Semantically Enhanced Uyghur Information Retrieval Model

Bo Ma
Research Center for Multilingual Information Technology, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumuqi, China
Email: yanyushu@gmail.com

Yating Yang and Xi Zhou
Research Center for Multilingual Information Technology, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, Urumuqi, China
Email: { yangyt, zhouxi }@ms.xjb.ac.cn

Junlin Zhou
Xinjiang Branch of Chinese Academy of Sciences, Urumuqi, China
Email: zhoujl@ ms.xjb.ac.cn

***Abstract*—Traditional Uyghur search engine lacks semantic information, aiming to solve this problem, a semantically enhanced Uyghur information retrieval model was proposed based on the characteristics of Uyghur language. Firstly word stemming was carried out and web pages were represented by the form of 3-triples to construct the Uyghur knowledge base, then the matching between ontologies and web pages was established by computing concept similarity and relation similarity. Semantic inverted index was built to save the association between semantic entities and web pages, and user query analysis was implemented by expanding the queries and analyzing the relations between the queries, finally by combining the benefits of both keyword-based and semantic-based methods, ranking algorithm was implemented. By comparing with the Google search engine and the Lucene based method, the experiments validate the effectiveness and the feasibility of the model preliminarily.**

***Index Terms*—Uyghur, ontology, semantic search, semantic relation, information retrieval**

I. INTRODUCTION

Along with the development of technologies and the enrichment of the resources of Internet, WWW has become a dynamic and huge information service network. Although traditional search engine is convenient, it has the problems of low precision and recall, which is caused by the lacking of semantic information of the keyword matching technology and the misunderstanding of the users' intensions. In order to provide better service, the major search engines such as Google, Bing, etc. have semantic search as a supplement to their search service.

In recent years, semantic search has been paid much attention, which introduces the semantic web technologies into traditional search engine. It combines the concepts of ontology to process the annotation with matching of web resources and users' queries to improve the search performance, and constructs the next generation of search engine [1].

Semantic search technologies can be divided into three categories, they are statistical based metrics, linguistic based metrics and ontology based metrics. Latent Semantic Analysis (LSA) is a method of statistical based metrics, which use algebraic methods to analyze the potential relations between a set of documents and terms [2]; linguistic based metrics refine the concepts in document sets by using thesaurus, one of the most used methods is describing concepts by using WordNet [3]; ontology based metrics firstly construct high-quality ontology and knowledge base, then use them to annotate documents and execute the mapping between concepts and user queries, for example, KIM, TAP, Hakia [4,5,6,7]. With the development of semantic web, a large number of structured open metadata emerge, such as Freebase, Linked Data, Apex and YAGO. Researchers began to use these metadata to build semantic search framework, for example, PowerSet uses Freebase to annotate Wikipedia.

To our knowledge, there are no semantic search prototypes for minority languages till now, in this paper, a semantically enhanced Uyghur information retrieval model is proposed, key technologies are presented and experiments are carried out to validate the effectiveness of the model.

The remainder of this paper is organized as follows: Section 2 presents the characteristics of Uyghur and the stemming algorithm for this language. In Section 3, the semantic retrieval model is proposed and the key technologies are discussed. Section 4 uses experiments to validate the effectiveness of this model. Finally, we conclude the paper in Section 5; possible extensions of the proposed model are also mentioned in this section.

Manuscript received June 23, 2011; revised November 2, 2011; accepted December 31, 2011.

This work was supported in part by the Science and technology projects of Xinjiang Uyghur Autonomous Region under Grant 201012112. Corresponding author: Ma Bo.

doi:10.4304/jsw.7.6.1315-1320

II. UYGHUR KNOWLEDGE REPRESENTATION

A. Uyghur stemming algorithm

Uyghur is an official language of Xinjiang Uyghur Autonomous Region, which is spoken by the Uyghur people. The Uyghur language belongs to the Uyghur Turkic branch of the Turkic language family, which is controversially a branch of the Altaic language family. It has an alphabet of 32 letters and more than 120 forms of characters. Uyghur is an agglutinative language, in which word is the smallest independent unit [8]. A word is composed of stem and suffix. For example, a user input a word: جۇڭگونىڭ تەرەققىياتى (development of China), the search engine should separate the word into stem and suffix, such as جۇڭگونىڭ= جۇڭگو+نىڭ , تەرەققىياتى= تەرەققىيات+ى . Then it should return web pages include stem جۇڭگو (China) and تەرەققىيات (development) and other relevant pages.

According to linguistic theory, morphemes are the smallest meaning-bearing units of language as well as the smallest units of syntax [9]. MATHIAS CREUTZ and KRISTA LAGUS have proposed a model family called Morfessor for the unsupervised induction of a simple morphology from raw text data.

The model of language (M) consists of a morph vocabulary, and a grammar. That is to compute the maximum a posteriori (MAP) estimate for the parameters [10]:

$$\arg\max_{M} P(M \mid corpus) = \arg\max P(corpus \mid M) \cdot P(M) \tag{1}$$

The MAP estimate consists of two parts: the probability of the model of language P(M) and the maximum likelihood (ML) estimate of the corpus conditioned on the given model of language, written as P(corpus |M).

$$P(M) = N! \cdot \prod_{i=1}^{N} [P(form(\mu_i)) \cdot P(usage(\mu_i))] \tag{2}$$

The probability of a morph is divided into two parts: the form and the usage of the morph, the probability of $P(usage(\mu_i))$ is equal to the prior probability of the occurrence frequency and the length of μ_i, $P(form(\mu_i))$ is computed as follows:

$$P(form(\mu_i)) = \coprod_{j=1}^{length(\mu_i)} P(c_{ij}) \tag{3}$$

Where $P(c_{ij})$ represents is the probability of the *jth* letter in the *ith* morph in the lexicon.

$$P(corpus \mid M) = \prod_{j=1}^{w} \prod_{k=1}^{n_j} P(\mu_{jk}) \tag{4}$$

W is the number of words in the corpus, where each word can be represented by morph sequence, if a word can be segmented into n_j forms, $P(\mu_{jk})$ means the probability of the *kth* morph of the *jth* word.

We firstly use morphological segmentation method to extract morphemes of Uyghur, and then the morphemes are compared with the stems and suffixes we have collected, and the corrected stems are used for extraction of Uyghur stemming, unknown morphemes are further processed by linguistic experts.

We have collected more than 25,000 Uyghur stems, which are capable for most of the Uyghur word segmentation task, and vowel harmony is processed by rule based approaches, detailed information can be referred in [1].

B. Uyghur knowledge base construction

Ontology is a "formal, explicit specification of a shared conceptualization" [11]. It is a formal representation of the knowledge by a set of concepts within a domain and the relations between those concepts. It is used to reason about the properties of that domain, and may be used to describe the domain.

In recent years, many open ontologies emerge, we are not aim to create more new ontologies, but to reuse the existing ones. The crawled pages are processed into structured data, and stored as ontology resource; the processing steps are as follows:

1) Giving every page a unique URI (Uniform Resource Identifier). For example, a Uyghur page: http://uyghur.people.com.cn/155989/15153298.html, because the URL is unique, we use http://uyghur.people.com.cn/155989/15153298 as the URI for this page;

2) Define six properties for each page, they are label, tag, content, link (used for store the URL), pagelink (used for store the hyperlinks in the page), and relatedlink (used for store the related link);

3) Store the URI and properties in N-TRIPLE format, six N-TRIPLE files were built.

After format conversion, the mapping between contents and ontology concepts were executed, we have collected ontologies covering sports, finance, entertainment, news, .etc form Swoogle and Google, the structured data along with ontologies construct our knowledge base.

C. Mapping between documents and ontologies

Mapping between documents and ontologies is the selection of ontology concepts in essence, we implemented the mapping by computing the similarity between documents and concepts and similarity between concepts themselves:

$$Sim(W,O,r) = \lambda_1 Sim_c(W,O) + \lambda_2 Sim_r(W,O,r) \tag{5}$$

Where $\lambda_1 + \lambda_2 = 1$, W means the word set of a document, O means a specific ontology which matches the document, $r = \{r, w_i, w_j\}, w_i, w_j \in W$, means the relation between the words in a document, $Sim_c(W,O)$ represents the similarity between W and O. $Sim_r(W,O,r)$ represents the similarity between concepts.

Concept similarity $Sim_c(W,O)$ is defined as follows: when a word w_i matches the name or the content of *rdfs:label* property of concept c_j, we define them as exact match, when the stem of w_i matches the name or the content of *rdfs:label* of concept c_j, we define them as partial match. $Sim_c(W,O)$ is computed as follows:

$$Sim_c(W,O)=\sum_{i=1}^{i=m}\sum_{j=1}^{j=n}Sim_c(w_i,c_j) \tag{6}$$

Where m is number of words in the document, n is the number of matched concepts in ontologies, $Sim_c(w_i,c_j)$ is computed as follows:

$$Sim_c(w_i,c_j)=\begin{cases}1 & if \quad w_i,c_j \quad completely \quad match\\ 0.5 & if \quad w_i,c_j \quad partially \quad match\\ 0 & otherwise\end{cases} \tag{7}$$

For $Sim_r(W,O,r)$, computing the similarity between words can be transformed into the computing between the corresponding concepts [12]:

$$Sim_r(W,O,r)=\sum sim(c_i,c_j,r) \tag{8}$$

$$sim(c_i,c_j,r)=\left(\sum\frac{\eta_{ij}}{l}\right)\cdot e^{-\alpha l_{\min}}\cdot\frac{e^{\beta h}-e^{-\beta h}}{e^{\beta h}+e^{-\beta h}} \tag{9}$$

Where $l_{\min}$ is the shortest path between c_i and c_j, h is the least common subsumer of c_i and c_j, η_{ij} is the number of path between c_i and c_j, l is the length of each path; α and β is constant, which is used to control the impact of l and h to the similarity. The normalizing result of the above formula is shown as follows:

$$Sim(W,O,r)'=Sim(W,O,r)\Big/\max\{Sim(W,O,r)\} \tag{10}$$

Figure 1 shows the mapping of ontologies and instances (to help understanding, we use English to show the information in the graph, in the actual system, the language is Uyghur), the top half of Figure 1 shows the internal structure of the ontology, the bottom half includes two instances of the ontology and shows the mapping between the ontology and instances.

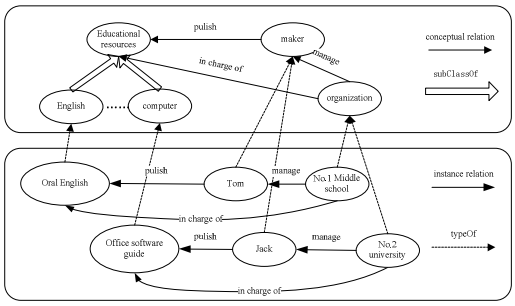

Figure 2. Ontology-instance mapping graph.

III. SEMANTIC ENHANCED RETRIEVAL MODEL

Semantic enhanced retrieval model includes four modules: resources collection, semantic annotation, query analysis and results ranking. The resources collection module uses web crawler to download relevant WebPages; semantic annotation module annotates the crawled pages and establishes the semantic index; query analysis module analyzes the semantic relevance by matching users' query and ontology concepts; and the results ranking module ranks the search results based on the content matching and semantic relevance between user input and returned pages. The architecture is shown as follows:

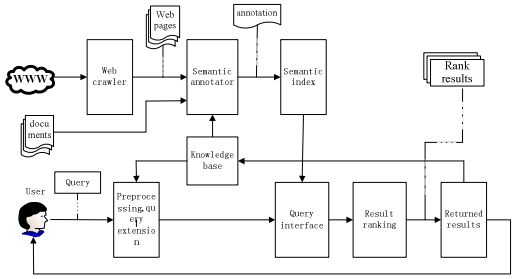

Figure 2. The Semantic enhanced retrieval model.

A. Semantic indexing

Inverted index is the most used index structure in modern search engines. In this paper, we choose it as the basic index structure, and the semantic indexing is established by combining keyword based index and semantic annotation, the steps are as follows:

1) Establish inverted index for web documents: establish mapping between the words of web documents and the concepts of ontologies, index the words along with the corresponding concepts according to the discussion of 1.3;

2) Establish inverted index for semantic entities in the knowledge base: extract the textual representation from the *rdfs:label* property of the entity, the textual representation are then searched in the document index, the retrieved documents are tagged as the potential document set *A*;

3) Extract the context of semantic entity: the ontological relations are exploited to extract its semantic context, the textual representation from the *rdfs:label* property of its directly linked entities are extracted and searched in the document index, the results are tagged as the potential document set *B*;

4) We compute the intersection between set *A* and set *B*, and the results are tagged as set *C*, which is the document set corresponding to the semantic entity;

5) Weighting annotations: the weights of semantic entities are computed as follows [13]:

$$\lambda\cdot S_d+(1-\lambda)\cdot C_d \tag{11}$$

Where λ ($0<\lambda<1$), S_d means the weight of document *d* of its own, and C_d means the weight of *d* in its context.

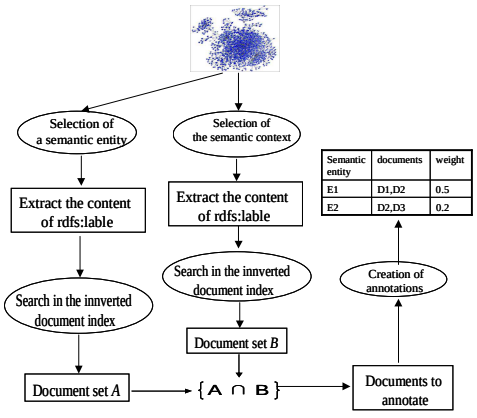

Figure 3. Semantic annotation based on context information.

The establishment of semantic index is shown in figure 3:

B. Query analyzing

The task of query analysis is to establish the mapping between query and ontology knowledge base. A given word generally has synonyms, hypernyms, hyponyms and so on, which we call lexical relation. Besides lexical relation, there is also semantic relation between words which describes the connections between concepts. In this paper, we present an approach which combines lexical relation and semantic relation to analyze user's query:

1) Accept user's input, extract the stems and save them into a set $N=\{n_1,n_2,...n_k\}$, $1\le i\le k$, k is the number of stems;

2) Analyze the lexical relation with synset, for each stem n_i $(1\le i\le k)$, acquire the extension of it and save them as a set $S_i=\{S_{i1},S_{i2},...,S_{im}\}$, $m=|S_i|$. R_{ij} is the lexical relevance between S_{ij} and n_i, $0<=R_{ij}<=1$;

3) Map the stems in S_i to ontology library, save the corresponding ontology in set $T_i=\{T_{i1},T_{i2},...,T_{if}\}$, and then we will get k sets. $R_{ij}^{'}$ is the lexical relevance between T_{ij} and n_i, assuming T_{ij} is the extension of S_{ik} $(1\le k\le m)$, then $R_{ij}^{'}$ is equal to R_{ik};

4) Given a set $W=\{(\alpha_1,\alpha_2,\ldots,\alpha_k)\,|\,(\alpha_i\in T_i,T_i\neq\phi,1\le i\le k)\cup(\alpha_i=\beta,T_i=\phi,1\le i\le k)\}$, compute the value of SR_n, which is then used for ranking the w_n $(w_n\in W)$:

$$SR_n=\alpha\sum_{i,j=1}^{i,j=k}sr_{ij}+\mu\sum_{i=1}^{i=k}r_i \tag{12}$$

Where sr_{ij} is the semantic relevance between α_i and α_j, r_i is the lexical relevance of α_i. α and μ are the weights of lexical relevance and semantic relevance $(0<=\alpha,\mu<=1,\ \alpha+\mu=1)$. The computation is as following. Given two instances $\alpha_i,\alpha_j\in w_n,\alpha_i,\alpha_j\neq\beta$, if they can reach each other within a limited number of steps by using breadth-first search, then we consider them relevant, their semantic relevance is related to the length of minimum sequence, the shorter, higher of the relevance, and vice versa. If α_i or α_j is equal to β, or they are not relevant, then $sr_{ij}=sr_{\min}$.

5) Sorting the elements in W according to the value of SR, the smaller the value is, the more possibility that it meets the user's query.

6) Searching the elements in W in semantic index, retrieved results are then ranked by the result ranking module.

C. Ranking algorithm

Page ranking is one of the key technologies of search engine, because the quality of the returned results will directly influence user's experience. When the keywords are inputted, the best results are not only contains the keywords, but also considered the relations between them. For example, when a user input the keyword "Beijing", "hotel", "Tiananmen Square", the user may want to find the information about hotels around Tiananmen Square, not the isolated information about the keywords. Because the semantic annotation we built has already considered the relations between concepts, so the retrieved results can be better organized by the ranking algorithm. Because the knowledge base we built can not cover all the concepts in the document set, we use *TF/IDF* to evaluate the uncovered concepts as a complement.

In this paper, we propose a modulative method that ranks results based on how predictable a result might be for users, which is a combination of semantic and information-theoretic techniques.

Firstly we calculate the relevance between keywords and page content:

$$I(t)=\sum tf(t\ in\ \mathrm{d})\cdot idf(t)^2\cdot boost(t.field\ \mathrm{in\ d})\cdot lengthNorm(t.field\ \mathrm{in\ d}) \tag{13}$$

Where $tf(t\ in\ \mathrm{d})$ represents the frequency of term t in document d, $idf(t)$ represents inverse document frequency, which is a measure of the general importance of the term. $boost(t.field\ \mathrm{in\ d})$ represents the stimulation factor for each field when the index is established. $lengthNorm(t.field\ \mathrm{in\ d})$ represents the length factor for each field.

Then the similarity between user's query and the documents is calculated, we define $K=\{k_1,k_2,...,k_m\},k_i,k_j\in K$ is the extension of the query, $C=\{c_1,c_2,\ldots c_n\},c_i,c_j\in C$ is the corresponding concepts of *K*, η_{ij} represents the number of the relations between c_i and c_j in the knowledge base, δ_{ij} represents the number of relations between c_i and c_j in the document context. According to the probability theory, the probability of a particular document which is interested by the user can be calculated by the formula $P(q,d)=\sum\delta_{ij}\Big/\sum\eta_{ij}\ (1\le i,j\le n,i\neq j)$, l represents the length of a path between concepts, the similarity between query and concepts are as follows:

$$SemMatch(q,d)=\sum P(q,d)\cdot(2^l)^{-1} \tag{14}$$

Now we add a search mode μ which ranging from 0 to 1, with 0 indicating purely conventional mode and 1 indicating purely semantic mode, respectively. Based on this, we build a modulative ranking model shown below:

$$SemRank=(1-\mu)I_\mu(t)\times\mu(1+SemMatch(q,d)) \tag{15}$$

IV. EXPERIMTNTAL RESULTS

10G Uyghur WebPages were crawled from Internet, and then preprocessed into N-triples; two indices were built, they were traditional index and semantic index. 10 commonly used user queries were chosen to construct the input set, and the average length of them were 4 words. Google Uyghur version and Lucene were chosen as the benchmark, and precision, recall, Mean Average

Precision (MAP) and Precision at 10 (P@10) were chosen as the evaluation metrics.

$$precision = \frac{TP}{TP + FP} \tag{16}$$

$$recall = \frac{TP}{TP + FN} \tag{17}$$

$$MAP = \frac{\sum_{r=1}^{N} (P(r) \cdot rel(r))}{N} \tag{18}$$

Where N represents the number of retrieved documents, TP is the number of returned relevant pages, FN is the number of returned irrelevant pages, and FP is the number of the relevant pages that did not returned; r represents the position of a particular page in the retrieved results, and $P(r)$ represents the precision of the topped r retrieved results, $rel(r)$ is a binary function, which determines whether the *rth* page in the retrieved documents is relevant. The experimental results are shown as follows:

From the experimental results, we can see that there are not big difference on precision and MAP for the three systems, but the result of semantic search on P@10 outperforms the other two systems, which means the pages retrieved by our proposed model can better meet users' needs.

TABLE I.
COMPARISON OF DIFFERENT SYSTEMS' P@10

Query	Semantic Search	Lucene	Google
1	0.7	0.5	0.6
2	0.5	0.3	0.5
3	0.3	0.3	0.4
4	0.6	0.5	0.5
5	0.6	0.4	0.7
6	0.4	0.2	0.2
7	0.8	0.5	0.6
8	0.1	0.2	0.1
9	0.2	0.2	0.3
10	0.35	0.3	0.4
Mean	0.47	0.34	0.43

TABLE II.
COMPARISON OF DIFFERENT SYSTEMS' PRECISION

Query	Semantic Search	Lucene	Google
1	0.74	0.58	0.66
2	0.58	0.61	0.68
3	0.42	0.46	0.51
4	0.67	0.67	0.54
5	0.56	0.61	0.47
6	0.36	0.29	0.44
7	0.67	0.52	0.56
8	0.29	0.32	0.33
9	0.34	0.38	0.41
10	0.48	0.36	0.37
Mean	0.51	0.48	0.50

TABLE III.
COMPARISON OF DIFFERENT SYSTEMS' RECALL

Query	Semantic Search	Lucene	Google
1	0.66	0.61	0.70
2	0.48	0.63	0.67
3	0.45	0.48	0.54
4	0.59	0.66	0.59
5	0.52	0.59	0.45
6	0.35	0.38	0.48
7	0.61	0.54	0.61
8	0.22	0.33	0.32
9	0.35	0.34	0.43
10	0.51	0.43	0.39
Mean	0.47	0.50	0.52

TABLE IV.
COMPARISON OF DIFFERENT SYSTEMS' MAP

Query	Semantic Search	Lucene	Google
1	0.46	0.41	0.44
2	0.28	0.29	0.31
3	0.23	0.24	0.25
4	0.37	0.34	0.33
5	0.24	0.27	0.22
6	0.22	0.18	0.23
7	0.33	0.27	0.28
8	0.18	0.19	0.19
9	0.22	0.23	0.26
10	0.33	0.24	0.30
Mean	0.29	0.27	0.28

ACKNOWLEDGMENT

Our thanks to Xi Zhou, Lei Wang, Turghun Ousiman, and all the members of the research center of multilingual information technology. This work is founded by Science and Technology project of Xinjiang Uyghur Autonomous Region titled 'Uyghur, Kazak Search Engine Retrieval Server.'

REFERENCES

[1] Bo Ma, Yating Yang, Xi Zhou, Junlin Zhou. An Ontology-based Semantic Retrieval Model for Uyghur Search Engine[C] // IEEE 2nd Symposium on Web Society(SWS2010), Beijing: 2010: 191-195.

[2] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman. Indexing by latent semantic analysis[J]. Journal of the Society for Information Science, 1990, 41 (6): 391–407.

[3] E. Vorhees. Query expansion using lexical semantic relations[C] // 17th Annual International ACMSIGIR Conference on Research and Devel- opment in Information Retrieval (SIGIR 1994), Dublin, Ireland: 1994: 61–67.

[4] A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic annotation, indexing, and retrieval[J]. Journal of Web Semantics, 2004, 2 (1): 49–79.

[5] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, KIM—a semantic platform for information extraction and retrieval[J]. Journal of Natural Language Engineering, 2004, 10 (3–4): 375–392.

[6] R.V. Guha, R. McCool, E. Miller, Semantic search[C]// the 12th International World Wide Web Conference (WWW2003), Budapest, Hungary: 2003: 700–709.

[7] http://www.hakia.com.

[8] TURDI Tohti, WINIRA Musajan, ASKAR Hamdulla.Key Technoques of Uyghur, Kazak, Kyrgyz Full-text Search Engine Retrieval Server[J]. Computer Engineering，2008，34(21)：44-46.

[9] MATTHEWS, P. H. 1991. *Morphology* 2nd Ed. Cambridge Textbooks in Linguistics.

[10] Creutz, Mathias. Unsupervised Models for Morpheme Segmentation and Morphology Learning. ACM Transactions on Speech and Language Processing 2007; 4(1).
[11] Gruber T R. Toward principles for the design of ontologies used for knowledge sharing. International Journal Human Computer Studies, 1995, 43(5-6) : 907-928
[12] WANG ZhiXiao, ZHANG DaLu.Optimization Algorithm for Edge-Based Semantic Similarity Calculation[J].PR & AI，2010，23(2)：273-277.
[13] M. Fernández, D. Vallet, P. Castells, Probabilistic score normalization for rank aggregation[C]// 28th European Conference on Information Retrieval (ECIR 2006), London, UK: 2006: 553–556.

Bo Ma was born in Liaoning Province, China in 1984. He received the B.S. degree from Huazhong University of Science and Technology in 2007. Currently he is a student and he will receive the Ph.D degree in computer science from Graduate University Chinese Academy of Sciences in 2012. He is a student member of China Computer Federation. His research interests are data mining and semantic search.

Yating Yang was born in Xinjiang Province, China in 1985. She received the B.S. degree in computer science from Chang'an University in 2007. Currently she is a student and she will receive the Ph.D degree in computer science from Graduate University Chinese Academy of Sciences in 2012. She is a student member of China Computer Federation. Her research interest is multilingual information processing.

Xi Zhou was born in Hunan Province, China in 1978. He received his M.S. degree in computer science from Graduate University Chinese Academy of Sciences in 2003. He is the leader of Research Center for Multilingual Information Technology, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences. And his research interest is multilingual information processing.

Zhou Junlin was born in Shanxi Province, China in 1945. He received his B.S. degree from Xi'an Jiaotong University. He is the vice president of Xinjiang Branch of Chinese Academy of Sciences and a supervisor of postgraduate. He is a member of China Computer Federation. And his research interest is multilingual information processing.