Design and Implementation of a SOLR Plug-in for Chinese-English Cross-Language Query Expansion Based on SKOS Thesauri

[1]Wei Sun, [2]Fabrizio Celli, [3]Ahsan Morshed, [4]Yves Jaques, [5]Johannes Keizer
[1]Institute of Agriculture Information, Chinese Academy of Agricultural Sciences, Beijing 100081, P. R. China
sunwei@caas.net.cn
[2]FAO of the United Nations, Rome, Italy
Fabrizio.Celli@fao.org
[3]FAO of the United Nations, Rome, Italy
Ahsan Morshed@fao.org
[3]FAO of the United Nations, Rome, Italy
Yves.Jaques@fao.org
[3]FAO of the United Nations, Rome, Italy
Johannes.Keizer@fao.org

Abstract. Given that existing studies for query expansion techniques for Chinese-English are relatively few and their level of standardization low, in order to improve efficiency of Chinese-English cross-language retrieval, this paper discusses the design and implementation of a SOLR plug-in for Chinese-English cross-language query expansion based on SKOS thesauri and used within the AGRIS agricultural bibliographic system. The paper also elaborates the key techniques involved in the plug-in. Finally, taking the AGRIS data resources as an example, the paper shows application examples for segmentation of mixed Chinese and English, user query parsing and AGRIS retrieval system etc., techniques that have improved the Chinese-English cross-language retrieval efficiency to a certain extent, and laid a technical foundation for research about knowledge retrieval and discovery in related fields.

Keywords: SOLR, SKOS, Linked open data, Index, Query Expansion

1 Introduction

With the rapid expansion of networked information and the continuous enrichment of multilingual information, cross-language retrieval has become a key factor in global knowledge sharing while user demand for retrieval efficiency are also escalating. Query expansion techniques can improve recall and precision rates for information retrieval, and hold out the hope of solving not only the problems of language

extension but semantic extension as well. Although methods of semantic concept expansion can make up for the inherent limitations of mechanical string expansion that focuses on query words, most of the existing related techniques cover query expansion among western languages [1-3], while few techniques cover those between Chinese and English, particularly based on standard resource models such as SKOS[4-6]. In addition, semantic and cross-language query expansion will increase the indexing workload, and the indexing quality is an important indicator to measure retrieval efficiency. Therefore, how to build a unified standard for semantic expansion technologies and how to index efficiently and qualitatively for meeting users' retrieval demands within a wider scope while improving search efficiency are still urgent problems in the field of information retrieval.

AGRIS (International Information System for the Agricultural Sciences and Technology) is a public domain Database with nearly 3 million structured bibliographical records on agricultural science and technology [7] developed by FAO (Food and Agriculture Organization of the United Nations) since 1974 to make agricultural information world-wide available. Its open database covers many aspects of agriculture, including forestry, animal husbandry, aquatic sciences and fisheries, and human nutrition, and its content is provided by more than 150 participating institutions from 100 countries. AGRIS' focus is on solving problems concerning global agricultural science and technology information resources sharing [7]. It is thus clearly useful for such a system to solve cross-language semantic retrieval problems based on a unified standard and to improve its efficiency.

Recently emerging "Linked Data" technologies can use a lightweight, scalable and extensible dynamic mechanism to achieve knowledge organization of semantic associations among dynamic, heterogeneous, and distributed data[8]. As a technical standard for resource description proposed by W3C, RDF is the basic language for Linked data, while SKOS is a common data model for sharing and linking knowledge organization systems via the Semantic Web [9]; Apache Solr is an open source search server written in Java and based on the Apache Lucene search library: it has REST-like HTTP/XML and JSON APIs that make it easy to use from any programming language [10]. Indexes created by Solr are fully compatible with the Lucene search engine library, but with more powerful full-text search capabilities and high scalability.

Based on Linked data, the paper describes the design and implementation of the SOLR plug-in for Chinese-English cross-language query expansion or Solr Query expander (SQE) - which is independent of retrieval systems - and also reveals the extended results by taking the AGRIS system as an example.

2 Retrieval System Architecture Based on SQE

To realize the SQE plug-in retrieval system based on SOLR technology, we need to begin by clarifying the data flow between the SOLR server and the retrieval system, and the role of SQE during the process of indexing and retrieval. The paper elaborates this by using the AGRIS system as an example. As shown in **Fig. 1**, the complete

retrieval in AGRIS based on SOLR technology is based on three processes: initialization, indexing, and searching.

2.1 The Initialization of SOLR

When Tomcat starts up, it initiates by reading some related configuration documents including solr.xml, solrconfig.xml, schema.xml and other related xml configuration files, and will then finish the loading of the related SOLR handler, and the configuration for the searching and indexing processes.

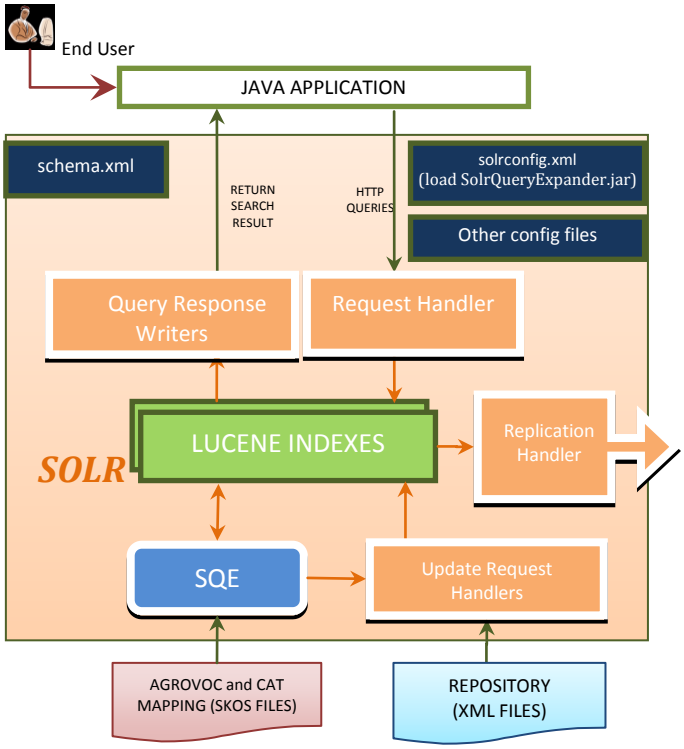

Fig. 1. AGRIS architecture based on SQE

During the process of indexing configuration, the SOLR Query Expander will index the "AGROVOC-CAT mapping (SKOS file)" which begins with a Jena analysis of the SKOS file prior to Lucene indexing. The index result is then updated by the SOLR Update Request Handler.

2.2 Indexing

During the process, first an XML document needs to be parsed and then imported into SOLR by a SOLR server update. Then, during the indexing process of the "REPOSITORY (XML FILES)" by Lucene, the SOLR Query Expander searches the indexed SKOS file and adds the expanded terms to the index, thus producing a new index via the Update Request Handler.

2.3 Searching

This process doesn't need SQE. After SOLR receives a request, it analyzes the user query with a sequence of analyzers, and then looks for the result in the constructed index to finish the search. The search result will be returned with the query response writers.

3 Logical Structure of SQE

The SQE logical structure (**Fig. 2**) contains two processes, "SKOS load and Index" and "Muti-term Expansion and Index Updating", which are marked by "①" and "②" separately.

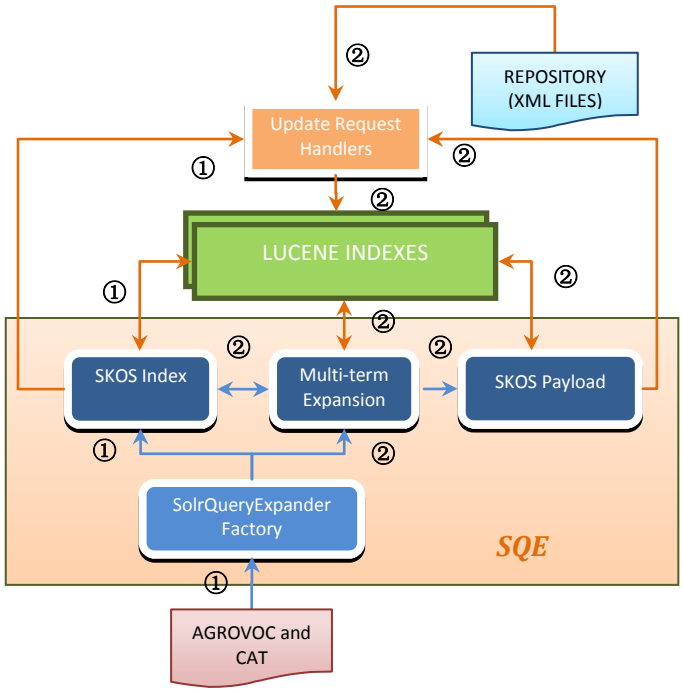

Fig. 2. Logical structure of SQE

3.1 The Logical Process of Loading and Indexing SKOS

In this process, "AGROVOC-CAT mapping (SKOS file)" is first loaded by an operation of the SOLR Query Expander Factory. This SKOS file is then indexed with Lucene by producing a SKOS engine class. The indexed result will be stored in the specific directory and updated by the Update Request Handler.

3.2 The Logical Process of Muti-term Expansion and Index Updating

In the second process, "THE REPOSITORY (XML FILES)" can be loaded, overriding the SOLR method in the SOLR Query Expander Factory. During the process of analysis of XML files, multi-term expansion takes each analyzed term as input and searches the constructed SKOS index to match concepts. If a match is found, it adds the concept's label to the constructed index. Then, it implements the index updating via operation of the SKOS payload and the Update Request Handler.

4 Key Techniques of SQE

4.1 Techniques for Producing SKOS

SQE is a plug-in based on a SKOS thesaurus document. The document is generated from the linked data version of the agricultural Chinese and English thesauri CAT and AGROVOC. Once the thesauri have been published, their related resources can be searched and browsed, and associations among them can be created with namespaces. The SKOS document contains associations between phrases in AGROVOC and CAT, which are generated by searching the SKOS with a Jena query. A fragment of the SKOS document is shown in **Fig. 3**.

```
<skos:Concept rdf:about="http://aims.fao.org/aos/agrovoc/c_5410">
        <skos:prefLabel>ornamental birds</skos:prefLabel>
        <skos:altLabel>观赏禽</skos:altLabel>
        <skos:narrower rdf:resource="http://aims.fao.org/aos/agrovoc/c_14425"/>
        <skos:broader rdf:resource="http://aims.fao.org/aos/agrovoc/c_935"/>
</skos:Concept>
<skos:Concept rdf:about="http://aims.fao.org/aos/agrovoc/c_5411">
        <skos:prefLabel>Ornamental bulbs</skos:prefLabel>
        <skos:altLabel>Flowering bulbs</skos:altLabel>
        <skos:altLabel>球根花卉</skos:altLabel>
        <skos:broader rdf:resource="http://aims.fao.org/aos/agrovoc/c_5417"/>
        <skos:related rdf:resource="http://aims.fao.org/aos/agrovoc/c_1144"/>
</skos:Concept>
```

Fig. 3. A fragment of SKOS

4.2 Techniques for Updating and Controlling Indexes

In order to meet the needs of system administrators and make cross-language retrieval, indexing, and updating of semantic expansion more flexible, the control on SQE using a SOLR configuration document can make SQE loading more flexible, so that the goals of controlling query expansion are achieved.

Fig. 4 is a part of the configuration for indexing and retrieving SOLR configuration documents. SQE plug-in controls SKOS loads with an expand variable. When the expand feature is on, SOLR will load the SKOS document during the indexing process and use SKOS-based concept expansion to update the index. On the contrary, when the expand feature is off, the SKOS file will not be loaded, and concept expansion won't be performed during indexing. Thus, in the case the SKOS document has not been changed, it can be indexed once, and not re-indexed until it has been changed. Thus the efficiency of indexing is improved to some extent.

```
<fieldType name="html_st_text" class="solr.TextField" positionIncrementGap="100">
      <analyzer type="index">
            <tokenizer class="org.wltea.analyzer.solr.IKTokenizerFactory" isMaxWordLength="false"/>
            <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt"
enablePositionIncrements="true" />
            <filter class="fao.agris.luceneSKOS.solr.SKOSFilterFactory" skosFile="ChEnmapping.rdf"
expand="on"/>
            <filter class="solr.LowerCaseFilterFactory"/>
      </analyzer>
      <analyzer type="query">
            <tokenizer class="org.wltea.analyzer.solr.IKTokenizerFactory" isMaxWordLength="true"/>
            <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt"
enablePositionIncrements="true" />
            <filter class="solr.LowerCaseFilterFactory"/>
      </analyzer>
</fieldType>
```

Fig. 4. Screenshots of SOLR configuration document

4.3 Analysis techniques of mixed Chinese-English

As can be seen from **Fig. 4** (a fragment of the SOLR configuration document), an analyzer is used during both indexing and retrieval. IK analyzer is a lightweight Chinese analyzer kit based on Java [11] [12]. IK analyzer 3.2.0 - or later versions - provides expansion for Solr 1.4 at API level. The new version of IK Analyzer uses a unique algorithm, "the most fine-grained segmentation algorithm of forward iteration". The algorithm supports two kinds of segmentation mode: the maximum word length and the fine-grained. This algorithm has a high processing speed, and also supports segmentation of letters, numbers and Chinese phrases. But it is only a Chinese analyzer kit, and it can only segment English by whitespace: so it doesn't

support mixed Chinese and English segmentation and English phrase segmentation. So, a technical issue of the analyzer is to solve mixed Chinese and English segmentation and to improve segmentation efficiency.

As the mapping database based on AGROVOC and CAT contains a large number of Chinese and English phrases, SQE plug-in development required the usage of techniques of mixed Chinese and English segmentation. Analysis of mixed Chinese and English in SQE is transformed by the IK Analyzer package, which takes the phrases in SKOS as a user dictionary, uses the "Most fine-grained segmentation method" and adds analysis of mixed Chinese and English on the basis of the IK Analyzer.

Firstly, English letters are included in pre-segmentation texts, and segmented using the most fine-grained technique by taking a letter as a unit. Then, the analysis results are filtered with a letter as a unit (**Fig. 5**). Thus, the segmentation problem of mixed Chinese and English is solved.

```
//Outputing unknown phrases by a word
   for(int i = uBegin ; i <= uEnd ; i++){
newLexeme = new Lexeme(context.getBuffOffset() , i , 1    ,
Lexeme.TYPE_CJK_UNKNOWN);
      if(!(segmentBuff[i] >= 'a' && segmentBuff[i] <= 'z')&&!(segmentBuff[i] >= 'A'
&& segmentBuff[i]<='Z') && segmentBuff[i]!=' ')
context.addLexeme(newLexeme);
  }
```

Fig. 5. Code fragment 2 of mixed Chinese and English segmentation

5 Examples demonstrating key techniques of SQE

5.1 Analysis Examples of Mixed Chinese and English

SQE plug-in is based on an analysis of mixed Chinese and English. In the AGRIS system, we use "most fine-grained segmentation" to analyze mixed Chinese and English texts, segmenting English not only by referring to white space but also based on SKOS labels. The examples of mixed Chinese and English analysis are as follows:

- Pre-analyzed text: Welcome to the Institute for Development Strategy.生产费用可以按照与产品的关系分类。
- Text analyzed: Welcome | institute | development strategy | development | strategy | 生产费用丨生产费丨生产丨费用|可以|按照|产品|关系|分类

5.2 Demonstration of Query Expansion Results

When we search “rice” in the SOLR analysis page, we will get 8 expanded terms, basmati rice, broken rice, 谷物, small grain cereals (grain), cereals, paddy, 稻米, 大米 with SQE.

5.3 Example of User Query Parsing

For different query expansion content, parsing methods are also different. Parsing techniques for user queries in AGRIS system add new and corresponding parsing content based on SOLR techniques[13]. Given differences between Chinese and English parsing techniques, parsing techniques in AGRIS system based on SQE also add new content.

In the SKOS, Chinese “生产费用” is an exactMatch term for the English “Operating costs”. Here we take the query expression related to “生产费用” and “Operating costs” as an example to illustrate the decomposed query expression result. Except for Solr query syntax, the search result should be in accordance with the decomposed query expression in **Table 1** below.

Table 1. Parsing result of user query

No.	Query Expression	Decomposed Query Expression
1	生产费用	“生产费用” or “Operating costs”
2	Operating costs	“Operating” or “costs” or “生产费用”
3	“生产费用”	“生产费用” or “Operating costs”
4	“Operating costs”	“Operating costs” or “生产费用”

In order to improve the retrieval precision, Chinese phrases are segmented using the maximum matching method, while English words and phrases are all segmented into a user query according to SOLR parsing rules.

5.4 Example of the AGRIS System

The paper has made search testes by taking Chinese and English data from 2008 to 2010 as data sources in AGRIS. Specific operations are to first search three pairs of Chinese and English corresponding phrases, and then classify statistically the search result. Initial test results are shown in **Table 2**.

Table 2. Retrieval result before and after query expansion

query	Before expansion	After expansion
Rice	5 classes, 447 results	5 classes, 464 results
大米	2 classes, 9 results	5 classes, 461 results

生产费用	0 classes, 0 results	0 classes, 0 results
Operating costs	13 results	3 classes, 54 results
生产成本	11 results	1 classes, 45 results
Production costs	5 classes, 355 results	5 classes, 943 results

As can be seen from **Table 2**, recall rate of most of the search result has been improved significantly not only from the field angle but also from the number of records retrieved. In addition, although the initial indexing time at server startup before and after expansion is 54 and 345 seconds respectively, the difference in actual search retrieval time before and after expansion is measured in milliseconds.

6 Conclusion

In summary, based on an analysis of mixed Chinese and English, the SQE plug-in designs and implementation that this paper discusses has realized Chinese-English cross-language expansion retrieval. Analysis also shows that the recall rate of retrieval is improved to a certain extent according to the search test. As SQE is based on SKOS thesauri and independent of retrieval systems, it can provide technical support for knowledge organization and discovery research in a wide variety of other situations.

The improvement of retrieval recall rate is limited by the thesauri. Although indexing is only performed at startup and the affect on retrieval efficiency is not apparent, data management efficiency can be furthered by improving indexing efficiency. Therefore, the next step is to work on index updating efficiency by improving the underlying SKOS thesauri. A second improvement could be to configure the system such that two indexes are created at startup, one expanded and one not, allowing queries to be dynamically routed to either one of the two indexes as query expansion can in some cases create large, unwanted result sets.

References

1. Vossen P. EuroWordNet a Multilingual Database with Lexical Semantic Networks. Computational Linguistics, 25(4), 628-630(1999)
2. CINDOR. http://www.cindorsearch.com/.
3. Lexical Database Homepage About WordNet. http://wordnet.princeton.edu/
4. Wu D, Wang HL. The Mechanism of Ontology Applied to Cross Language Information Retrieval. Library and information service, 9, 10-13(2006)
5. Wu D. Ontology Driven Cross Language Information Retrieval. New technology of library and information service, 5, 22-26(2006)
6. Chen SJ, Zhang YJ. English-Chinese Cross-language Information Retrieval Using Lucene System. Computer Engineering, 13, 62-64(2005)
7. AGRIS System. http://agris.fao.org/
8. Linked Data - Design Issues. http://www.w3.org/DesignIssues/LinkedData.html
9. SKOS Simple Knowledge Organization System RDF Schema. http://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html

10. Lucene Apache Solr. http://lucene.apache.org/solr/
11. ik-analyzer. http://code.google.com/p/ik-analyzer/
12. IKAnalyzer Chinese Analyzer V3.1.6 user handbook. http://wenku.baidu.com/view/b4598a5abe23482fb4da4c84.html
13. Apache Lucene - Query Parser Syntax. http://lucene.apache.org/java/2_4_0/queryparsersyntax.html