MINISTRY EDUCATION AND TRAINING
UNIVERSITY OF DANANG

Lam Tung Giang

THE METHODS TO SUPPORT RANKING IN CROSS-LANGUAGE WEB SEARCH

Specialty : Computer Science
Code : 62 48 01 01

DOCTORAL THESIS SUMMARY

Danang - 2017

The dissertation is completed at Danang University of Technology, University of Danang.

Supervisors:

- Associate Professor **Vo Trung Hung**, PhD.
- Associate Professor **Huynh Cong Phap**, PhD.

Opponent 1: Professor **Hoang Van Kiem**, PhD.

Opponent 2: Associate Professor **Le Manh Thanh**, PhD.

Opponent 3: Associate Professor **Phan Huy Khanh**, PhD.

The dissertation was defended before the Board at UD meeting at 14h00, May 26th ,2017

PREFACE

The Cross-Language Web Search is related to the task of gathering information request given by the user in one language (the source language) and creating a list of relevant web documents in another language (the target language). Ranking in web search is related to creating the result from a search query in the form of a list of documents, ordering descending by relevance level to the information need given by the user, and assuring that "best" results appear early in the result list displayed to the user.

There are two central problems in cross-language web search. The first problem is related translation, which helps to represent queries and documents to be searched in a same environment for matching. The second problem is ranking for calculating the relevance level between documents and queries.

In cross-language information retrieval (CLIR), the dictionary-based approach is popular as a result of the simplicity and the readiness of machine readable dictionaries. The main limits of this approach include the ambiguities and the dictionary coverage. There is a need of developing techniques to improve query translation.

The web search is different from the traditional information retrieval, which has been being applied for library systems. An HTML document contains many different parts: title, summary or description, content; each part can affect differently on the search.

On the basis of literature and practical review, the topic "The methods to support ranking in cross-language web search" is selected as the research content of the Doctoral thesis, with the aim of creating a cross-language web search model and proposing technical solutions applied in the model to improve the search result ranking effectiveness.

1. The Goals, Objectives and Scope of the Research

The goal of this thesis is the proposal of 2 groups of

technical methods applied in cross-language web search. The first group is related to translation and consists of post-translation query processing, disambiguation, post-translation query processing methods. The second group includes the cross-language proximity models and a learning-to-rank model based on Genetic Programming. The main measure for the effectiveness in the thesis is the MAP (Mean Average Precision) score.

2. Thesis structure

In addition to the introduction, conclusion and future work sections, the structure of thesis contains the following chapters:

Chapter 1: Overview and research proposal.

Chapter 2: Automatic translation in cross-language information retrieval.

Chapter 3 : Techniques to support query translation.

Chapter 4: Re-ranking.

Chapter 5: The Vietnamese-English web search system.

3. Thesis Contribution

- The proposal of new methods for disambiguation in the translation module;
- The proposal of pre-translation query processing method;
- The proposal of query refining methods in the target language;
- The proposal of cross-language proximity models;
- The proposal of a learning-to-rank model based on Genetic Programming;
- The design of a Vietnamese-English web search system.

CHAPTER 1: OVERVIEW AND RESEARCH PROPOSAL

1.1. Information Retrieval

1.1.1. Introduction

1.1.2. Formal definition

1.1.3. Processing schema

There are two phases in an information retrieval system:

Phase 1: Data collection, processing, indexing and storage.
Phase II: Querying.

1.1.4. Traditional IR models

Traditional IR models include Boolean model, Vector Space model and Probabilistic model.

1.1.5. Models based on in-document term relation

The Latent Semantic Indexing (LSI) model and the proximity models are based on the relation between terms in documents.

1.2. Evaluation in Information Retrieval

1.3. Cross-language Information Retrieval

1.3.1. Introduction

Cross-language Information Retrieval concerns the case when the language of documents being searched is different from the one of the query.

1.3.2. Approaches

The main two approaches in CLIR are query translation and document translation.

1.4. Re-ranking techniques

1.5. Ranking in web-search

1.6. The limitation and research proposals

1.6.1. Limitation

The main limitations consist of the translation quality and the lack of using special structure of HTML documents in ranking.

1.6.2. Research proposals

Research missions include the proposal of two groups of techniques: (1) translation techniques to create an environment, where query representation and document representation are comparable for matching; (2) techniques for improving ranking quality of search result list. As the result, the author proposes a ranking model for cross language web search.

The following techniques are selected as research topics:

- Automatic translation techniques;

- The techniques supporting query translation, including pre-translation query processing in the source language and post-translation query optimizing in the target language;

- Learning to Rank methods;

- Building a cross-language web search system.

1.7. Summary

Research missions include the proposal of two groups of techniques: (1) translation techniques to create an environment, where query representation and document representation are comparable for matching; (2) techniques for improving ranking quality of search result list.

CHAPTER 2: AUTOMATIC TRANSLATION IN CROSS-LANGUAGE INFORMATION RETRIEVAL

2.1. Automatic translation approaches

2.2. Disambiguation in dictionary-based approach

The three main problems, which can cause low performance for a dictionary-based CLIR system, includes the dictionary coverage, the query segmentation and the selection of correct translation for each query term. The third problem is a hot research topic and is known as the disambiguation.

2.3. Dictionary-based disambiguation models

2.3.1. Variations of Mutual Information

2.3.1.1. *Based on co-occurrence statistics of pairs of words*

The common formula for calculation Mutual Information value, describing the relation of a pair of words, has the following form:

$$MI_{cooc} = \log\left(\frac{p(x,y)}{p(x) \times p(y)}\right) \quad (2.1)$$

where $p(x,y)$ is the probability of the event the two words x,y co-occur in the same sentence with the distance no more 5 words; $p(x)$ and $p(y)$ are the probabilities of the two words x and y appearing

in the document collection.

2.3.1.2. *Based on search engines*

With the two words *x* and *y*, the strings *x*, *y* and *'x AND y'* are used as queries and are sent to the search engine. The values *n*(*x*), *n*(*y*), *n*(*x*, *y*) are numbers of documents containing *x*, *y* and *x*, *y* together. Then,

$$MI_{ir} = \frac{n(x,y)}{n(x) \times n(y)} \tag{2.2}$$

2.3.2. Algorithms for selecting best translations

The algorithms in this part are executed with the Vietnamese keywords $v_1,..,v_n$ and lists of translation candidates $L_1,\ldots, L_n$ (each list $L_i = (t_1, \ldots, t_{k_i})$ contains the translation candidates of v_i).

2.3.2.1. *Algorithm using cohesion score*

2.3.2.2. *Algorithm SMI*

Each candidate translation $qtran_e$ of the given query is represented in the form $qtran_e = (e_1, ..., e_n)$, where e_i is selected from the list L_i. The function *SMI* (Summary Mutual Information) is defined as follow::

$$SMI(qtran_e) = \sum_{x,y \in qtran_e} MI(x,y) \tag{2.3}$$

The candidate translation with the highest value of the *SMI* function is selected as the English translation for a given Vietnamese query q_v.

2.3.2.3. *The algorithm SQ (Select translations sequentially)*

At first, a list from pairs of translations (t_i^k, t_{i+1}^j) of all adjacent columns (*i, i+1*) is created. At first, the two columns *i0* and *i0+1* contain the pair with highest value of MI function is selected to create the *GoodColumns* set. The best translation from neighborhood columns of the first two columns (are selected based on a *cohesion*

score function as follow:

$$cohesion(t_i^k) = \sum_{c \in GoodColumns} MI(t_i^k, t_c^{best}) \quad (2.4)$$

The column containing the best translation is added into the *GoodColumns* set. The process continues until all columns are examined. Next, the translation candidates in each column are resorted. Finally, each Vietnamese keyword is associated with a list of translations, ordering descending by cohesion score.

2.3.3. Building query translation

2.3.3.1. *Combining the two ways*

The query has the following form:

$$q = (t_1^1 w_1^1 OR\ t_1^2 w_1^2 \dots t_1^{m_1} w_1^{m_1})\ w_1\ AND \dots AND\ (t_n^1 w_n^1 OR\ t_n^2 w_n^2 \dots t_n^{m_n} w_n^{m_n}) w_n \quad (2.5)$$

2.3.3.2. *Assign weight based on result of disambiguation process*

Given $t_i^1, t_i^2, \dots t_i^{m_i}$ being translation options of keyword v_i in the list L_i with weights $w_i^1, w_i^2, \dots w_i^{m_i}$ respectively, the query in the target language has the following form:

$$q = (t_1^1 w_1^1 OR\ t_1^2 w_1^2 \dots t_1^{m_1} w_1^{m_1})\ AND \dots AND\ (t_n^1 w_n^1 OR\ t_n^2 w_n^2 \dots t_n^{m_n} w_n^{m_n}) \quad (2.6)$$

2.4. Experiment with the formula SMI

Table 2.1: Experiment results

	Configuration	***P@1***	***P@5***	***P@10***	***MAP***	***Comparison***
1	nMI	0.497	0.482	0.429	0.436	74.79%
2	SMI	0.511	0.488	0.447	0.446	76.50%
3	Google translate	0.489	0.535	0.505	0.499	85.59%
4	Manual translation	0.605	0.605	0.563	0.583	100%

2.5. Experiment with the algorithms creating structured query

2.5.1. Experimental environment

2.5.2. Experimental configurations

2.5.3. Experiment results

Table 2.2: Comparison of P@k and MAP

	Configuration	***P@1***	***P@5***	***P@10***	***MAP***	***Comp.***
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%

2.6. Summary

The chapter 2 presents author's researches related automatic translation techniques used in CLIR. The contributions include 2 methods for dictionary-based query translation. The first method defines a Summary Mutual Information function for selecting the best translation for each keyword contained in the original query. The second method is based on an algorithm of selecting best translations sequentially.

The formula SMI gives a better result in the comparison with the algorithm *Greedy*, however it is not better Google Translate tool. The algorithm of selecting best translations sequentially outperforms the Google Translate tool. The condition for applying this algorithm is the search engine should support structured queries.

CHAPTER 3: TECHNIQUES TO SUPPORT QUERY TRANSLATION

3.1. Query segmentation

3.1.1. Using the tool *vnTagger*

3.1.2. The algorithm WLQS

The algorithm WLQS (Word-length-based Query

Segmentation) - proposed by the author - splits the query into separated keywords based on the keywords lengths. The idea behind this algorithm is an author's hypothesis: if a compound word exists in the dictionary and contains other words inside, the translation of the compound word tends to be better than the translations of the words inside.

3.1.3. The combination of WLQS and *vnTagger*

This section introduces the combination of the algorithm WLQS and the tool vnTagger, consisting of 5 steps: looking words in dictionaries, assigning labels, removing words fully contained inside another word, removing overlapped words, adding remained tagged words.

3.2. Improving the query in the target language

3.2.1. Pseudo relevance feedback in CLIR

In CLIR, PRF can be applied in different stages: before or/and after translation process with the aim of improving query performance.

3.2.2. Refining the structured query in the target language

With the documents returned by the first query, the query term weights are re-calculated to build a new query in the form:

$$q' = (t_1^1 w_1^1 OR\ t_1^2 w_1^2 \dots t_1^{m_1} w_1^{m_1})\ AND$$
$$\dots AND\ (t_n^1 w_n^1 OR\ t_n^2 w_n^2 \dots t_n^{m_n} w_n^{m_n})$$

To expand the query, there are 4 different formulas to calculate new weights for keywords in the query

The formula FW1:

$$w(\mathrm{t_j}) = \frac{\lambda}{|D_r|} \times \sum_{d_i \in D_r} wd_i^j \qquad (3.1)$$

The formula FW2 combines local *tf-idf* weight and the *idf* weight of the keywords:

$$w(\mathrm{t_j}) = \frac{\lambda}{|D_r|} \times \sum_{d_i \in D_r} wd_i^j \times \log(\frac{N+1}{N_{t_i}+1}) \qquad (3.2)$$

Here, N is the number of documents in the document

collection, N_t is the number of documents containing the term t, λ is a tunable parameter.

With a term t_j and a keyword q_k, $mi(t_j,q_k)$ is the number of times the two words co-occur with the distance no more than 3 words. The formula FW3:

$$w(\mathrm{t_j}) = \lambda \times \sum_{q_k \in q} mi(t_j, q_k) \tag{3.3}$$

The formula FW4:

$$w(t_j) = \lambda \times \sum_{q_k \in q} mi(t_j, q_k) \times log(\frac{N+1}{N_{q_k}+1}) \tag{3.4}$$

By adding p terms with highest weights, the final query being created has the following form:

$$\begin{gathered} q_{final} = q' AND(expaned\ terms) = \\ = \left(w_1^1 OR\ t_1^2 w_1^2 \dots t_1^{m_1} w_1^{m_1}\right) \\ AND \dots AND\ (t_n^1 w_n^1 OR\ t_n^2 w_n^2 \dots t_n^{m_n} w_n^{m_n}) \\ AND\ e_1 w_1 \dots e_p w_p \end{gathered} \tag{3.5}$$

Where $t_i^1, t_i^2, \dots t_i^{m_i}$ are translate options of v_i in the list L_i with weights $w_i^1, w_i^2, \dots w_i^{m_i}$ respectively; $e_1, e_2, \dots, e_p$ are expanded terms being added to the query with the corresponding weights $w_1, w_2, \dots, w_p$.

3.3. Experiments

The experiment results show that the combination of the query keywords re-calculation algorithm and the query expansion helps to improve the precision and recall of the system.

3.4. Summary

The author's contribution presented in the chapter 3 includes: The query segmentation algorithm, executing in the pre-translation process by combining the WLQS algorithm and the tool *vnTagger;* the techniques for re-calculating query's terms and query expansion for the query in the target language.

CHAPTER 4: RE-RANKING

4.1. Genetic Programming for Learning to Rank

4.1.1. GP-based L2R model

The author use the dataset OHSUMED for evaluating the "*learn to rank*" method based on genetic programming. Each chromosome in GP is a ranking function *f*(*q*,*d*), measuring relevance level of documents to the query *q*. 4 options to create ranking functions are presented:

Option 1: Linear combination of 45 attributes:

$$TF - AF = a_1 \times f_1 + a_2 \times f_2 + \cdots + a_{45} \times f_{45} \quad (4.1)$$

Option 2: Linear combination of a random number of attributes:

$$TF - RF = a_{i1} \times f_{i1} + a_{i2} \times f_{i2} + \cdots + a_{in} \times f_{in} \quad (4.2)$$

Option 3: Apply a defined function to attributes. Limited with the use of functions *x*, *1/x*, *sin*(*x*), *log*(*x*), and *1/*($1+e^x$).

$$TF - FF = a_1 \times h_1(f_1) + a_2 \times h_2(f_2) + \cdots + a_{45} \times h_{45}(f_{45}) \quad (4.3)$$

Option 4: Create the function *TF-GF* with the tree structure, keeping all non-linear variants.

In the above formula, a_i are parameters, f_i are document attributes, h_i are functions. The fitness function being used is the MAP score.

4.1.2. Experiment results

4.1.3. Evaluation

The results show that methods *TF-AF*, *TF-RF* give good results. The values MAP, NDCG@k and P@k of these methods outperform the ones of the methods *Regression*, *RankSVM* and *RankBoost*, similar with some advantages in the comparison with *ListNet* and *FRank*. The method *TF-GF* gives a non-satisfied result. These results show that the use of linear function in the proposed Learning to Rank method can improve the system performance.

4.2. The proposed proximity models

The author proposes proximity models, applied in CLIR.

4.2.1. The CL-Büttcher model

4.2.2. The CL-Rasolofo model

4.2.3. The CL-HighDensity model

4.2.4. Experiments with proximity models

The following ranking functions are examined and compared:

$$s_{CL-Buttcher}(d,q) = score_{solr}(d,q) + score_{okapi}(d,q) + 10 \times score_{CL-Buttcher}(d,q) \quad (4.4)$$

$$s_{CL-Rasolofo}(d,q) = score_{solr}(d,q) + score_{okapi}(d,q) + 10 \times score_{CL-Rasolofo}(d,q) \quad (4.5)$$

$$s_{CL-HighDensity}(d,q) = score_{solr}(d,q) + score_{okapi}(d,q) + 5 \times score_{CL-HighDensity}(d,q) \quad (4.6)$$

Table 4.1: MAP score of experimental configurations

	Origin	***CL-Buttcher***	***CL-Rasolofo***	***CL-HighDensity***
top_three_ch	0.350	0.352	0.372	0.365
top_three_sq	0.370	0.375	0.397	0.389
top_three_all	0.380	0.386	0.403	0.397
Join-all	0.351	0.357	0.376	0.374
Flat	0.262	0.271	0.310	0.299
Google	0.372			
Baseline	0.381			

Table 4.2: The improvement levels of proximity models

	CL-Butcher	***CL-Rasolofo***	***CL-HighDensity***
top_three_ch	0.57%	6.29%	4.29%
top_three_sq	1.35%	7.30%	5.14%
top_three_all	1.58%	6.05%	4.47%

	CL-Butcher	***CL-Rasolofo***	***CL-HighDensity***
Join-all	1.71%	7.12%	6.55%
Flat	3.44%	18.32%	14.12%

4.3. Learning to Rank web pages

4.3.1. L2R models

The two models of Learning to Rank based on genetic programming are proposed to "learn" a ranking function in the form of a linear combination of basic ranking functions. The first model uses the training data, containing scores assigned to elements in HTML documents by basic ranking functions and labels, indicating whether a document is relevant to a query. The second model uses only assigned scores to elements in HTML documents; it compares ranking position given by a candidate ranking function with the one assigned by basic ranking functions.

4.3.2. Chromosome

With a set of n basic ranking functions F_0, F_1,…,F_n, each chromosome has the form of a linear function, combing basic ranking functions:

$$f(d) = \sum_{i=0}^{n} \alpha_i \times F_i(d) \tag{4.7}$$

With α_i are real numbers, d is the document being assigned score. The aim of learning process is select a function f given the best ranking performance.

4.3.2.1. *Fitness function*

The fitness function indicates how good each chromosome is. The fitness function in the proposed supervised L2R model is the MAP score

Algorithm 4.1: Calculation goodness (supervised)
Input: *The candidate function f, set of queries Q* ***Output:*** *goodness level if the function f*

```
begin
n = 0; sap = 0;
for each query q do
        n+=1;
        calculate score of each document assigned by f;
        ap = average precision of function f;
        sap += ap;
map = sap/n
return map
end
```

In the unsupervised L2R model, $r(i,d,q)$ is the ranking of document d in the search result list from query q, using ranking function F_i; $rf(d,q)$ is the rank of document d in the search result list from query q, using ranking function f. The algorithm is as follow:

Algorithm 4.2: Calculation goodness (unsupervised)

```
Input: The candidate function f, set of queries Q
Output: goodness level of function f
begin
s_fit = 0;
for each query q do
        calculation score of each document given by f;
        D = set of top 200 documents;
        for each document d in D do
                k+=1;d_fit = 0;
                for i=0 to n do
                        d_fit +=distance(i,k,q)
                s_fit += d_fit
return s_fit
end
```

The experiments are conducted with 3 variants of the function $distance(i,k,q)$:

Table 4.3: Variants of the function *distance*

Variant	***distance*(*i,k,q*)**
1	*abs*(*r*(*i,d,q*)-*rf*(*d,q*))
2	*abs*(*r*(*i,d,q*)-*rf*(*d,q*))/*log*(*k+1*)
3	(*r*(*i,d,q*)-*rf*(*d,q*))/ *k*

4.3.2.2. *The learning process*

4.3.3. Experimental environment

4.3.4. Experimental configurations

The learning process is examined with the following configurations:

Configuration ***SQ***: using structured query.

Configuration ***SC***: using result of supervised L2R algorithm.

Configurations ***UC1, UC2, UC3***: using result of unsupervised L2R algorithm with different variants of fitness function defined in Table 4.3.

4.3.5. Experiment results

Table 4.4: Experiment results

Configuration	***MAP***
Baseline	0.3742
Google	0.3548
SQ	0.4307
SC	0.4640
UC1	0.4284
UC2	0.4394
UC3	0.4585

4.4. Summary

Given a query in the source language, the application of techniques presented in chapter 2 and chapter 3 allows us to create and optimize a structured query in the target language as the query translation. The chapter 4 inherits these results and presents several techniques, proposed by the author, for re-ranking the list of search results. The author's contributions in this chapter include:

- Extract and index elements in web pages with the search engine to define basic ranking functions;

- Define cross language proximity functions *CL-Buttcher, CL-Rasolofo* and *CL-HighDensity* as new basic ranking functions;

- Propose Learning to Rank models in a cross language web search system, where the ranking function is "learnt" in the form of a linear combination of basic ranking functions.

The experiment results show that the proposed L2R models help to improve the system performance (measured by MAP score).

CHAPTER 5: VIETNAMESE-ENGLISH CROSS-LANGUAGE WEB SEARCH SYSTEM

The chapter 5 presents design details for a cross language Vietnamese-English web search system and the experimental results to evaluate the effects of application proposed techniques and to compare the performance with other solutions.

5.1. System design

5.1.1. System components

The main components in the system include: *pre-translation query processing, query translation, query optimization, English search and re-ranking*. These techniques have been presented in chapters 2, 3 and 4.

5.1.2. Dictionary data

5.1.3. Data to be indexed

5.2. Evaluation methods

5.3. Experiments with query translation methods

5.3.1. Experimental configuration

5.3.2. Table 5.1: Experimental configuration

Configuration	***Description***
Baseline	The query is translated manually
Google	The query is translated by the Google Translate tool
nMI	Using the disambiguation algorithm *greedy*
SMI	Using the disambiguation algorithm *SMI*

Top_one_all	Using the algorithm to select translations sequentially, extract one best translation for each keyword
Top_three_all	Using the algorithm to select translations sequentially, extract three best translations for each keyword to create a structured query translation
Top_three_weight	Using the algorithm to select translations sequentially, extract three best translations for each keyword to create a structured query translation with the weight calculated from the disambiguation process
Top-Three_flat	Using the algorithm to select translations sequentially; create a structured query by grouping translation options for each keyword with the operator OR, then grouping created groups by operator AND
Join-All	Create a structured query by grouping all translation options for each keyword with the operator OR, then grouping created groups by operator AND

5.3.3. Experiment results

Table 5.2: Comparison of P@k and MAP

Configuration	***P@5***	***P@10***	***P@20***	***MAP***	***Comp.***
Baseline	0.636	0.562	0.514	0.3838	100%
Google	0.616	0.54	0.507	0.3743	97,52%
nMI	0.5	0.464	0.418	0.269	70,09%
SMI	0.496	0.478	0.427	0.2862	74,57%
Top_one_all	0.56	0.526	0.451	0.3245	84,55%
Top_three_all	0.64	0.582	0.52	0.3924	102,24%
Top_three_weight	0.64	0.592	0.52	0.3988	103,91%

Top-Three_flat	0.592	0.556	0.499	0.3737	97,37%
Join-All	0.612	0.574	0.509	0.3865	100,70%

5.3.4. Evaluation

Among methods using only one best translation for each keyword, the configuration SMI gives a better result in the comparison with the configuration nMI. The configuration *Top_one_all* gives the best result with the translation as a structured query. The solution of creating structured query gives good results. Among the configurations using 3 best translations for each keyword, the configuration *Top_three_weight* gives the best result.

5.4. Experiments with query optimization

5.4.1. Experimental configurations

Table 5.3: Configurations to evaluate query optimization

Configurations	***Description***
Baseline	Manual translation
FW2_Top_three_all	Using algorithm *Top_three_all* for query translation. Apply the formula FW2 to modify the query.
FW2_Top_three_weight_A	Using algorithm *Top_three_weight* for query translation. Apply the formula FW2 to modify the query.
FW2_Top_three_weight_B	Using algorithm *Top_three_weight* for query translation. Apply the formula FW2 to modify the query without recalculation of keyword weights.
Top-Three_flat	Using algorithm *Top_three_all* for query translation. Apply the formula FW2 to modify the query.

5.4.2. Experiment results

Table 5.4: Comparison

Configuration	*P@5*	*P@10*	*P@20*	*MAP*
Baseline	0.636	0.562	0.514	0.3838
FW2_Top_three_all	0.640	0.586	0.522	0.4261
FW2_Top_three_weight_A	0.644	0.586	0.522	0.4192
FW2_Top_three_weight_B	0.660	0.594	0.535	0.4312
FW2_Top-Three_flat	0.652	0.586	0.520	0.4220

5.4.3. Evaluation

The table of experiment results shows that the application of query refining have a positive effect on system performance with the best result obtained by the configuration *FW2_Top_three_weight_B,* following by the configuration *FW2_Top_three_all*.

5.5. Experiments with re-ranking

The proposed re-ranking methods based on Genetic Programming are evaluated and are compared with several other Learning to Rank methods, deployed with the RankLib tool.

5.5.1. Experimental Configurations

Table 5.5: Experimental Configurations

Configuration	*Description*
SC-1	Supervised L2R; using query translation method *FW2_Top_three_all*
UC3-1	Un-supervised L2R; using query translation method *FW2_Top_three_all*
SC-2	Supervised L2R; using query translation method *FW2_Top_three_weight_B*
UC3-2	Un-supervised L2R; using query translation method *FW2_Top_three_weight_B*
MART	Using RankLib with MART method
Coordinate Ascent	Using RankLib with *Coordinate Ascent* method

Random Forests	Using RankLib with *Random Forests* method

5.5.2. Experiment results

The highest average MAP scores are obtained with 2 configurations SC-1 and SC-2. These scores are higher than average MAP scores of configurations *MART*, *Coordinate Ascent* and *Random Forests*, deployed with the RankLib tool. The configurations UC3-1 and UC3-2 give the average MAP scores of 0.456 and 0.464, which are good for un-supervised L2R methods

5.5.3. Evaluation

The experiments show the effects of re-ranking methods based on Genetic Programming. The configurations *SC-1* and *SC-2* obtain the MAP scores of 0.476 and 0.484, which are 123,96% and 126,09% respectively in the comparison with the manual translation.

The average MAP scores of configurations *UC3-1* and *UC3-2*, which do not use the training data, are 0.456 and 0.464. These scores are higher 7% and 7.7% when compared with configurations *FW2_Top_three_all* and *FW2_Top_three_weight_B* respectively.

5.6. Evaluation of proposed techniques

Table 5.6: Evaluation of proposed techniques

Configuration	***Description***	***MAP***
Baseline	Using manual translation	0.384
Google	Using Google translate tool	0.374
Query translation methods		
SMI	Using disambiguation method SMI	0.286
Top_one_all	Extract one best translation for each keyword	0.325
Top_three_all	Select translations sequentially, extract 3 best translations for each keyword and create a	0.392

	structured query	
Top_three_weight	Select translations sequentially, extract 3 best translations for each keyword and create a structured query with weights calculated from the disambiguation process	0.399
Query optimization		
FW2_Top_three_all	Use the method *Top_three_all* and apply the formula FW2 for query expansion	0.427
FW2_Top_three_weight_B	Use the method *Top_three_weight.* Apply the formula FW2. Do not re-calculate keyword weights	0.431
Learning to Rank		
UC3 FW2_Top_three_all	Using the ranking function learnt by the un-supervised method UC3 on the query created by the configuration *FW2_Top_three_all.*	0.456
SC FW2_Top_three_all	Using the ranking function learnt by the supervised method SC on the query created by the configuration *FW2_Top_three_all*	0.476
UC3 FW2_Top_three_weight	Using the ranking function learnt by the un-supervised method UC3 on the query created by the configuration *FW2_Top_three_weight.*	0.464

SC FW2_Top_three_weight	Using the ranking function learnt by the supervised method SC on the query created by the configuration *FW2_Top_three_weight.*	0.484

The methods using only one best translation for each keyword SMI and Top_one_all obtain the MAP scores of 0.286 and 0.325, which are 74,48% and 84,64% in the comparison with the *baseline* configuration.

With the methods using 3 best translation options for each keyword to create the structured query, then optimizing the query in the target language and applying "learning to rank" methods, the MAP scores are improved.

5.7. Summary

In the chapter 5, an evaluating environment is created to verify the effectiveness of proposed techniques and to compare the results with other techniques. The summary results show that the combined application of proposed techniques helps to improve the system performance (measured by the MAP score).

CONCLUSION AND FUTURE WORKS

1. Conclusion

1.1. Thesis content summary

The thesis presents author's research results on the methods for ranking search results in cross language web search. The author reviews theories and researches in the field of Information Retrieval, Cross Language Information Retrieval and Re-ranking. On the basis of processing schema of an IR system, the author introduces a model for ranking web page in cross language web search and defines the research contents.

In the chapters 2, 3 and 4, the author studies problems of query processing, automatic translation, re-ranking and proposes

techniques to solve these problems with the aim of improving the ranking performance of cross language web search systems.

In the chapter 5, an evaluating environment is created to verify the effectiveness of proposed techniques and to compare the results with other techniques. The summary results show that the combined application of proposed techniques helps to improve the system performance (measured by the MAP score)

1.2. Results

1.2.1. *Theory*

The theoretical results, proposed by the author, include 2 groups of techniques, which can be applied in components of a cross-language web search system.

The first group consists of techniques for translation: pre-translation query processing, automatic query translation and query modification in the target language:

- The algorithm WLQS is used in the combination with the open source tool *vnTagger* and serves as a tool for query segmentation. This algorithm gives the result in the form of groups of keywords and candidates translations.

- The algorithms for disambiguation are based on the definition of Mutual Information. The Summary Mutual Information algorithm is defined to select one best translation for each keyword. The second algorithm selects translations sequentially and create lists of ordered best translations for each keywords.

- Several methods of building the query translation in the target language. First, the author proposes to create a structured query based on the lists of translation candidates for each keyword. Next, the author uses the Pseudo-Relevance Feedback for recalculating keyword weights and query expansion to improve the translated query in the target language.

The second groups consists of techniques for re-ranking search result lists in cross-language information retrieval:

- Cross-language proximity models, including 2 models based on the ideas of monolingual proximity models Büttcher and Rasolofo. In the third model, the score is calculated based on BM25 scoring scheme and limited for sentences containing many query keywords. The Cross-language proximity models can be used as basic ranking functions.

- The method for re-ranking search results of web search. Within the search engine Solr, the documents are analyzed into fields and contents of each field are indexed. A set of ranking functions is defined as basic ranking functions. The Learning to Rank technique is applied to create a final ranking function to re-ranking search results.

All proposed techniques are integrated as components in a cross language web search model

1.2.2. *The experiment results*

The experiments are conducted to verify proposed techniques. The results are presented in several scientific articles:

- The experiment applying the algorithm WLQS and the function Summary Mutual Information as a disambiguation solution shows that this technique is better than the popular formula nMI in selecting best translation for keywords in a query.

- The experiment with the model in which queries are segmented by the combination of algorithm WLQS and the tool vnTagger, then the disambiguation is executed by the algorithm selecting translation sequentially and the final structured query is created in the target language outperforms the tool *Google Translate*.

- The experiment applying Pseudo Relevance Feedback to modify and expand queries in the target language shows that this technique allows to improve system performance in the precision and recall.

- An experiment is conducted to learn ranking function. From the experiments with the LETOR dataset and with defined

cross-language proximity functions, the author conducts the Learning to Rank experiment for a cross-language web search system to learn ranking functions. The experiment results show that the system performance is improved (in the MAP score).

All in all, the techniques proposed by the author help to improve the ranking performance of a cross language web search system. An important result of the thesis is the combination of these techniques as components in a cross language web search system helps to improve the ranking performance and even better the method using manual translation in conducted experiments.

2. Future works

In addition to the obtained results in the thesis, some issues may continue to be studied in the future:

- The algorithms for processing query presented in the thesis can be sensitive with languages, contents and sizes of queries. In the limited size of the thesis, the author have focused on the queries in Vietnamese and the documents to be search written in English. The queries being studied have mostly an average size (5-10 words). Other types of queries should be studied in the future.

- Optimization of query processing. The running time of pre-translation query processing, disambiguation algorithms should be optimized further in both data organization and algorithm.

- Research on other machine learning algorithms for learning other types of ranking function combination. A disadvantage of the genetic programming is the time cost. Besides, the learning process in the thesis is executed with a limited number of ranking functions. A future work can be done with more ranking functions.

THE PUBLISHED ARTICLES

[1] Giang L.T., Hùng V.T., "Các phương pháp xếp hạng lại trong trộn kết quả tìm kiếm". *Tạp chí Khoa học và Công nghệ các trường Đại học Kỹ thuật*, vol. 91, pp. 59–64, 2012.

[2] Giang L.T., Hùng V.T., "Ứng dụng lập trình di truyền trong học xếp hạng". *Tạp chí Khoa học và Công nghệ các trường Đại học Kỹ thuật*, vol. 92, pp. 58–63, 2013.

[3] Giang L.T., Hùng V.T., "Đánh giá thực nghiệm mô hình truy vấn thông tin đa ngữ". *In: Hội nghị quốc gia lần thứ VI Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin*, pp. 103–107, 2013.

[4] Giang L.T., Hung V.T., Phap H.C., "Building Evaluation Dataset in Vietnamese Information Retrieval". *Journal of Science and Technology Danang University*, vol. 12, no. 1, pp. 37–41, 2013.

[5] Giang L.T., Hung V.T., Phap H.C., "Experiments with query translation and re-ranking methods in Vietnamese-English bilingual information retrieval". *In: Proceedings of the Fourth Symposium on Information and Communication Technology*, pp. 118–122, 2013.

[6] Giang L.T., Hung V.T., Phap H.C., "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.

[7] Giang L.T., Hung V.T., Phap H.C., "Improve Cross Language Information Retrieval with Pseudo-Relevance Feedback". *In: FAIR 2015*, pp. 315–320, 2015.

[8] Giang L.T., Hung V.T., Phap H.C., "Building proximity models for Cross Language Information Retrieval". *Issue on Information and Communication Technology- University of Danang*, vol. 1, no. 1, pp. 8–12, 2015.

[9] Giang L.T., Hùng V.T., Pháp H.C., "Áp dụng học máy dựa trên lập trình di truyền trong tìm kiếm Web xuyên ngữ". *Tạp chí Khoa học và Công nghệ, Đại học Đà Nẵng*, vol. 1, no. 98, pp. 93-97, 2016.