24–27.

[9] Dinh Quang Thang, Le Hong Phuong, Nguyen Thi Minh Huyen, Nguyen Cam Tu, Mathias Rossignol, and Vu Xuan Luong, Word segmentation of Vietnamese texts : a comparison of approaches, *6th international conference on Language Resources and Evaluation - LREC*, 2008, pp. 1933–1936.

[10] Gerard Salton and Chris Buckley, Improving retrieval performance by relevance feedback, *J. Am. Soc. Inf. Sci.*, vol. 41, no. 4, pp. 288–297, 1990.

[11] Jianfeng Gao, Jian-Yun Nie, Endong Xun, Jian Zhang, Ming Zhou, and Changning Huang, Improving query translation for cross-language information retrieval using statistical models, *Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. - SIGIR '01*, pp. 96–104, 2001.

[12] Mirna Adriani, Using Statistical Term Similarity for Sense Disambiguation in Cross-Language Information Retrieval, vol. 80, pp. 69–80, 2000.

[13] Fatiha Sadat, Research on Query Disambiguation and Expansion for Cross-Language Information Retrieval, *Commun. IBIMA*, vol. 2010, pp. 1–11, 2010.

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