ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2015.0144

GRAMMAR RULE BASED INFORMATION RETRIEVAL MODEL FOR BIG DATA

T. Nadana Ravishankar[1], Dinesh Mavaluru[2] and R. Jayabrabu[3]
[1]*Department of Computer Science and Engineering, B.S. Abdur Rahman University, India*
E-mail: nadanaravishankar@gmail.com
[2]*College of Computing and Informatics, Saudi Electronic University, Kingdom of Saudi Arabia*
E-mail: dineshmavaluru@gmail.com
[3]*Department of Information Systems, College of Computer Science and Information Systems, Jazan University, Kingdom of Saudi Arabia*
E-mail: jayabrabu@gmail.com

Abstract

Though Information Retrieval (IR) in big data has been an active field of research for past few years; the popularity of the native languages presents a unique challenge in big data information retrieval systems. There is a need to retrieve information which is present in English and display it in the native language for users. This aim of cross language information retrieval is complicated by unique features of the native languages such as: morphology, compound word formations, word spelling variations, ambiguity, word synonym, other language influence and etc. To overcome some of these issues, the native language is modeled using a grammar rule based approach in this work. The advantage of this approach is that the native language is modeled and its unique features are encoded using a set of inference rules. This rule base coupled with the customized ontological system shows considerable potential and is found to show better precision and recall.

Keywords:

Information Retrieval, Big Data, Cross Language Information Retrieval, Query Disambiguation, Telugu

1. INTRODUCTION

Cross-language Information Retrieval (CLIR) is defined [1] as the problem of finding content that are expressed in a language different from that of the user query language. A wide range of approaches to Cross Language Information Retrieval [2], [3], [4] involve some sort of direct mapping between terms in each language, either from source to target or from target to source. These approaches show poor performance [5] when applied in Indian languages due to the inflectional, agglutinating nature of the languages and the large number of word forms. Thus the effectiveness of the approaches is dependent largely on the expressive nature of the ontology and the preprocessing system. This work relies on the modeling of the language using linguistic grammar rules. Thus if new terms or phrase forms are encountered, the grammar system is used to predict and preprocess them. This improves the recall and precision of the system.

The rest of this paper is organized as follows. Section 2 reviews the salient prior work on Big data Cross Language Information Retrieval. Section 3 gives a brief overview of Telugu language and describes how the grammar rule based system is built. Section 4 discusses the methodology of query disambiguation stage. Section 5 discusses the operation of the system. Section 6 describes the implementation and the results. Section 7 concludes the work with a summary of our findings and a discussion of issues that could be productively explored in future work.

2. LITERATURE REVIEW

In this section the relevant literature in Big data CLIR and specifically Telugu IR are examined.

2.1 PREVIOUS WORK IN TELUGU IR/ISSUES IN TELUGU CROSS LANGUAGE INFORMATION RETRIEVAL

In early research work on [6] Cross Language Information Retrieval in Indian Languages is a common approach was to replace each query term with the translations found in a bilingual dictionary. In this case only one translation is possible, this works as well as no matter what. But when different numbers of translations are known for different terms the bilingual dictionary approach suffers from an unhelpful imbalance because common terms often have many translations. In [7] a canonical method was proposed to estimate term specificity in essence the same way is done when stemming is employed in same language retrieval (i.e., any document term that can be mapped to the query term is counted). It reduces the term weights for query terms that have at least one translation that is a common term in the document language, which empirically turns out to be a reasonable choice. Learned translation probabilities are used rather than translations found in a dictionary [8]. This Learned translation probabilities from parallel corpora is used to developed statistical machine translation. Word nets are used to generate the target query to include all possible translations of each term, which would increase recall at the cost of precision [9]. In the case of phrases and idioms it loses their meaning when translated word for word and it leads to unexpected results. It is clear that to obtain better results, query disambiguation is required. In general, search queries contain two or three terms, due to which disambiguation might not be possible in some cases. But, in many queries, the search terms should be mutually disambiguating. [10] Summaries that simple conjunction and disjunction (the Boolean AND and OR operators) be enough to disambiguate terms in most cases.

It is observed [11] that simple techniques such as limiting the translated term to the same part of speech, and including phrase translations improves CLIR performance meaningfully. Query term disambiguation better achieved by using thesauri or ontologies, such as wordnets, which encode associative and hierarchical relationships between terms [12]. Thesaurus can also be used for query expansion, either by automatically adding synonyms, or through user feedback by presenting appropriate thesaurus entries to the user. Query terms can be broadened (e.g. from “bus” to “vehicle”) or narrowed (e.g. from “bus” to

"transport") using hyponymy relations in thesauri. Some experiments have used existing Machine Translation systems to translate queries [13]. As might be expected, this works well only when the queries are long, which is not really typical. Also, this might not be efficient, because the Machine Translation system would take much more time for translation than a simple dictionary or thesaurus lookup would. In [14] parts of speech taggers are used to labels the words in a given user query after tagger divides every phrase into several members as names and verbs it is easy to lookup in any available resource for translation. In order to solve the problem derived by word polysemy (a single term can have several meanings). In the last few years language grammar rules are evaluated following several criteria [15], have been widely used for helping users to disambiguate the user query during their searches showing information relevant with respect to the user context of interest. In [16] the experiments are done to retrieve documents written different from the user query language (Telugu). A method for automatically learning transliteration model from a sample of name pairs in two languages is done. However, this model faced the problem of translating Names and Technical Terms from English to Kannada/Telugu [17]. The investigation research in [18] shows that the Indian Language Information Retrieval systems face severe recall problems when using conventional Information Retrieval techniques. In this research work they crawled the Web extensively for Indian languages, characterized the Indian language web and in the process came up with some solutions for the low recall problem. The results in [19] shows the effects of lexical analysis on Marathi monolingual search over the news domain corpus and observe the effect of processes such as lemmatization, inclusion of suffixes in indexing and stop-words elimination on the retrieval performance. Significantly it improved the retrieval performance of languages like Marathi which is agglutinative in nature. The use of ASCII characters to represent Indian language alphabets is also useful in computer applications where local language tools such as email and chat are not yet available fully and this transliteration scheme as a possible standard for Indian language transliteration [20].

2.2 JUSTIFICATION FOR THE PROPOSED APPROACH

It is clear from the above that a good CLIR system will have a morphological analyzer, ontology for disambiguation, grammar rules for prediction and matching and an effective information retrieval system. This is the focus of this work.

The user query is disambiguated using a lookup in the ontology and reconstructed using language grammar rules. Thus, the scalability issues are improved and the need to constrain the vocabulary for a domain is not needed. The usage of grammar rules overcomes the problem of query ambiguity. The translation to the source language is started once the query is finalized. The translation is user assisted and named entity issues are overcome. In the search stage, the disambiguated query is sent to the search engine via the application programming interface in both the source language and the target language. This widens the scope of the search and generates a larger subset of results for re-ranking. The role of the re-ranking algorithm is crucial. Thus, the entire set of results is shown only in the source language of the users. The work has been specifically developed for Telugu using the grammar rules for the language.

The specific objectives of this work are to demonstrate a methodology for Cross lingual information retrieval which addresses the issues of user query ambiguity in query processing using ontology and grammar rules to improve the relevancy of the retrieved content and presents the final outcome in a user friendly manner.

3. GRAMMAR MODELING FOR THE TELUGU LANGUAGE

This section presents an outline of Telugu and its grammar rules relevant for the Big data CLIR task.

3.1 OVERVIEW OF TELUGU LANGUAGE

Telugu is one of the major and second most popular languages in India [20]. Its literature goes back to 11th century A.D. Telugu alphabet are syllables; therefore, it should be rightly called a syllabary and most appropriately a mixed alphabetic-syllabic script. In the Telugu alphabets the correspondence between the symbols (graphemes) and sounds (phonemes) is more or less exact. However, there exist some differences between the alphabet and the phonemic inventory of Telugu. The overall pattern consists of 60 vowels, 3 vowel modifiers and 41 consonants. Telugu writing system is a combination of syllabic alphabet in which all consonants have an inherent vowel. Diacritics, which can appear above, below, before or after the consonant they are used to change the inherent vowel. When they appear at the beginning of a syllable, vowels are written as independent letters. When certain consonants occur together, special conjunct symbols are used which combine the essential parts of each letter.

3.2 TELUGU GRAMMAR

In Telugu language morphology plays an important role in not only generating various word forms from nouns and verbs but determining their shapes as well. In noun phrases, nouns carry distinct morphological variations indicating various syntactic and semantic functions expressed in proposition. Unlike English, word order, does not determine the syntactic relations between a noun and its governing category verb. Nouns in Telugu normally carry the markings of gender, number, person and case. A noun in Telugu is inflected in a complex way. A number of nouns in Telugu often change their form before the marking of gender, number, and person and case. Systematic changes occur in the base particularly when inflected for non-nominative cases such as accusative, dative, instrumental, ablative and locative. Conventionally noun-nominative base of a noun is also known as oblique base or oblique form. However, it should be noted that such a base is neither unique nor common. Though the inflection classes are insensitive to gender distinctions, there are distinctions of gender discernible from morphology of agreement on verbs, adjectives, possessives, predicate nominal, numerals and deictic categories.

It is necessary to identify the distinctions in gender like males and females. A number of nouns denoting human males end in "–du", and human females end in "–di". In two numbers Telugu nouns usually occur, singular and plural. However, only plural nouns are explicitly marked. In case of large number of nouns the

form of the plural suffix is "–lu", while in case of some nouns of human male category, the form of plural suffix alternant is "–ru". Telugu nouns when function as nominal predicate show agreement with the gender, number and person of the surface subject of the clause. Pronominalized possessive nouns (possessors) show agreement (in gender, number and person) with the nouns of possession and function as heads of possessive phrases. In these two cases nouns are marked by pronominal suffixes of the relevant gender number person. The person marking on nouns is however, explicit only in 1st and 2nd person both singular and plural, In the case of 3rd person, only the number is marked explicitly and not the person. Nouns are usually inflected by case by case markers and post-positions to indicate their semantic syntactic function in clausal predication. The terms case markers and post-positions roughly correspond to Type-1 and Type-2 post-positions of Krishnamurti and Gwynn. They use the term post-positions corresponds in meaning to prepositions in English. However, they makes a distinction between two types of post-positions, viz. Type-1 and Type-2 based on the criteria like the freedom of distribution (bound and free) and the nature of composition of post-positions (Type-1 post-positions are attached to Type-2 post-positions and not vice-versa).

Telugu uses a wide variety of case markers and post-positions and their combinations to indicate various relations between nouns and verbs or nouns. Case suffixes and post-positions fall into two types "Grammatical" and "Semantic or location and directional". Grammatical case suffixes are those which express grammatical case relations such as nominative, accusative, dative, instrumental, genitive, commutative, vocative and causal. The semantic cases include such as nouns inflected for location in time and space. Nouns when attached with various combinations of adverbial nouns and case markers or post-positions express many more such relations.

In Telugu, verb denotes the state of or action by a substance and may be finite or non-finite. All finite verbs and some non-finite verbs can occur according to situation before the utterance final stage characterized by of following terminal contours: rising pitch, meaning question; level pitch, falling pitch, meaning command. A finite verb does not occur before any of the non-final stages. On the morphological level, no non- finite verb contains a morpheme indicating person; this statement should not, however, be taken to mean that all finite verbs necessarily contain a morpheme indicating person. Since any verb, finite or non-finite, occurs only after some marked stage, by definition of these stages, all verbs have phonetic stress or prominence on their first syllable, which invariably part of the root. Almost every Telugu verb has a Finite and a non- finite form. A finite form is one that can stand as the main verb of a sentence and occur before a final pause. A non-finite form cannot stand as a main verb and rarely occurs before a final pause.

Based on the above, a set of grammar rules have been coined for the CLIR task. These are discussed next.

3.3 GRAMMAR RULES

In Telugu the eight finite forms of the verb may be arranged in three structural types, which are set up according to the differences in the grouping of the three substitution classes like stem or inflection root, tense-mode suffix and personal suffix (es). For example the finite forms of a simple verbal base are discussed below under the three structural types: atladu (to play), with two allomorphs: atla- before a vowel.

Type 1: Stem or Inflection root

1. **Imperative:**

   Singular – du example : atla – du

   Plural – andi example: atla – andi

Type 2: Tense – mode suffix

2. **Admonitive or abusive:**

   In this case due to semantic restrictions, many verbs cannot occur in this mode. A few verbs like kAlu (to burn), kUlu (to fall), cAvu (to die), pagulu (to break), etc., will come under this mode.

3. **Obligative (in all persons): -Ali**

   Example:

   atlad –Ali (I, We, You) (singular, plural)

Type 3: Personal suffix (es)

4. **Habitual- future or non-past: -ta-**

   Example:

   atla – ta – Anu I shall play
   atla – ta – Am We shall play
   atla – ta – Ava You will play
   atla – ta – Aru They will play
   atla – ta – Adu He shall play
   atla – tun – di She will play
   atla – ta – Ay They play

5. **Past tense: -din-**

   Example:

   atla – din – Anu I played
   atla – din – Am we played
   atla – din – Ava you played (Singular)
   atla – din – Aru you played (plural)
   atla – din – Adu he played
   atla – din – di she/ it played
   atla – din – Aru they played

6. **Hortative: -da-**

   Example:

   atla – da – tAm let us play, or we shall play

7. **Negative tense: -data-**

   Example:

   atla – data – nu I (do, did, and shall) not play
   atla – data – m we (do, did, and shall) not play
   atla – data – va you (do, did, and shall) not play
   atla – data – Du he (does, did, and shall) not play
   atla – data – du she/ it (do, did, and shall) not play
   atla – data – ru they (do, did, and shall) not play

8. **Negative imperative or prohibitive: -Ak-**

   Example:

   atla – Ak – u you (sg.) don't play
   atla – Ak – andi you (pl.) don't play

In the same way Non Finite Verbs are ten verbs which may be arranged into two structural types like Unbound and Bound.

Unbound type:

1. Present participle - dutu atla - dutU playing
2. Past participle – di atla- di having played
3. Concessive - dinA atla-dinA even though played
4. Conditional - itE atla- itE if played
5. Infinitive - ta atla- ta to play
6. Negative participle - aku - atla- aku not playing
7. Habitual adjective - dE atla- dE that plays
8. Past adjective - dina atla- dina that played
9. Negative adjective - dani atla- dani not played

Bound type:

10. Present – ta - atladu - ta - occurs with any finite form of the verb un- to be and also a few non- finite forms.

Example:

atladu- ta- unnAnu I am playing

atladu - ta- un- nA even playing (now)

atladu - ta- un- tE if playing

atladu - ta- un- na that playing

The grammar rules are used to preprocess the text. The idea is to identify the appropriate word sense in the text. This helps avoid the issues of out of vocabulary text. If the user query is a complex one the reordered sentence will be sent to the morphological analyzer to identify the tense of a verb and inflections that are adding to verb. But the morphological structure of Telugu verbs inflects for tense, person, gender, and number. The nouns inflect for plural, oblique, case and postpositions.

The structure of verbal complex is unique and capturing this complexity in a machine analyzable and generate able format is a challenging task. Inflections of the Telugu verbs include finite, infinite, adjectival, adverbial and conditional markers. The verbs are classified into certain number of paradigms based on the inflections. For computational need In Telugu language there are 37 paradigms of verb and each paradigm with 160 inflections and sixty seven paradigms are identified for Telugu noun. Each paradigm has 117 sets of inflected forms. Based on the nature of the inflections the root words are classified into groups.

A database with all morphological information has been prepared. So the machine by itself captures all the morphological rules. Morphological analysis of nouns is less complex compared to verbs. Some of the Paradigms and the possible inflections of the verbs and nouns are given in the Table.1:

Table.1. Inflection Table

Name of the tense	Inflections considered for Telugu verbs
Present Tense Markers	tunnA, TunnA, tunTE, TunTE, Tum~m, tU , TU, to~m, To~m
Past Tense Markers	nnA, sunnA, A, sA, DA, cA, ppA, lcA, slA, tA, LLA, TTA, ccA, kunnA, kua~m, ia~m, ccA, ia~mcA, se, de, ce, ppe, te, ue, rce, nne, ye
Future Tense Markers	TA, ddA, A, tA, tua~m, ia~mcu, su, u, cu, ccu, dcu
Clitic	vO, nO, rO, dO, lO, lA, kO, sai, si, stu, akA, nnA, lE
Auxiliary Verbs	nivvu, vaccu, valayu, pO, ua~mdu, cUdu, peTTu, pArEyi, veyyi, avvu, mugia~mcu, cUpu,daluvu, manu, cupia~mcu, veLLu, goTTu, beTTu, sAgu, tIru
Negative Markers	aka, akua~mDA, akpoyinA, akapotE, a, akpotEnE, akunnA
Pronouns	vanni, aTTua~mdi, naTTua~mdi
Nouns	ammA, ayyA, nakkara, annamATa, nEkkara
Adjective	anavasara~m
Adverbial Adjective	a~mduku, a~mduvalana, a~mduna, aTuva~mTi, aTlu, aTlugA
Post Positions	lOga, lOpuna, dAkA, koddi, kadA, gAni, kanuka,kadu, gUDA, kAbOlu, kAni, gAdA, annA, kUDA, mua~mdu, ni, a~mTA, a~mTE, aMTu, mAku, baTTi, gAni, kUDa, mAllE, mari, gala, bO, lA, sariki, dagu nua~mDu, galugu, joccu, jAlu, baDuvu, tappa, pATiki, varaku, ka~mTE
Imperative Suffix	a~mDi, lEa~mDi
Imperative Negative Suffix	aka~mDi

4. METHODOLOGY

The overall approach consists of four interlinked components for Big data Cross Language Information Retrieval. The approach consists of modules for: (a) query preprocessing (b) information retrieval from the web (c) Re-ranking and (d) content presentation. The overall methodology of the approach is shown in Fig.1. In the query processing stage, the user query is tokenized into keywords and using the Stanford parser grammatical tree structure and parts of speech (POS) category for the given English sentence is done. Once the POS is identified then it would be reordered into Subject-Object-Verb (SOV) format. To overcome this problem reordering rules are applied in the source language level. If the user query is a simple one or two means it will go for ontology and relevant suggestions generated for the user. If the query is complex the grammar based processing system will be used for the overall processing.

Once the user query is disambiguated and then it would send to the web for results. The retrieved results will be re-ranked

finally, the content selected by the user from the results is transformed by using a summarization – smoothening approach and the results shown to the users in the target language itself the overall interaction is used in the interaction stage for implicit rule formation that helps refine the order of query suggestion generation and result generation. The various components that implement the approach are shown in Fig.2. There are three major components: query processing stage, results re-ranking stage and results smoothening stage.

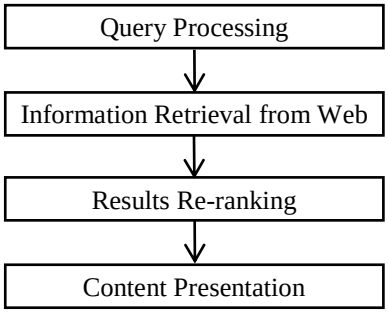

Fig.1. Overall Process

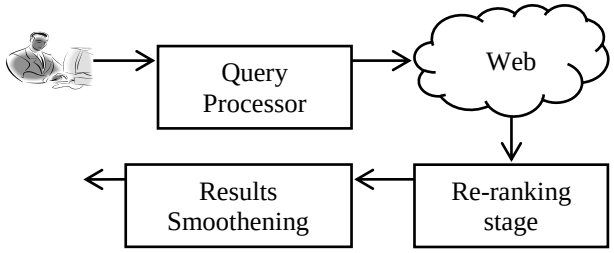

Fig.2. Overall Process

4.1 QUERY DISAMBIGUATION

The query processing system receives the complex query in the user language (Telugu). The objective of the query processing component in the system is to resolve ambiguity, handle out of vocabulary (OOV) words that are not present in the ontology and proper nouns. The outcome of the query processing system is to generate grammatically disambiguated user query that are closely related to the query in both the source and target language. For this, the language grammar rules are crucial. In this stage the morphological word generator re structure the user query and find the subject object and verb to construct the query in other language. In the preprocessing stage a tokenizer and stop word remover are used. The stop words list is a customized set of words, which are commonly present in snippets. In this work a set of stop words are taken. Preprocessing reduces the size and number of the input documents considerably, which is essential in any information retrieval system. Preprocessing removes all types of stop words, special characters, extensions, etc., to reduce the processing overhead created by including the stop words into the system's preprocessing framework. The lexical analysis is used to divide a stream of characters into a stream of words. Here in this research vector space model is used to calculate weight of a query term is given as:

$$w_{ik} = tf_{ik} \times idf_k$$

where, tf_{ik} is the number of occurrences of term *tk* in queries "*i*" and "idf_k" is the inverse frequency of the term t_k in the collection of possible terms. Commonly used measure for the inverse document frequency is:

$$idf_k = \log(N / n_k)$$

where, N is the total number of possibilities in the collection, and n_k is the number of results which contains a given term. Rule-based query disambiguation relies much on the phrase dictionary, and sometimes on ontology, to determine the correct translation of an ambiguous word. Once the user query processed and indicates the search for the web, the disambiguated query in Telugu and English are sent to the search engine.

4.2 RE-RANKING PROCESS

Once the query is sent to the web and a list of result set is formed, the re-ranking process computes a semantic similarity value between the query and each snippet, as follows: Let O is the set of all classes and instances in the ontology, and R is the set of all results in the search space. Let q to Q be an disambiguated query, let V_q be the set of variables in the SELECT clause of q, let w be the weight vector for these variables, where for each $v \in V_q$, $w_v \in [0,1]$. Let $T_q \in O_j$, j be the list of tuples in the query result set, where for each tuple $t \in T_q$ and each $v \in V_q$, $t_v \in O$. here each snippet in the search space is represent as a result vector $r \in R$, where r_x is the weight of the annotation of the document with concept x for each $x \in O$, if such annotation exists, and zero otherwise. The extended query vector as given by $q_x = W_v$, i.e., the query vector element corresponding to x is added the variable weight w_v if there is a tuple t, where $t_v = x$ (even if there is more than one such tuple, w_v is not added more than once for the same v and x). Note that the sum rarely has more than one term since this would mean that the same instance appears as a satisfying value for different variables in different (or the same) result set tuples.

4.3 SMOOTHENING PROCESS

The resultant snippets in English are taken one at a time. The basic unit of the process is to identify the root words of each term in the snippet. First the snippets are delineated in terms of sentences. Sentences are classified into simple and complex based on the structure. A simple sentence is one which follows the subject verb object form. All other sentences are complex sentences. For each sentence the terms are identified into – clauses and stop words. A clause is a verb/adverb/adjective. The stop words are identified from the sentences. The terms are converted into the root word using porter's stemming algorithm. Now language specific rules are applied to identify the translation heuristics. A single term may exist in different tense and word forms. Hence the query specific information tree sequence is used to disambiguate the sense of the term. Now, morphological rules are applied to get the translation for known grammar forms and terms. Out of Vocabulary terms are treated in the same manner as Proper nouns. Such terms are transliterated automatically. The resultant effect is of an imperfect translation as of now. In future the approach will be improved by the use of concept maps and automated translation systems.

5. OPERATION OF APPROACH

Here a user query is a search term given in the local language (target language).

Step 1: The user query is given to the query processor. The search term can be a proper noun or any sentential form.

Step 2: The query processor searches the query in the ontology for the meaning and related terms.

Step 3: If the related terms for the query are found in the Ontology, the terms are shown to the user. If the user query is a phrase or a complex sentence then

Step 4: The user query is sent for morphological process based on grammar rules. In this case the query entered is a combination of morphological forms.

Step 5: Now, the parser will find the subject, verb and object to reconstruct the query.

Step 6: The user can refine the query further or stop with the query related terms but use a set of related terms to process the query over web.

Step 7: This system suggests that suggestions must focus on categories. Hence, the query expansion is done using the related terms is used.

Step 8: The query is translated into the target language (English) using the ontology mapping and the morphological language interface.

Step 9: The results are re-ranked using the re-ranking algorithm.

Step 10: The re-ranked results are smoothening using the smoothening system.

Step 11: For this, the bi-lingual ontology is used to convert the Telugu word to the English word. Thus, all the previous stages mentioned are repeated again until the user is satisfied with the target language's representation. The outcome of this stage is representation of source query in the target language.

Step 12: Once the user query is disambiguated in all the stages and it is sent to the web for results. The search results are retrieved and shown to the user in a possible manner.

6. IMPLEMENTATION AND RESULTS

This section summarizes the experimental results obtained in the implementation of the system's aspects, like: accuracy and precision made to real users of the system.

6.1 ACCURACY EVALUATION

Using the Mean Absolute Error (MAE) metric [21, 22] the system's accuracy is evaluated that is, the capability of the system to disambiguate the user query and retrieving the results. MAE considers the average absolute deviation between the available and proposed systems [Eq.(1), Mean Absolute Error metric].

$$MAE = \frac{\sum_{i=1}^{n} abs(p_i - r_i)}{n} \quad (1)$$

where, n is the number of queries in the test set, p_i the predicted rating for an item, and r_i the true rating. The World Wide Web was used to develop experiments and comparison with other algorithms. This system is tested with 500 simple and complex words, phrases and sentences. Each user has tested at least 5 queries and a maximum of 10 queries. The obtained results are shown in Table.2. The performance of GRBA is quite uniform across the test cases, but considering the grammar rules for complex queries, a better average MAE is obtained. Other queries (with one or two words) exhibit results close to the best results. When the number of terms in a query is 4, there is a significant increase in the average MAE. Table.2 also shows the percentage of accuracy achieved in each case. In this case the coverage is high (78.44%) but its value decrease to (68.69%) when the number of terms in user query is greater than 6. Best value of n (terms in query) for the proposed algorithm is around 5; with this value a balance between MAE and coverage is reached.

6.2 PRECESSION AND RECALL EVALUATION

The overall system has implemented in Java and the Ontology and Grammar Rules has built for Telugu language. The well-known search interface was used. The comparison has done between Google results and that of our system. The most widely used measures to calculate retrieval effectiveness are Recall and Precision. Recall measures how well a system retrieves all the relevant results for a given user query; and Precision, how well the system retrieves only the relevant results along with this the system has tested for accuracy. The results of short queries (using the one or two keywords) and long queries (using simple and complex sentences) are presented in each run (user tested system with a minimum of 10 queries in each case). Table.3 and Table.4 list the average precision and average recall of each run and the relative improvement our method achieves over others. Totally 50 runs are taken in the implementation process with 50 different users. It is observed that our rule based algorithm performs better and the average improvement over other methods is about 73% from Table.4.

The rule based approach performs unexpectedly poor in the cases of:

1. Slang words with other languages.
2. Special character based phrases. (Cases like Exclamation, Question types and Negative phrases)
3. Linguistic confusion in long phrases

How to incorporate other language phrases into the system will be a focus of the future work. The usages of the special characters are considered an outlier. The tradeoff on focusing this and complexity of the system increase is to be tested in future. The linguistic confusion is a case of double negatives. The grammar needs to be improved to handle this.

Table.2. Mean average error value for GRBA

n - value	MAE value in each test										Average MAE
	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Q-7	Q-8	Q-9	Q-10	
1	1.054	1.052	1.053	1.052	1.048	1.046	1.052	1.043	1.04	1.047	1.049
2	1.025	1.041	1.02	1.017	1.021	1.023	1.034	1.011	1.018	1.021	1.023
3	0.856	1.032	0.754	1.052	1.048	1.046	1.052	1.043	1.04	1.047	0.997
4	1.026	0.879	1.321	1.017	1.021	0.921	0.768	1.011	0.985	0.765	0.971
5	1.021	0.766	0.786	0.978	0.897	0.933	0.768	0.954	0.867	0.988	0.896
6	0.865	0.987	0.899	0.679	0.976	0.879	0.921	0.763	0.654	0.893	0.852
7	0.985	0.789	0.698	0.945	0.834	0.786	0.675	0.897	0.976	0.887	0.847
8	0.954	0.876	0.789	0.854	0.657	0.745	0.832	0.783	0.854	0.854	0.820
9	0.876	0.765	0.874	0.756	0.856	0.943	0.743	0.549	0.678	0.765	0.781
10	0.834	0.782	0.673	0.675	0.562	0.675	0.879	0.765	0.843	0.876	0.756

(where *n* is the user value and Q denotes the query)

Table.3. Average precision and average recall of simple short queries

	Simple Short Queries				% of Impr.
	Precision		Recall		
	Existing	Rule based	Existing	Rule based	
Run 1	0.214	0.279	0.203	0.238	17.60%
Run 2	0.175	0.196	0.162	0.173	24.28%
Run 3	0.193	0.225	0.173	0.224	32.04%
Run 4	0.145	0.173	0.132	0.182	24.90%

Table.4. Average precision and average recall of complex queries

	Complex Queries				% of Impr.
	Precision		Recall		
	Existing	Rule based	Existing	Rule based	
Run 1	0.113	0.147	0.104	0.172	43.17%
Run 2	0.192	0.209	0.172	0.256	19.84%
Run 3	0.186	0.295	0.154	0.256	30.21%
Run 4	0.116	0.157	0.104	0.143	40.46%

7. CONCLUSION

In this research work, a progressive algorithm called grammar rule based model is introduced. Its main idea is to disambiguate the user given query using linguistic grammar based rules of the native language. Morphological analyzer and generator have been used with the limited resource of linguistic knowledge. There is a marked improvement in precision and recall observed in this system. However, there is some scope for further improvement and research. A mechanism for mapping the contents back into the local language will be done in future. The expansion of the grammar rules for complex queries is another focus of the work. The integration of the systems with information retrieval algorithms is also on the anvil as the present focus is to estimate and quantify the impact of the rule based system alone. For a truly complete system, the grammar rules must act as the preprocessing step and be supplemented by the information retrieval system in the post processing stage. Hence, there is considerable scope for improvement.

REFERENCES

[1] Debasis Mandal, Mayank Gupta, Sandipan Dandapat, Pratyush Banerjee and Sudeshna Sarkar, "Bengali and Hindi to English CLIR Evaluation", *Advances in Multilingual and Multimodal Information Retrieval*, Vol. 5152, pp. 95-102, 2008.

[2] S.Saraswathi, M. Asma Siddhiqaa, K. Kalaimagal and M. Kalaiyarasi, "BiLingual Information Retrieval System for English and Tamil", *Journal of Computing*, Vol. 2, No. 4, pp. 85-89, 2010.

[3] Daqing He and Dan Wu, "Translation enhancement: A new relevance feedback method for cross-language information retrieval", *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 729-738, 2008.

[4] Wessel Kraaij, Jian-Yun Nie and Michel Simard, "Embedding web-based statistical translation models in cross-language information retrieval", *Journal of Computational Linguistics*, Vol. 29, No. 3, pp. 381-419, 2003.

[5] Mallamma V Reddy and M. Hanumanthappa, "Kannada and Telugu Native Languages to English Cross Language Information Retrieval", *International Journal of Computer Science and Information Technologies*, Vol. 2, No. 5, pp. 1876-1880, 2011.

[6] R. Shriram, V. Sugumaran and E. Kapetanios, "Cross-Lingual Information Retrieval and Delivery Using Community Mobile Networks", *1st International Conference on Digital Information Management*, pp. 320 – 325, 2006.

[7] Dinesh Mavaluru, R. Shriram and W. Aisha Banu, "Ensemble Approach for Cross Language Information Retrieval", *Computational Linguistics and Intelligent Text Processing*, Vol. 7182, pp. 274-285, 2012.

[8] Manoj Kumar Chinnakotla, Sagar Ranadive, Om P. Damani and Pushpak Bhattacharyya, "Hindi and Marathi to English Cross Language Information Retrieval", *Advances in Multilingual and Multimodal Information Retrieval*, Vol. 5152, pp 111-118, 2008.

[9] Dinesh Mavaluru and R. Shriram, "Telugu English Cross Language Information Retrieval: A Case Study", *International Journal Of Research In Advance Technology In Engineering*, Vol. 1, Special Issue, pp. 78-83, 2013.

[10] K. Saravanan, R. Udupa and A. Kumaran, "Cross lingual Information Retrieval System Enhanced with Transliteration Generation and Mining", *Proceedings of Forum for Information Retrieval Evaluation*, 2010.

[11] Ashish Francis Almeida and Pushpak Bhattacharyya, "Using Morphology to Improve Marathi Monolingual Information Retrieval", Indian Institute of Technology, Bombay. India. Source http://www.isical.ac.in/ ~fire/paper/Ashish_almeida-IITB-fire2008.pdf, 2008.

[12] Yuanhua Lv and ChengXiang Zhai, "Positional language models for information retrieval", *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 299–306, 2009.

[13] A. Menon, S. Saravanan, R. Loganathan and K. Soman "Amrita Morph Analyzer and Generator for Tamil: A Rule-Based Approach", *Tamil Internet Conference*, pp. 239-243, 2009.

[14] Kumar Sourabh and Vibhakar Mansotra, "An Experimental Analysis on the Influence of English on Hindi Language Information Retrieval", *International Journal of Computer Applications*, Vol. 41, No. 11, pp. 30-35, 2011.

[15] Marcello Federico and Nicola Bertoldi, "Statistical cross-language information retrieval using N-best query translations." *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 167–174, 2002.

[16] Prasad Pingali and Vasudeva Varma, "Multilingual Indexing Support for CLIR using Language Modeling", *IEEE Data Engineering Bulletin*, 2007.

[17] M. Anand Kumar, R.U. Rekha, K.P Soman, S Rajendran and V.V. Dhanalakshmi, "A Novel Data Driven Algorithm for Tamil Morphological Generator", *International Journal of Computer Applications*, Vol. 6, No. 12, pp. 52–56, 2010.

[18] Madhavi Ganapathiraju and Lori Levin, "TelMore: Morphological Generator for Telugu Nouns and Verbs", *Proceedings of Second International Conference on Universal Digital Library*, pp. 17-19, 2006.

[19] Ganapathiraju Madhavi, Balakrishnan Mini, N. Balakrishnan and Reddy Raj, "Om: One tool for many (Indian) languages", *Journal of Zhejiang University Science A*, Vol. 6, No. 11, pp. 1348-1353, 2005.

[20] R. Sri Badri Narayanan, S. Saravanan and K. Soman, "Data Driven Suffix List And Concatenation Algorithm For Telugu Morphological Generator", *International Journal Of Engineering Science and Technology*, Vol. 3, No. 8, 2011.

[21] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen and John T. Riedl, "Evaluating collaborative filtering recommender systems", *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 5-53, 2004.

[22] B. Manikandan and R. Shriram, "A novel approach for cross language information retrieval", *3rd International Conference on Electronics Computer Technology*, Vol. 6, pp. 34-38, 2011.