Sādhanā (2018) 43:157
https://doi.org/10.1007/s12046-018-0942-7

Ontology-based Tamil–English cross-lingual information retrieval system

D THENMOZHI* and CHANDRABOSE ARAVINDAN

Department of Computer Science and Engineering, SSN College of Engineering, Kalavakkam 603 110, India
e-mail: theni_d@ssn.edu.in; aravindanc@ssn.edu.in

MS received 15 November 2014; revised 6 July 2017; accepted 24 June 2018; published online 14 August 2018

Abstract. Cross-lingual information retrieval (CLIR) systems facilitate users to query for information in one language and retrieve relevant documents in another language. In general, CLIR systems translate query in source language to target language and retrieve documents in target language based on the keywords present in the translated query. However, the presence of ambiguity in source and translated queries reduces the performance of the system. Ontology can be used to address this problem. The current approaches to ontology-based CLIR systems use manually constructed multilingual ontology, which is expensive. However, many methods exist to automatically construct ontology for any domain in English but not in other languages like Tamil. We propose a methodology for Tamil–English CLIR system by translating the Tamil query to English and retrieve pages in English to address these issues. Our approach uses a word sense disambiguation module to resolve the ambiguity in Tamil query. An automatically constructed ontology in English is used to address the ambiguity of English query. We have developed a morphological analyser for Tamil language, Tamil–English bilingual dictionary and named entity database to translate a Tamil query to English. The translated query is reformulated using ontology and the reformulated queries are given to a search engine to retrieve English documents from the Internet. We have evaluated our methodology for agriculture domain and the evaluation results show that our approach outperforms other approaches in terms of precision.

Keywords. Cross-lingual information retrieval system; ontology; Tamil–English query translation; query expansion; semantic web.

1. Introduction

Internet provides a rich source of information and is growing at an enormous rate. English is still the dominant language in the Internet, which contributes most of the information. However, world Internet usage statistics reveal that the number of non-English Internet users is steadily increasing, but all of them are not able to formulate queries in English. Tamil users such as farmers and people working for small scale industries who are not able to express their needs in English are also growing in the Internet. They generally search for information using Tamil search engines. But the content provided by these search engines is not adequate. Making the huge repository of information on the Web, which is available in English, accessible to non-English Internet users has become an essential challenge in recent times. When the non-English users want to access the existing search engines, most of the time they formulate improper English queries.

Cross-lingual information retrieval (CLIR) systems aim to solve the afore-mentioned problem by allowing the users to express their information need in their native language while the CLIR system takes care of matching it appropriately with the relevant documents in the target language. In general, CLIR systems translate the query in source language to target language and retrieve documents in target language. When the translated query has multiple meanings, all the documents that are retrieved may not be relevant to the user. For example, the user query "payinkaal" is translated to "tiller", which has multiple meanings, namely part of a boat, agriculture equipment and name of a person. All the retrieved documents are not relevant to the user. Hence, it is necessary to include semantics into the search process to retrieve only relevant pages to the users. Also, the search process is improved by refining the queries to more specific queries. It is difficult for the Tamil users who are not able to express their needs in English to formulate such refined queries. We propose an ontology-based CLIR system that suggests possible refined queries and retrieves documents for all the queries.

Many research works have been reported for handling semantics in information retrieval (IR) using ontology [1–5]. Queries are accepted in formal languages like SPARQL in these research works. It is difficult for the users

*For correspondence

such as farmers to pose such queries. CLIR systems [6–10] facilitate non-English users to pose natural language queries in their own languages but fail to handle semantics. A few research works [11–15] have been reported on ontology-based CLIR systems that deal with semantics using bilingual ontologies. However, very few approaches are evolved to build multilingual ontologies [16–18] automatically from available resources like text documents, databases, etc. Also, many methods exist to automatically construct ontology for any domain in English but not in other languages. No such methodologies exist for learning Tamil ontology. Hence, we use a word sense disambiguation (WSD) module to resolve the ambiguity in Tamil queries during translation.

We propose a CLIR system in agricultural domain for Tamil farmers. The system retrieves relevant documents from an English corpus in response to a query expressed in Tamil language. Here, the query given in Tamil language is translated syntactically and semantically to English for IR process. The ambiguity of the translated query can be resolved by reformulating the query using an ontology. The ambiguity still persists even if we use a general purpose ontology like WordNet. For example, when we use WordNet, the query "Tiller" is reformulated as "Tiller Shoot", "Tiller Farmer", "Tiller Lever", "Tiller is part of Rudder", "Harrow Tiller", and "Tiller Farm Machine". Among these queries, "Tiller Shoot", "Tiller Lever" and "Tiller is part of Rudder" will not retrieve any pages related to agriculture equipment. Hence, it is important to use a domain-specific ontology to reformulate the queries. We use an agriculture ontology that has been learnt from text documents automatically [19].

Section 2 briefly describes various works related to ontology-based retrieval and CLIR systems. Section 3 elaborates our framework designed for cross-lingual semantic retrieval system. Section 4 provides the details of experiments conducted to analyse the performance of the proposed ontology-based CLIR system. Section 5 gives conclusion and future directions for this research.

2. Related work

IR is the process of extracting relevant information for the given query. The huge increase in the amount of information in the Internet and the complexity to reach such information caused an excessive demand for tools and techniques that can handle data semantically [2]. Ontology-based retrieval is a solution to semantic web. However, many ontology-based retrieval systems do not deal with cross-language issues. Several approaches are reported to address the cross-language issues but fail to deal with ambiguity problems. A few research methodologies have been reported that deal with both cross-language and semantic issues but have many open issues. This section reviews existing research works and open issues related to ontology-based retrieval, CLIR and ontology-based CLIR.

2.1 *Ontology-based retrieval*

Bhogal *et al* [20] and Jain and Singh [21] presented a comprehensive survey on query expansion using ontology for IR. Zimmermann *et al* [1] extended RDF framework and SPARQL language by annotating with more information for representing, reasoning and querying semantic web data. Kara *et al* [2] proposed a methodology for semantic retrieval based on domain ontology. They proved that the methodology outperforms traditional keyword-based methods and query expansion methods. However, the queries are extended based only on the class hierarchy information of the ontology, but not based on the semantic relationships of the ontology. Also, the method retrieves information only from a set of documents that are semantically indexed.

Mustafa *et al* [3] proposed an approach to ontology-based semantic IR. The query in triple form is matched with a triple in ontology and gets reformulated with the ontology terms for retrieval. They have evaluated 300 manually collected documents in the domain of research thesis. The approach does not handle incomplete and imprecise triples of the queries. Also, the approach can be extended for cross-lingual applications. Hogan *et al* [4] implemented a semantic web search engine that consists of components of IR system such as crawling, data enhancing, indexing and a user interface for search, browsing and retrieval of information. This search engine operates on RDF framework of ontology.

Fernandez *et al* [5] introduced an ontology-based approach for semantically enhanced IR. In this approach, the query is accepted in a formal SPARQL language, lists of semantic entities are returned and documents that are indexed with these semantic entities are retrieved. This IR system requires the user to be familiar with the formal languages like SPARQL. It is desirable to have a common IR system that can be used by any user who does not have formal language knowledge. Sy *et al* [22] developed a user-centred and ontology-based IR system in which the given query is reformulated either by adding or removing concepts from the query. This is done by graphically selecting the documents as interested or not interested by the user. This IR is semi-automatic due to query refinement using explicit specification of interest.

2.2 *CLIR*

Sujatha and Dhavachelvan [23] presented a survey on CLIR and multilingual information retrieval (MLIR) systems in Indian and Foreign languages. Sorg and Cimiano [6] developed a CLIR system using cross-language links of Wikipedia. The user can give query in English, French and

German languages and retrieve documents from English corpus or from German corpus. They developed a model to map bag of words that represent a document to bag of concepts using Wikipedia. They [24] extended this approach by analysing different strategies for exploiting the Wikipedia structure to define the concept space. Evaluations have been performed for both CLIR and MLIR systems for English, French, German and Spanish languages. However, ambiguity of the query in source and translated languages is not resolved in these approaches.

Several organizations in India are working on the CLIR system for Indian Languages [25]. Bandyopadhyay *et al* [8] developed a Bengali, Hindi and Telugu–English CLIR system as part of the ad-hoc bilingual task. Chinnakotla *et al* [9] developed Hindi–English and Marathi–English CLIR systems. Pingali and Varma [10] developed a Hindi and Telugu–English CLIR system. Mandal *et al* [26] developed a CLIR system for two most widely spoken Indian languages, Hindi and Bengali. All these works use bilingual dictionaries. Jagarlamudi and Kumaran [27] also worked on Hindi–English cross-lingual system in which a word alignment table was used that was learnt by a statistical machine translation (MT) system trained on aligned parallel sentences. All these research methodologies have been evaluated for English corpus of LA Times 2002. Rao and Devi [28] developed Tamil–English CLIR Track for news articles taken from "The Telegraph", English news magazine in India. All these approaches use word by word translation method in news domain.

Sivakumar *et al* [7] developed a Hindi–English CLIR system that identifies equivalent English document for the given Hindi document based on cosine similarity measure. The features of the documents to find the similarity are reduced using latent semantic indexing. This approach requires a parallel corpus that contains documents in both languages. This system works well for document queries but not for user-generated queries.

Thenmozhi and Aravindan [29] developed a CLIR for Tamil farmers using MT approach. This approach translates the Tamil query to English using a morphological analyser, bi-lingual dictionary and NE recognizer. WSD is incorporated to avoid ambiguity in Tamil to English translation. However, the methodology does not handle the ambiguity in the translated query.

2.3 *Ontology-based CLIR*

Yu *et al* [11] developed a Chinese–English CLIR system based on domain ontology. Abusalah *et al* [13] developed an Arabic–English CLIR system based on ontology for travel and tourism. Yahya *et al* [12] developed English–Malay and Malay–English CLIR systems based on Quran ontology. However, the methodologies require ontologies in both source and target languages. Construction of multilingual ontology is a time-consuming task. Ontologies are built manually in these research works. Methodologies for constructing such ontologies automatically from existing resources like text document, databases, etc. are not available. Also, the approaches do not consider the semantic relationships of the ontology to improve the retrieval performance.

Monti *et al* [14] proposed a methodology for ontology-based CLIR. Italian–English retrieval has been evaluated using this approach for archaeological domain. This approach uses ontology for source language to refine the query and then translated to the target language. However, ambiguity in the translated query is not resolved by this approach, which may occur frequently especially in English language. Pourmahmoud and Shamsfard [15] developed a Persian–English CLIR system using ontology. Bilingual ontology and dictionary are used to translate the query to the target language query. Ontology is used to disambiguate the meaning of source query to target query when the source query has multiple meanings in target query. Probabilistic approach has been used to disambiguate the target query in this research. However, suggesting more refined queries to the user is not supported by this retrieval system.

By considering several issues discussed in this section, we propose a framework for CLIR system that addresses the ambiguity in both source language and target languages to improve the retrieval performance.

3. Proposed methodology

The proposed Tamil–English CLIR system translates the given Tamil query to an English query and also suggests multiple reformulated queries for searching and retrieval using ontology. This process is depicted in figure 1.

3.1 *Morphological analysis*

The words present in the given query are transformed to their root form using a morphological analyser that uses several rules for handling plurals, case suffixes, oblique,

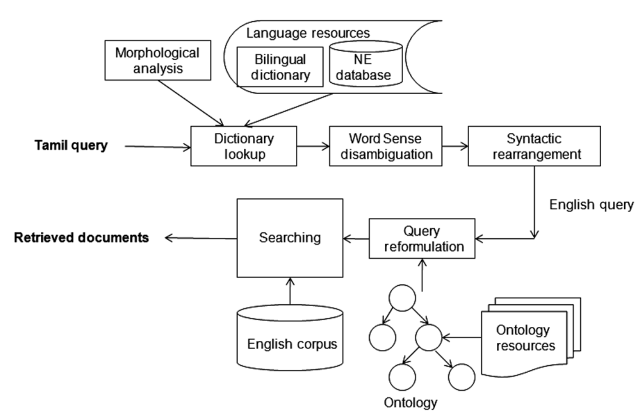

Figure 1. System architecture.

etc. The morphological analyser identifies the root form of the word and its suffixes. In Tamil, "kaL" is the major plural suffix. It has variants like "tkaL", "NGkaL", "RkaL" and "KkaL". After removing the suffix "kaL", the morphological analyser modifies the root word to get its base form by replacing "NG with m", "R with l", etc. Postpositions, namely accusative, dative, genitive, locative and plain postpositions, come next to case suffices like ai, in, il, itam, etc. The morphological analyser removes these postpositions along with case suffixes to bring the word to its root form. A list of different types of postpositions is given as follows:

- Accusative postpositions: vita, pola, kontu, nokki, patti, kuRittu, cuRRi, vittu, thavira, munnittu, venti, otti, poRuttu, poRuttavari
- Dative postpositions: aaka, enRu, mun, pin, ul, itaiye, natuve, mattiyil, veliye, mel, kizh, etiril, pakkattil, arukil, patil, maaraaka, neRaaka, uriya, ulla, takunta
- Genetive postpositions: mitu, mel, valiyaaka, mUlamaaka, vazhiyaaka, pEril, poRuttu
- Locative postpositions: irunthu, occurs only after case markers itam and il
- Plain postpositions: utan, kUta, utaiya, vacam, itam, varai, aaka, toRum, aara
- Oblique suffix: ththu.

Table 1 shows some of the compound words in Tamil and their root words along with suffixes. This analyser identifies multiples of suffixes to convert the word to its root form. For example, root word "maram" is obtained from the word "marangkaLinvazhiyaaka" (through trees) by removing multiple suffixes.

Table 1. Examples for morphological analysis.

Word	Suffix	Suffix type	Root word
puukkaL (flowers)	kkaL	plural	puu
marangkaL (trees)	ngkaL	plural	maram
naatkaL (days)	tkaL	plural	naaL
kaRkaL (stones)	RkaL	plural	kal
avanai vita (than him)	ai-vita	Accusative postposition	avan
avanukkenRu (for him)	ukk-enRu	Dative postposition	avan
kathavinmel (on the door)	In-mel	Genitive postposition	kathavu
avanitamirutnthu(from him)	Itam-irutnthu	Locative postposition	avan
viidu varai (to the house)	varai	Plain postposition	viidu
patikka (to study)	kka	Non-finite form of verb	pati
maraththu (tree)	ththu	Oblique	maram

3.2 *Dictionary look-up*

We have used a bi-lingual dictionary to obtain the English translation of the Tamil words. The dictionary look-up process uses the morphological analyser and sandhi rules to obtain the English translation of the Tamil words. The steps involved in obtaining the English translation of the query are given in Algorithm 1. The algorithm accepts a sequence of Tamil words T as input and gives a sequence of English words E. Each word in T is first searched in a named entity (NE) database to obtain its transliteration. If T is not present in the NE database, the word is searched in the dictionary to obtain its English translation. If the word is not present in the dictionary, we remove the suffix of the query term and obtain its root word using the morphological analyser. Then the root word is searched in NE database and in bilingual dictionary for its transliteration and translation, respectively. For example, for the query "ponni arisi" (ponni rice), the word "ponni" is present in NE database and is transliterated. The word "arisi" is translated as "rice" using the dictionary. However, some parts of named entities need to be translated. For example, for the query, "Maduraiyil paayum nathikaL" (rivers flow in Madurai), the term "Maduraiyil" is searched in NE database and there is no such entry in it. Then the term is searched in the dictionary and it is not found in the dictionary too. Next, the morphological analyser identifies the root term "Madurai" and its suffix "yil" for the word. Then these lexical units are searched in the NE database and in the dictionary. The term "Madurai" is transliterated using the NE database and the suffix "yil" is translated to "in" using the dictionary. If the term is not present in both NE database and dictionary after removal of all suffixes, then it is added to the target query as it is. If the word of the query has multiple meanings from the dictionary, we use WSD to obtain a single meaning. For example, for the query, "manjal valarkka ettra mann" (soil suitable for growing turmeric), the word "manjaL" has two meanings in the dictionary namely "turmeric" and "yellow". We obtain the meaning as "turmeric" using Algorithm 2. Algorithm 1 finds the sequence of set of translations E_s for all the words present in T. For example, for the query, "manjaL vaLarkka ettra mann", E_s=<{turmeric, yellow}, {grow}, {suitable for}, {soil}>. This algorithm returns the sequence of English translations as E = <turmeric, grow, suitable for, soil> using Algorithm 2.

If the root form of the word is not directly present in the dictionary, we use sandhi rules, namely, "U removal", "VY adding", "Doubling", "TR replacement" and "K-CH-TH-P" rules to split the word into two. For example, the query word "nerpayir" (paddy crop) can be transformed to "nel" "payir" using the "TR replacement" rule before obtaining the translation. If the root form of the word cannot be split, then this dictionary look-up process removes each suffix of the root form of the word until there is an entry in the dictionary to find the translation. For example, for the given word "veeLaanmai" (agriculture), meaning agriculture, the root word available in the dictionary is “veeLaan”. Removing the suffix “mai”, the translation for “veeLaan” is extracted as “agriculture”.

Algorithm 1 Query translation

Input: Sequence of words in Tamil query T
Output: Sequence of words in English query E

1: Let E_s be the sequence of set of words in English = $\emptyset$
2: **for** (each word $w_i \in T$) **do**
3:   **if** (w_i present in NE database) **then**
4:     Add its transliterated word in English to E_s
5:   **else**
6:     **if** (word w_i is present in the dictionary) **then**
7:       Add set of translations of w_i to E_s
8:     **else**
9:       Apply the morphological rules to obtain w_i' as the root word of w_i
10:       **if** (w_i' present in NE database) **then**
11:         Add its transliterated word in English to E_s
12:       **else**
13:         **if** (word w_i' is present in the dictionary) **then**
14:           Add set of translations of w_i' to E_s
15:         **else**
16:           **if** (word w_i' is not present in the dictionary) **then**
17:             Split w_i' using sandhi rules
18:             Add English translation of each segment of the word to E_s
19:             **if** w_i' cannot be split **then**
20:               Remove each suffix of w_i' until there is an entry in the dictionary
21:               Add its translated word to E_s
22:             **end if**
23:           **end if**
24:         **end if**
25:       **end if**
26:     **end if**
27:   **end if**
28: **end for**
29: Obtain sequence of words in English E using Algorithm 2

Table 2 shows some of the examples for obtaining meaning using dictionary look-up. Since the dictionary is built from the scratch as no resource is available for this domain, the system exhibits a dynamic learning approach wherein any new word that is encountered in the translation process may be added to the bilingual dictionary by allowing the user to dynamically insert into the dictionary along with its corresponding English meaning.

Table 2. Examples for query translation.

Query	Meaning from dictionary	Steps used
veeLaanmai (agriculture)	agriculture	20–21
pooni arisi (ponni rice)	ponni rice	3–4, 6–7
veeLaanmai katan thittangkal (agriculture loan plans)	agriculture loan plan	13–14
nerpayir (paddy crop)	paddy crop	16–18
manjaL vaLarkka ettra mann	{turmeric,yellow} grow	6–7
(soil suitable for growing turmeric)	suitable for soil	
veeLan (agriculture)	agriculture	6–7
Maduraiyil paayum nathikaL (rivers flow in Madurai)	in Madurai flow rivers	10–11, 13–14

3.3 *WSD*

The process for WSD is presented in Algorithm 2. When a word in the query has multiple senses, then for each sense of a given word, it is compared to all possible senses of the surrounding words in the given query. The count of number of words common between the sense descriptions is calculated and assigned as the score for the particular sense of the word. The sense that has the highest score is declared the most appropriate one for the target word in the given context. For example, the query “manjaL vaLarkka ettra mann” has ambiguous meaning for the word “manjaL”. It has two different translations namely “yellow” and “turmeric”. To disambiguate this, WordNet sense information is obtained for “yellow” and “turmeric”. After removing all the stop words, the key terms are retrieved. Similarly, key terms for the surrounding words namely “soil” and “grow” are obtained from the WordNet sense information. Surrounding words are obtained by removing the word with ambiguous meaning and the stop words. The key terms for the word with different senses and the surrounding words are listed in table 3.

The process for WSD is presented in Algorithm 2. It accepts the sequence of set of English translations E_s and gives the sequence of English translations E. For example, the algorithm accepts E_s = <{turmeric, yellow}, {grow}, {suitable for}, {soil}> as input. Surrounding words are obtained by extracting the sets of E_s with cardinality 1 and removing the words in the sets if they are stop words. Thus the surrounding words are “grow” and “soil”. Find the surrounding words sense set K_s by extracting all the words except stop words from the sense information of the words, namely grow and soil. The set e_i in E_s with a cardinality greater than one is considered to be a word with multiple meanings. For each word e_j in e_i, extract all the words except stop words from the sense information as word set K_{ej} and count the number of common elements v_{ej} between K_{ej} and K_s. For this example, we have obtained $V_{turmeric} = 1$ and $v_{yellow} = 0$. The term with maximum sense score is considered as a single term after eliminating ambiguity. Thus “turmeric” is added as an element to E. Finally, this algorithm returns the sequence of English translations as E = <turmeric, grow, suitable for, soil>.

Table 3. Keyterms from WordNet senses for query terms.

WordNet senses		Surrounding words	
Turmeric	Yellow	Soil	Grow
cultivate	yellow	plant	corn
tropical	color	grow	grow
plant	pigment	earth	forest
India	chromatic	land	mushroom
yellow	resembling	plow	tree
fower	hue	agriculture	hair
aromatic	sunflower	soil	vegetable
rhizome	ripe		backyard
source	lomon		
condiment			
dye			

Algorithm 2 Word sense disambiguation.

Input: Sequence of set of words in English E_s
Output: Sequence of words in English E

Let surrounding words sense set $K_s = \emptyset$
for (each set $e_i \in E_s$) **do**
  if ($|e_i| = 1$) **then**
    Obtain sense information of element in e_i using WordNet
    Remove stop words
    Add set of key terms to K_s
  end if
end for
Let WordNet sense score set $V = \emptyset$
for (each set $e_i \in E_s$) **do**
  if ($|e_i| > 1$) **then**
    for (each word $e_j \in e_i$) **do**
      Let WordNet sense score v_{ej} for the word e_j be 0
      Obtain sense information of e_j using WordNet
      Remove stop words
      Add set of key terms to word sense set K_{ej}
      Let $v_{ej} = | K_s \cap K_{ej} |$
      Add WordNet sense score v_{ej} to V
    end for
    Find word w with highest sense score as $w = argmax_{v \in V}(f(v))$
    Add w to E
  else
    Add e_i to E
  end if
end for
Return E

3.4 *Syntactic rearrangement*

CLIR focuses on the cross-language issues from the IR perspective rather than MT perspective [10]. However, syntactic rearrangement (SR) of the translated queries may give better search results. Also, it gives more clarity to the user about the translated query. For example, a Tamil query "udal nalaththirrku ettra payirkaL" (crops suitable for body health) is translated to English query "body health suitable for crops" in a word by word approach. The search engines retrieve "body health" related pages to the top. If the query is rearranged to "crops suitable for body health", it may give a better clarity and search result. Tamil is a subject–object–verb (SOV) language in which the sentence is present in subject, object and verb order. However, English is primarily a subject–verb–object (SVO) language. Tamil–English query translation involves identifying the individual translated words into subject, verb and object and placing them in correct order. In order to perform the translation, part of speech (POS) information is added for all the words in the dictionary. A local word reordering is performed based on POS tagging to obtain SVO pattern of English query [30].

3.5 *Query reformulation*

The translated query may be ambiguous. When the terms of the translated query (English) have ambiguous meaning, most of the pages of the search result would be irrelevant with respect to agriculture domain. It is apparent that refining the query to more specific query will improve the performance of the search result. For example, the Tamil query "mutkalappai" (harrow) is translated to English query "harrow" using this approach. When this query is given to a search engine like Google, most of the pages are not relevant to agriculture equipment. The query term "harrow" has several meanings such as area in London, school, software, actress and harrow council along with agriculture equipment. It is necessary to resolve this ambiguity. Also, refining query to more specific query in English is difficult for Tamil users. Hence, it is important to help the user with possible queries related to the given query. This refinement may be possible with the help of general purpose ontology like WordNet. However, this ontology neither refines the query to eliminate the ambiguity (i.e., it refines to more specific queries in all domains) nor performs any refinement. For example, when we use WordNet, the translated query "Plough" is reformulated as "Plough is part of Great Bear", "Asterism Plough", "Bull Tongue Plough", "Mouldboard Plough Plough" and "Plough Tool". Among these queries, only the last two queries are related to agriculture equipment. Also, WordNet does not give any related terms for the queries like "Traction Equipment". Hence, it is important to use a domain-specific ontology to reformulate the queries. We use agriculture ontology that has been learnt automatically from text documents [19] to reformulate the query and to suggest with possible refined queries in agriculture domain. A part of domain-specific ontology is shown in figure 2. The query reformulation (QR) using ontology is also useful

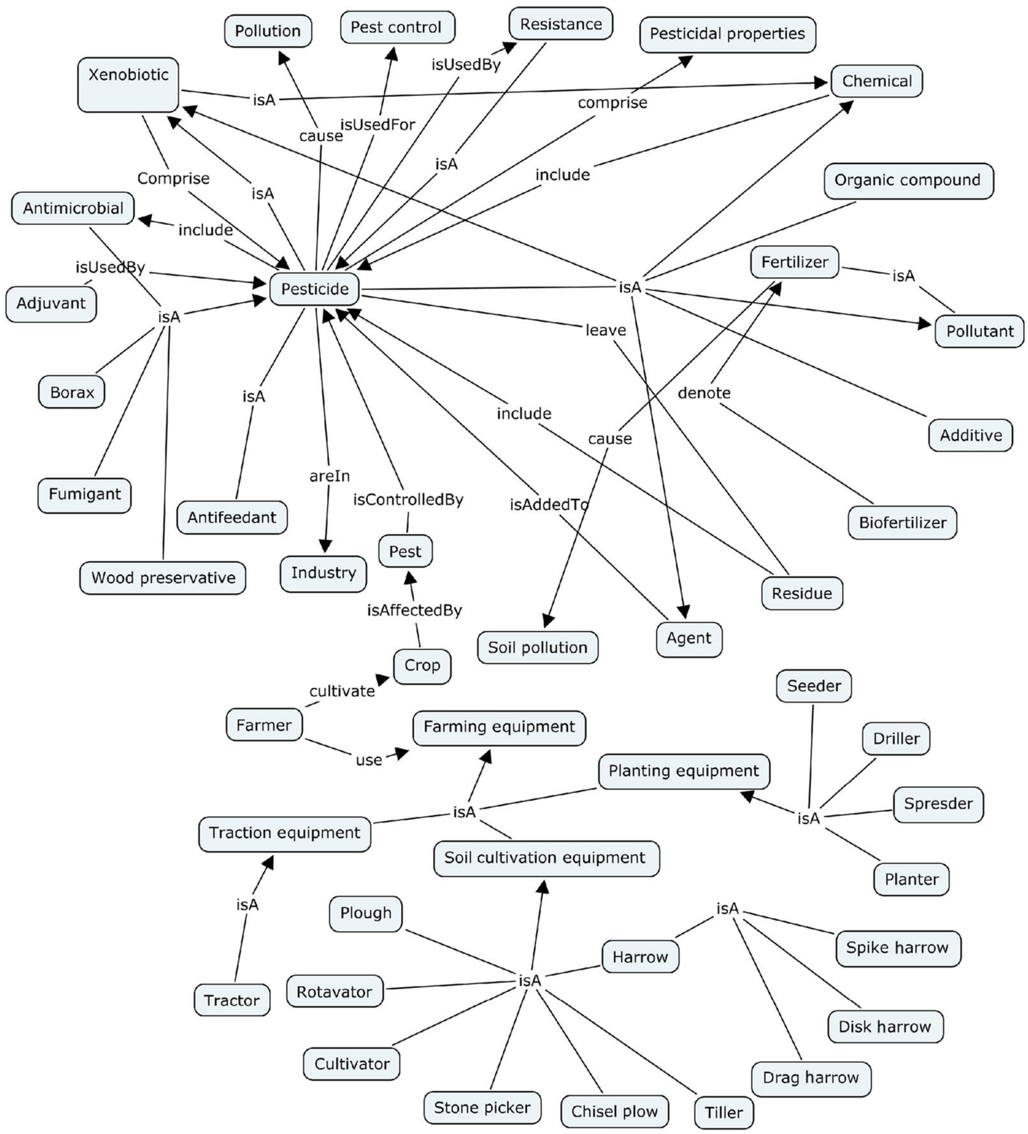

Figure 2. Agriculture ontology.

to disambiguate the source query that cannot be handled by our WSD approach. For example, let us consider the query "ManjaLin payankaL" (uses of turmeric). The term "ManjaL" has two meanings namely "yellow" and "turmeric" in the dictionary. Since, the sense of the surrounding word "payankaL" does not contribute to resolve this ambiguity, our WSD will not be helpful. In this case, both the translations, namely "uses of yellow" and "uses of turmeric", will be given to QR process where the term "turmeric" is present in the ontology. Thus the query "uses of turmeric" will be retained by refining it as "uses of turmeric + crop", and the other query is ignored for the search process.

Ontology is represented as a digraph $A = <C, R>$, where $C = \{c_1, ...c_m\}$ and $R = \{r_1...r_n\}$.

- Let q be the translated query string.
- Let S be the set of reformulated queries.
- For any concept $(c_i \in C) = q$, extract all c_j, if c_j is an adjacent node of c_i.
- Generate reformulated query set S for q by adding elements using a function for all c_j.

$$f(c_i, c_j) = \begin{cases} 1.\, c_j + c_i, & if (c_i, c_j) \mapsto r \text{ and } \\ r = \text{hierarchical relation} & \\ 2.\, c_i + r + c_j, & if (c_i, c_j) \mapsto r \text{ and } \\ r \neq \text{hierarchical relation} & \\ 3.\, c_i + c_j, & if (c_j, c_i) \mapsto r \text{ and } \\ r = \text{hierarchical relation} & \\ 4.\, c_j + r + c_i, & if (c_j, c_i) \mapsto r \text{ and } \\ r \neq \text{hierarchical relation} & \end{cases}$$

For example, let translated query $q = $ "*harrow*". Elements of reformulated query set S are

- "harrow + soil cultivation equipment" by Function 3
- "disk harrow + harrow" by Function 1
- "drag harrow + harrow" by Function 1
- "spike harrow + harrow" by Function 1

For the translated query $q =$ "*pest*", elements of S are

- "pest is control by pesticide" by Function 2
- "crop is affect by pest" by Function 4

3.6 *Searching*

The reformulated queries are converted into URLs for the search engine that is being used. The URLs are then passed on to the browser which retrieves the relevant documents from the Internet and the search results are displayed to the user.

3.7 *Walk through examples*

We consider two examples to show the significance of all the processes involved in our approach. However, all the processes are not useful for all the queries. The examples are given here.

1. Vaigai aatril uLLa miin vagaikaL (fish types present in Vaigai river).
2. ManjaLin payankaL (uses of turmeric).

Query 1: Vaigai aatril uLLa miin vagaikaL
Steps involved:

1. Tokenize the query into terms.
2. The term "Vaigai" is searched first in NE database.
3. It is found and it is transliterated. Now the resultant query term is {Vaigai}.
4. The term "aatril" is searched in NE database.
5. It is not found, hence the morphological analyser identifies the root word as "aaru" using Sandhi rule from "aatr" and the suffix "il".
6. Then the word "aaru" is searched in NE database.
7. It is not found and hence "aaru" is searched in Tamil-English dictionary.
8. Two English translations are obtained for "aaru" from the dictionary and thus we get the resultant query terms <Vaigai, {in river, in six}>.
9. Next, the third term "uLLa" is searched in NE database. It is not found and hence searched in the dictionary and the translation "present" is obtained. Thus the resultant query terms are <Vaigai, {in river, in six}, present>.
10. The fourth term "miin" is searched in NE database. It is not found and hence searched in dictionary and the translation "fish" is obtained, which results in the query terms <Vaigai, {in river, in six}, present, fish>.
11. The last term "vagaikaL" is searched in NE database. It is not found; then the suffix "kaL" is removed and the remaining word "vagai" is searched in NE database. It is not found and hence "vagai" is searched in the dictionary; there it is found as "type".
12. The translation for "kaL" is appended to "type" and thus the resultant query terms are <Vaigai, {river, six}, present in, fish, types>.
13. To perform WSD, the surrounding terms obtained after removing stop words are {vaigai, fish, types} considered for both the queries.
14. The WordNet sense information of these terms is compared to the sense information of "river" and "six". The sense score for "river" is higher than the sense score of "six", which results in the query terms <Vaigai, in river, present, fish, types>.
15. The MT process transforms the positions of the "Vaigai river" and "fish types", resulting in the query "fish types present in Vaigai river".
16. Then each term is searched in agriculture ontology for further refinement. Currently, our agriculture ontology does not contain any of the concepts present in the query.
17. The QR is not performed for this target query and the final query is "fish types present in Vaigai river", which is used for searching process.

Query 2: ManjaLin payankaL
Steps involved:

1. The term "manjaLin" is searched in NE database. It is not found; hence the suffix "in" is removed and the remaining term "manjaL" is searched in NE database, which is not present there.
2. The term "manjaL" is now searched in the dictionary and there are two translations, namely "yellow" and "turmeric", found in the dictionary.
3. The resultant query terms are <of, {yellow, turmeric}>.
4. The second word "payankaL" is search in NE database. It is not found; hence the suffix "kaL" is removed and the remaining term "payan" is searched in NE database, which is not present there, and hence it is searched in the dictionary.
5. The translation for "payan" is obtained as "use" from the dictionary. By adding the translation of the suffix "kaL", we obtain the resultant term as "uses". Thus the resultant query terms are <of, {yellow, turmeric}, uses>.
6. The WSD process removes the stopword "of" and extracts the surrounding term as "uses".
7. The sense information of "uses" is compared to the senses of "yellow" and "turmeric".
8. There is no sense score obtained for both "yellow" and "turmeric", which result in two queries "of yellow uses" and "of turmeric uses".

Table 4. Results of our approach for the queries.

Query no.	Source query	Translated query	Precision (%) Top 20
Q1	ManjaL vaLarkka ettra mann (Soil suitable for growing turmeric)	Soil suitable for grow turmeric	100
Q2	Vaigai aatril uLLa miin vagaikaL (Fish types present in Vaigai river)	Fish types present in Vaigai river	65
Q3	Maduraiyil paayum nathikaL (Rivers flow in Madurai)	Rivers flow in Madurai	95
Q4	Udal nalaththirrku ettra payirkaL (Crops suitable for body health)	Crops suitable for body health	95
Q5	ManjaL (Turmeric)	Turmeric crop	100
Q6	VeNkaaram (Borax)	Borax pesticide	95
Q7	EthiruutikaL (Antifeedants)	Antifeedants pesticide	85
Q8	Thunai marunthu poruL (Adjuvants)	Adjuvant is used by pesticide	100
Q9	ManjaLin payankaL (Uses of turmeric)	Uses of turmeric crop	100
Q10	Mutkalappai (Harrow)	1. Harrow Soil Cultivation Equipment 2. Drag Harrow Harrow 3. Disk Harrow Harrow 4. Spike Harrow Harrow	100
Q11	Uzhudhal UpakaranangkaL (Traction Equipments)	1. Traction Equipment Agriculture Equipment 2. Tractor Traction Equipment	100
Q12	Payinkaal (Tiller)	1. Tiller Soil Cultivation Equipment 2. Power Tiller Tiller 3. Rotary Tiller Tiller	100
Q13	PuussikaL (Pests)	1. Crop affect Pest 2. Pest control Pesticide	100
Q14	Kalappai (Plough)	Plough Soil Cultivation Equipment	100

9. The MT process transforms these queries to “uses of yellow” and “uses of turmeric”.
10. The QR process refines the word “turmeric” to “turmeric crop” and thus the resultant queries are “uses of yellow” and “uses of turmeric crop”.

4. Implementation and experiments

4.1 *Implementation*

We have evaluated the performance of ontology-based Tamil–English CLIR system in agriculture domain. Several queries formulated by Tamil farmers have been used to evaluate the performance of our system. The queries we have used for evaluation are of 5–6 words length. Hence, the context window for translation includes the complete query to determine for target query. We have developed a Tamil–English bilingual dictionary of size 6.08 MB that contains most of the words related to agricultural domain. We have built an NE database with 3611 entities, including 2580 place names, 132 river and lake names, and 899 person names with respect to Tamilnadu. We have collected the data from Internet[1,2,3,4] to build the NE database. We have used a rule-based morphological analyser developed for Tamil–English CLIR system [29]. This analyser is similar to Amritha’s morphological analyser [31]. Our morphological analyser finds the root term of the query and its various suffixes by handling suffixes and sandhi rules. We use agriculture ontology, which has been automatically learnt from text [19] to resolve the ambiguity in the translated queries.

[1] http://www.fallingrain.com/world/IN/25/.
[2] https://en.wikipedia.org/wiki/List_of_rivers_of_Tamil_Nadu.
[3] https://en.wikipedia.org/wiki/List_of_lakes_in_Tamil_Nadu.
[4] https://en.wikipedia.org/wiki/List_of_Tamil_people.

Table 5. Results of our experiments.

Query no.	Source query	P@20 (%)					
		E1	E2	E3	E4	E5	E6
Q1	ManjaL vaLarkka ettra mann	0	100	100	0	100	100
Q2	Vaigai aatril uLLa miin vagaikaL	45	35	65	30	35	65
Q3	Maduraiyil paayum nathikaL	95	85	95	85	85	95
Q4	Udal nalaththirrku ettra payirkaL	95	90	95	90	90	95
Q5	ManjaL	100	100	50	100	50	100
Q6	VeNkaaram	95	95	25	95	25	95
Q7	EthiruutikaL	85	85	45	85	45	85
Q8	Thunai marunthu poruL	100	100	0	100	0	100
Q9	ManjaLin payankaL	100	100	50	100	50	100
Q10	Mutkalappai	100	100	0	100	0	100
Q11	Uzhudhal upakaranangkaL	100	100	0	100	0	100
Q12	Payinkaal	100	100	10	100	10	100
Q13	PuussikaL	100	100	45	100	45	100
Q14	Kalappai	100	100	10	100	10	100
	MAP (%)	86.79	92.14	42.14	84.64	38.93	95.36

4.2 *Experiments*

The performance of any retrieval system can be analysed by the metrics precision and recall. We have not evaluated our methodology with a finite number of documents set as proposed in [12] and [15], wherein the list of relevant pages are known to measure recall. We have utilized full-text search engines like Google, which returns a huge number of pages for the queries, and hence the performance is measured in terms of only precision. Precision is calculated for top 20 pages (P@20) retrieved by the search engine that are mostly viewed by the users. Precision is measured by providing web-based user interface to the domain experts to mark the retrieved pages that are relevant to the query or not. The results of our approach for various queries are shown in table 4. We have obtained a mean average precision of 95.36% for P@20.

We have performed the following six experiments to ascertain the significance of the components, namely WSD, SR and QR, using ontologies that are employed in our approach.

E1: experiments without WSD
E2: experiments without SR
E3: experiments without QR
E4: experiments without WSD and SR
E5: experiments without SR and QR
E6: experiments with all components

The performance of all these six experiments in terms of P@20 is presented in table 5.

It is observed from table 5 that our ontology-based retrieval, which includes all the components, namely WSD, SR and QR, where the translation quality is high, gives a mean average precision of 95.36% for P@20. The translation quality is reduced due to the absence of any components used in our approach. Table 5 shows that the experiments without ontology reduce the retrieval performance to 42.14% and 38.93%. However, WSD and SR also contribute to the performance of the retrieval. The error rates for all the six experiments are 0.13, 0.08, 0.58, 0.15, 0.61 and 0.05. This shows that the error rate is very much reduced when all the components, namely WSD, SR and QR, are involved in the translation process (E6), where the translation quality is high. The SR has lesser impact in the retrieval performance (E2). However, the absence of ontology considerably increases the error rates to 0.58 (E3) and 0.61 (E5).

4.3 *Perforamance comparison of search methods*

It is evident from table 5 that ontology significantly improves the performance of the retrieval system. The ontology may be a general purpose ontology or a domain-

Table 6. Query types.

Query type	Meaning
SGT	Source query to Google Tamil
SY	Source query to Yahoo
SWU	Source query to Web Ulagam
STW	Source query to Tamil Wikipedia
TG	Translated query by Google
TCLIR	Translated query by CLIR [29]
RCLIRGO	Reformulated query by CLIR using General Purpose Ontology
RCLIRDO	Reformulated query by CLIR using Domain-specific ontology

Table 7. Performance comparison of search methods.

Query no.	P@20 (%)							
	SGT	SY	SWU	STW	TG	TCLIR	RCLIRGO	RCLIRDO
Q1	80	85	0	10	0	100	100	100
Q2	50	30	0	–	65	65	65	65
Q3	60	35	–	95	95	95	95	95
Q4	70	100	–	0	20	95	30	95
Q5	60	85	60	50	0	50	0	100
Q6	15	–	–	–	25	25	15	95
Q7	–	0	–	–	–	45	45	85
Q8	15	0	–	5	0	0	0	100
Q9	100	100	100	5	0	0	66.67	100
Q10	–	–	0	0	–	100	100	100
Q11	60	35	0	0	100	0	0	100
Q12	0	0	0	0	–	10	48.33	100
Q13	100	85	100	100	100	45	18.75	100
Q14	40	5	–	25	7.5	5	33	100
MAP	46.43	40	18.57	20.71	29.46	45.36	44.05	95.36

specific ontology. To ascertain the significance of domain-specific ontology in the retrieval performance, we have compared both variations (using general purpose and domain-specific ontology) of our ontology-based cross-lingual retrieval performance to Tamil search engines, namely Google Tamil, Yahoo, Web Ulagam and Tamil Wikipedia that use keyword search, queries translated by Google for the given query and CLIR system proposed in [29], which translates the Tamil query to English using MT approach. Table 6 shows the different query types used for comparing the search methods. The P@20 values of different search methods are summarized in table 7.

It is observed from table 7 that the performance of ontology-based CLIR using agriculture ontology is better than those of the other search methods. The results of individual queries for various search methods are given in Appendix I to show the significance of domain purpose ontology.

5. Conclusions

The proposed ontology-based CLIR system helps Tamil farmers to pose their query in Tamil and to retrieve documents from the Internet in English. The ambiguity in Tamil query is addressed using WSD. The ambiguity in the translated query is resolved using agriculture ontology, which has been learnt from text documents automatically. This CLIR system helps the user with more possible queries. These queries are semantically relevant to the given query, unlike Google, which suggest based on some keywords that are used to index the documents. We have evaluated our system by using several queries framed by Tamil farmers. We have measured the performance using the metric precision. We have performed different experiments to ascertain the importance of various components, namely WSD, SR and QR, using ontologies that are employed in our approach. We have compared our ontology-based system to conventional keyword search using several Tamil search engines, CLIR system and Google translation system. Our system outperforms the other methods in terms of mean average precision. Our system retrieves the highly ranked pages to top 20, unlike other Tamil search engines. This system can be further extended to provide a summary in English for top pages, translate the summary to Tamil or provide an answer to the query in Tamil like an expert system.

Acknowledgements

We would like to thank the management of SSN Institutions for funding the High Performance Computing (HPC) lab, where this research is being carried out. We also thank the anonymous reviewers for their constructive comments, which helped us to improve the manuscript.

Appendix I

Tables 8, 9, 10, 11, 12, 13, 14, 15 and 16 show the significance of domain-specific ontology in the retrieval performance by comparing with other search methods.

Table 8. Performance comparison for the user query "Mutkalappai".

Query	Query type	Precision (%) Top 10	Precision (%) Top 20
Mutkalappai	SGT	33.3	–
Mutkalappai	SY	20.0	–
Mutkalappai	SWU	0	0
Mutkalappai	STW	0	0
Mutkalappai	TG	–	–
Harrow	TCLIR	100	100
Disk Harrow Harrow	RCLIRGO	100	100
Harrow Cultivator	RCLIRGO	100	100
Harrow Tiller	RCLIRGO	100	100
Harrow Soil Cultivation Equipment	RCLIRDO	100	100
Drag Harrow Harrow	RCLIRDO	100	100
Disk Harrow Harrow	RCLIRDO	100	100
Spike Harrow Harrow	RCLIRDO	100	100

Table 9. Performance comparison for the user query "Uzhudhal Upakaranangkal".

Query	Query type	Precision (%) Top 10	Precision (%) Top 20
Uzhudhal Upakaranangkal	SGT	70	60
Uzhudhal Upakaranangkal	SY	70	35
Uzhudhal Upakaranangkal	SWU	0	0
Uzhudhal Upakaranangkal	STW	0	0
Tillage Equipment	TG	100	100
Traction Equipment	TCLIR	0	0
Traction Equipment	RCLIRGO	0	0
Traction Equipment Agriculture Equipment	RCLIRDO	100	100
Tractor Traction Equipment	RCLIRDO	100	100

Table 10. Performance comparison for the user query "Payinkaal".

Query	Query type	Precision (%) Top 10	Precision (%) Top 20
Payinkaal	SGT	0	0
Payinkaal	SY	0	0
Payinkaal	SWU	0	0
Payinkaal	STW	0	0
Payinkaal	TG	–	–
Tiller	TCLIR	10	10
Tiller Shoot	RCLIRGO	0	0
Tiller Farmer	RCLIRGO	100	95
Tiller Lever	RCLIRGO	0	0
Tiller is part of Rudder	RCLIRGO	0	0
Harrow Tiller	RCLIRGO	100	100
Tiller Farm Machine	RCLIRGO	90	95
Tiller Soil Cultivation Equipment	RCLIRDO	100	100
Power Tiller Tiller	RCLIRDO	100	100
Rotary Tiller Tiller	RCLIRDO	100	100

Table 11. Performance comparison for the user query "Puussikal".

Query	Query type	Precision (%) Top 10	Precision (%) Top 20
Puussikal	SGT	100	100
Puussikal	SY	90	85
Puussikal	SWU	100	100
Puussikal	STW	100	100
Insects	TG	100	100
Pest	TCLIR	10	45
Pest Epidemic Disease	RCLIRGO	80	50
Bubonic Plague Pest	RCLIRGO	0	0
Pneumonic Plague Pest	RCLIRGO	0	0
Septicemic Plague Pest	RCLIRGO	0	0
Nudnik Pest	RCLIRGO	0	0
Pest Tormentor	RCLIRGO	0	15
Vermin Pest	RCLIRGO	80	85
Pest Animal	RCLIRGO	0	0
Crops affect Pest	RCLIRDO	100	100
Pest control Pesticide	RCLIRDO	100	100

Table 12. Performance comparison for the user query "Kalappai".

Query	Query type	Precision (%) Top 10	Precision (%) Top 20
Kalappai	SGT	50	40
Kalappai	SY	0	5
Kalappai	SWU	10	–
Kalappai	STW	30	25
Plow	TG	10	10
Plough	TG	10	5
Plough	TCLIR	10	5
Plough is part of Great Bear	RCLIRGO	0	0
Asterism Plough	RCLIRGO	0	0
Bull Tongue Plough	RCLIRGO	0	0
Mouldboard Plough Plough	RCLIRGO	90	80
Plough Tool	RCLIRGO	70	85
Plough Soil Cultivation Equipment	RCLIRDO	100	100

Table 13. Performance comparison for the user query "VeNkaaram".

Query	Query type	Precision (%) Top 10	Precision (%) Top 20
VeNkaaram	SGT	20	15
VeNkaaram	SY	–	–
VeNkaaram	SWU	–	–
VeNkaaram	STW	–	–
Borax	TG	30	25
Borax	TCLIR	30	25
Mineral borax	RCLIRGO	20	15
Borax pesticide	RCLIRDO	100	95

Table 14. Performance comparison for the user query "EthiruutikaL".

Query	Query type	Precision (%)	
		Top 10	Top 20
EthiruutikaL	SGT	–	–
EthiruutikaL	SY	0	0
EthiruutikaL	SWU	–	–
EthiruutikaL	STW	–	–
EthiruutikaL	TG	–	–
antifeedants	TCLIR	50	45
antifeedants	RCLIRGO	50	45
antifeedants pesticide	RCLIRDO	90	85

Table 15. Performance comparison for the user query "Thunai Marunthu PoruL".

Query	Query type	Precision (%)	
		Top 10	Top 20
Thunai Marunthu PoruL	SGT	20	15
Thunai Marunthu PoruL	SY	0	0
Thunai Marunthu PoruL	SWU	–	–
Thunai Marunthu PoruL	STW	10	5
Adjuvant	TG	0	0
Adjuvant	TCLIR	0	0
Adjuvant	RCLIRGO	0	0
Adjuvant is used by pesticide	RCLIRDO	100	100

Table 16. Performance comparison for the user query "ManjaLin payankaL".

Query	Query type	Precision (%)	
		Top 10	Top 20
ManjaLin payankaL	SGT	100	100
ManjaLin payankaL	SY	100	100
ManjaLin payankaL	SWU	100	100
ManjaLin payankaL	STW	10	5
Uses of yellow	TG	0	0
Uses of yellow	TCLIR	0	0
Uses of yellow pigment	RCLIRGO	0	0
Uses of turmeric plant	RCLIRGO	100	100
Uses of turmeric food	RCLIRGO	100	100
Uses of turmeric crop	RCLIRDO	100	100

References

[1] Zimmermann A, Lopes N, Polleres A and Straccia U 2012 A general framework for representing, reasoning and querying with annotated semantic web data. *Web Semant. Sci. Serv. Agents World Wide Web* 11: 72–95

[2] Kara S, Alan Ö, Sabuncu O, Akpınar S, Cicekli N K and Alpaslan F N 2012 An ontology based retrieval system using semantic indexing. *Inf. Syst.* 37(4): 294–305

[3] Mustafa J, Khan S and Latif K 2008 Ontology based semantic information retrieval. In: *Proceedings of the 4th International IEEE Conference on Intelligent Systems, IS'08*, vol. 3, pp. 2214–2219

[4] Hogan A, Harth A, Umbrich J, Kinsella S, Polleres A and Decker S 2011 Searching and browsing linked data with SWSE: the semantic web search engine. *Web Semant. Sci. Serv. Agents World Wide Web* 9(4): 365–401

[5] Fernández M, Cantador I, López V, Vallet D, Castells P and Motta E 2011 Semantically enhanced information retrieval: an ontology-based approach. *Web Semant. Sci. Serv. Agents World Wide Web* 9(4): 434–452

[6] Sorg P and Cimiano P 2008 Cross-lingual information retrieval with explicit semantic analysis. In: *Working Notes for the CLEF 2008 Workshop*

[7] SivaKumar A P, Premchand P and Govardhan A 2011 Indian languages IR using latent semantic indexing. *Int. J. Comput. Sci. Inf. Technol.* 3: 245–253

[8] Bandyopadhyay S, Mondal T, Naskar S K, Ekbal A, Haque R and Godhavarthy S R 2008 Bengali, Hindi and Telugu to English ad-hoc bilingual task at CLEF 2007. In: *Advances in Multilingual and Multimodal Information Retrieval*, pp. 88–94

[9] Chinnakotla M K, Ranadive S, Damani O P and Bhattacharyya P 2008 Hindi to English and Marathi to English cross language information retrieval evaluation. In *Advances in Multilingual and Multimodal Information Retrieval*, pp. 111–118

[10] Pingali P and Varma V 2007 IIIT hyderabad at CLEF 2007-adhoc Indian language CLIR task. In *Working Notes for the CLEF 2007 Workshop*

[11] Yu F, Zheng D, Zhao T, Li S and Yu H 2006 Chinese-english cross-lingual information retrieval based on domain ontology knowledge. In: *Proceedings of the 2006 IEEE - International Conference on Computational Intelligence and Security*, vol. 2, pp. 1460–1463

[12] Yahya Z, Abdullah M T, Azman A and Kadir R A 2013 Query translation using concepts similarity based on quran ontology for cross-language information retrieval. *J. Comput. Sci.* 9(7): 889–897

[13] Abusalah M, Tait J and Oakes M 2009 Cross language information retrieval using multilingual ontology as translation and query expansion base. *Polibits* (40): 13–16

[14] Monti J, Monteleone M, Buono M P and Marano F 2013 Natural language processing and big data—an ontology-based approach for cross-lingual information retrieval. In: *Proceedings of the 2013 IEEE-International Conference on Social Computing (SocialCom)*, pp. 725–731

[15] Pourmahmoud S and Shamsfard M 2008 Semantic cross-lingual information retrieval. In: *Proceedings of the 23rd IEEE - International Symposium on Computer and Information Sciences, ISCIS'08*, pp. 1–4

[16] Navigli R and Ponzetto S P 2012 Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *J. Comput. Sci.* 193: 217–250

[17] Nastase V and Strube M 2013 Transforming wikipedia into a large scale multilingual concept network. *Artif. Intell.* 194: 62–85

[18] Xu R, Gao Z, Pan Y, Qu Y and Huang Z 2008 An integrated approach for automatic construction of bilingual Chinese–English wordnet. In: *Proceedings of ASWC 2008: The Semantic Web*, pp. 302–314

[19] Thenmozhi D and Aravindan C 2016 An automatic and clause based approach to learn relations for ontologies. *Comput. J.* 59(6): 889–907

[20] Bhogal J, Macfarlane A and Smith P 2007 A review of ontology based query expansion. *Inf. Process. Manag.* 43(4): 866–886

[21] Jain V and Singh M 2013 Ontology based information retrieval in semantic web: a survey. *Int. J. Inf. Technol. Comput. Sci.* 5(10): 62–69

[22] Sy M F, Ranwez S, Montmain J, Regnault A, Crampes M and Ranwez V 2012 User centered and ontology based information retrieval system for life sciences. *BMC Bioinf.* 13(Suppl 1): S4

[23] Sujatha P and Dhavachelvan P 2011 A review on the cross and multilingual information retrieval. *Int. J. Web Semant. Technol.* 2(4): 115–124

[24] Sorg P and Cimiano P 2012 Exploiting wikipedia for cross-lingual and multilingual information retrieval. *Data Knowl. Eng.* 74: 26–45

[25] Majumder P, Mitra M, Parui S K and Bhattacharyya P 2007 Initiative for Indian language IR evaluation. In: *Proceedings of the First International Workshop on Evaluating Information Access (EVIA)*, Tokyo, Japan, May 15

[26] Mandal D, Dandapat S, Gupta M, Banerjee P and Sarkar S 2007 Bengali and Hindi to English cross-language text retrieval under limited resources. In: *Working Notes for the CLEF 2007 Workshop*

[27] Jagarlamudi J and Kumaran A 2008 Cross-lingual information retrieval system for Indian languages. In: *Proceedings of the Advances in Multilingual and Multimodal Information Retrieval Workshop*, pp. 80–87

[28] Rao T P R K and Devi S L 2013 Tamil English cross lingual information retrieval. In: *Multilingual Information Access in South Asian Languages*, pp. 269–279

[29] Thenmozhi D and Aravindan C 2009 Tamil–English cross lingual information retrieval system for agriculture society. In: *Proceedings of the Tamil Internet Conference*, pp. 173–178

[30] Popovic M and Ney H 2006 POS-based word reorderings for statistical machine translation. In: *Proceedings of the International Conference on Language Resources and Evaluation*, pp. 1278–1283

[31] Menon A G, Saravanan S, Loganathan R and Soman K 2009 Amrita morph analyzer and generator for tamil: a rule based approach. In: *Proceedings of the Tamil Internet Conference*, pp. 239–243