AgroExplorer: a Meaning Based Multilingual Search Engine

Mrugank Surve, Sarvjeet Singh, Satish Kagathara, Venkatasivaramasastry K, Sunil Dubey, Gajanan Rane, Jaya Saraswati, Salil Badodekar, Akshay Iyer, Ashish Almeida, Roopali Nikam, Carolina Gallardo Perez[1], Pushpak Bhattacharyya[2], AgroExplorer Group

Media Lab Asia,
IIT Bombay.

{mrugank,dubey,pb}@cse.iitb.ac.in

Key Words: UNL Graph, Encoding, Decoding, Focused Crawler, Document Retrieval

Abstract: In this paper we describe **Agro Explorer,** a *language independent search engine with multilingual information access facility*. Instead of searching on plain text it does the search on the *meaning representation*, an Interlingua form called *Universal Networking Language (UNL)* expressions. Most of the current search engines(e.g. Google, Altavista, Yahoo) are pattern based. They do not consider the meaning of the query posed to them. The search is purely based on the keywords of the query. In contrast to this, our system first extracts the meaning of the query and then performs the search based on this extracted meaning. Our system also employs Interlingua based machine translation technology to present information in the language of choice of the user.

1. Introduction

Internet has revolutionized our lives. However, most of the information on the internet being in English causes the internet to be effectively unavailable to the rural masses unqualified in English. The benefits of IT have not been derived by a large section of Indian population, mainly in rural areas. The reasons are given as lack of infrastructure, inadequate dissemination of information and so on. However, the problem of *language barrier* should be cited as one of the primary reasons.

Most of the advanced information for the agricultural domain should be in local languages. This should be available on the web for the farmers to read, assimilate and use. There is also the need for cross-language information transfer where climatic and agricultural conditions are similar (like Bengal, Bihar, Assam, and Orissa), thereby avoiding duplication of research and information hunting effort. The need for multilingual information processing is enormous for a country like India.

In this paper we describe **Agro Explorer,** a *language independent search engine with multilingual information access facility*. Instead of searching on plain text it does the search on the *meaning representation*, an Interlingua form called *Universal Networking Language (UNL)* expressions [1]. Most of the current search engines(e.g. Google[2], Altavista[3], Yahoo [4]) are pattern based. They do not consider the meaning of the query posed to them. The search is purely based on the keywords of the query. In contrast to this, our system first

[1] Universitat de Polytechnica, Madrid
[2] Author for contact

extracts the meaning of the query and then performs the search based on this extracted meaning. Our system also employs Interlingua based machine translation technology to present information in the language of choice of the user.

In section 2 we give a brief introduction of the UNL system. Section 3 describes the architecture of our system along with a brief description of the individual components of the system. Section 4 takes a look at Relevance and Ranking techniques employed by our system. In section 5, we show some results in the form of screenshots of the system run on some representative queries.

2. The Universal Networking Language (UNL)

The Universal Networking Language (UNL) is an electronic language for computers to express and exchange information. UNL system consists of *Universal words (UW)* (explained below), *relations, attributes*, and the *UNL knowledge base (KB)*. The UWs constitute the vocabulary of the UNL, relations and attributes constitute the syntax and the UNL KB constitutes the semantics. The KB defines possible relationships between UWs.

UNL represents information sentence-by-sentence as a hyper-graph with concepts as nodes and relations as arcs. The representation of the sentence is a hyper-graph because a node in the structure can itself be a graph, in which case the node is called a *compound word (CW)*. Figure 1 represents the sentence *John eats rice with a spoon*.

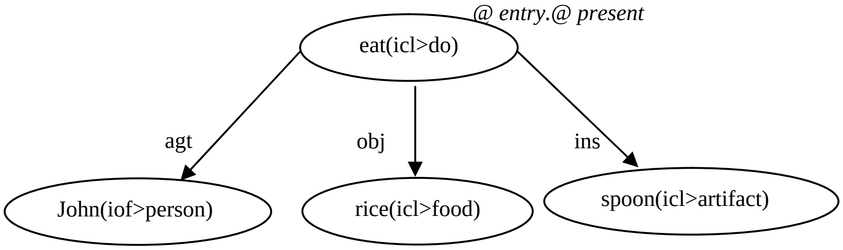


Figure 1. UNL graph of "John eats rice with a spoon"

In this figure, the arcs labeled with *agt* (agent), *obj* (object) and *ins* (instrument) are the relation labels. The nodes *eat (icl>do); John (iof >person), rice (icl>food)* and *spoon (icl>artifact)* are the *Universal Words* (*UW*). These are language words with *restrictions* in parentheses. *icl* stands for *inclusion* and *iof* stands for *instance of*. UWs can be annotated with attributes like *number, tense etc.*, which provide further information about how the concept is being used in the specific sentence. Any of the three restriction labels- *icl, iof* and *equ*- can be attached to an UW for restricting its sense.

2.1 Analysis system

Analysis process converts the source language sentences into the UNL expression. This process requires a parser called Enconverter[2].

Enconverter is a language independent parser, which provides a framework for morphological, syntactic and semantic analysis synchronously. It analyzes sentences using *word dictionary* and *analysis rules.* Given an input sentence, it starts from the leftmost end, and analyzes the sentence left to right applying *morphology rules* and *syntax rules*. When an input string is scanned, all matched morphemes are retrieved from the dictionary and they become the candidate morphemes. Rules are applied from the rule base on these candidate morphemes, and gradually from the input sentence the concepts corresponding to the words, with all available information associated with them, are extracted and structured, and taken away from the word list into the node net. This process goes on until the entire set of words and the available information in the input sentence has been exhausted.

2.2 Generation system

Generation process is the part of UNL System in charge of generating an expression in the natural language from the information described in the UNL expression. This process requires a generator called **Deconverter**[3].

DeConverter is a language independent generator that provides synchronously a framework for morphological and syntactic generation, and word selection for natural collocation. DeConverter can deconvert UNL expressions into a variety of native languages, using a different set of files such as the Word Dictionary, Grammatical Rules and Co-occurrence Dictionary of each language.

Outline of the functions in DeConverter: First of all, DeConverter transforms the sentence represented by an UNL expression - that is, a set of binary relations - into the directed hyper graph structure called **Node-net**. The root node of a Node-net is called Entry Node and represents the main predicate of the sentence. It then applies generation rules to every node in the Node-net respectively, and generates the word list in the target language. In this process, the syntactic structure is determined by applying Syntactic Rules, while morphemes are generated by applying Morphological Rules.

The Generation capability of this system covers context-free as well as context-sensitive language. Since its capability is high enough, it is expected to be able to generate many languages of the world. Co-occurrence Relations between words contribute to better word selection, thus it is possible hereby to generate more natural sentences.

3. System Architecture

Figure 2. shows the overall architecture of the system with all the modules. As shown, the Focused Crawler crawls the web and collects pages related to the Agricultural domain and creates an HTML corpus. This corpus is then passed to an HTML parser which separates the text and design part of the pages. The design part of the HTML pages is saved for later use. The raw text in the form of sentences is then passed to the Enconverter which converts it into UNL form. The UNL corpus thus created is then preprocessed and passed to the Indexer module which creates an inverted index on the UNL expressions. This is the offline process which takes place in the background.

Once a query is entered by the user, we first get the UNL of the query by passing it through the Enconverter. After preprocessing, this UNL expression is passed to the Search Module which uses the inverted index created earlier and performs a graph-based search on the UNL expression of the query.

The search module returns documents which are in UNL format. Then depending on the language selected by the user, the UNL documents are passed to the corresponding Deconverter, which converts the document into the target language. This document is then merged with the HTML design templates which were saved earlier. Thus the translated page is shown to the user in the same format as it was present on the web.

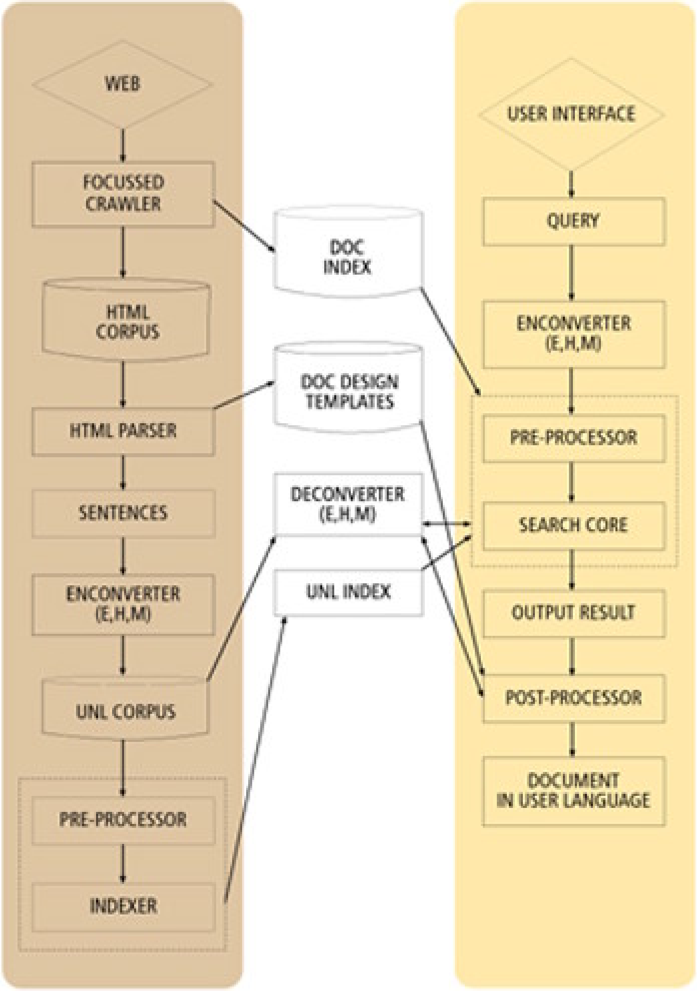


Figure 2. Block Diagram of the System

3.1 Focused Crawler

Generic crawlers crawl the hypertext graph of the web and fetch pages which could then be indexed by the indexer. But even with high-end multiprocessors and exquisitely crafted crawling software, the largest crawls cover only 30-40% of the entire web. The fraction of agriculture related pages fetched by such crawlers would be even less than this. Also the cost of maintaining and updating the index for such a giant crawl is prohibitive.

A Focused Crawler[4], on the other hand, crawls the web and fetches documents pertaining to a specific topic only, which in our case is Agriculture. It entails a very small investment in hardware and network resources and yet achieves a respectable coverage.

We begin by creating a taxonomy using The Open Directory Project and some sample pages from each category. These are then presented to the classifier which builds a model based on this taxonomy. We also collect pages of interest to us i.e. pages from the agriculture domain. These pages are used as the seed set of pages for the crawler. Our aim is to crawl as many relevant pages possible while avoiding the non-relevant pages. To achieve this, we note that pages that are relevant to the domain of crawling have greater probability of having outlinks to pages that are also relevant. So there is a high probability that outlinks from pages belonging to the agriculture domain will also point to agriculture related pages. The outlinks from the seed set of pages are extracted and maintained in a queue according to their priority. This priority is based on the score assigned by the classifier to the page from which the links were extracted.

At this point we can start the focused crawler. The crawler picks up the link of highest priority and fetches the page. Then it presents this page to the classifier which assigns it a relevance score. If this score is below a threshold the page is not included in the Crawl. Otherwise, the page is included and all the outlinks from the page are extracted and added to the link queue. In this fashion, the crawl continues until all the links are exhausted or the crawler is stopped.

3.2 HTML Parser

The HTML corpus created by the focused crawler is not directly processable by the Enconverter. We need to extract the text from the HTML page which would be subsequently given to the Enconverter for conversion to UNL. Also, The design of the page needs to be preserved so that when the translated page is shown to the user, it is in the same format and style as it was originally present on the web.

In order to achieve this, a customized HTML parser was developed. The HTML Parser stores the design of the page in the document design template, which consists of only HTML tags with the placeholders for sentences. The sentences are stored in a separate file which would be used as input to the Enconverter.

3.3 Enconverter

The Enconverter converts the text extracted from the HTML corpus into its equivalent UNL representation. It uses a lexicon and a rule-base for the enconversion. It thus creates the UNL Corpus which would be used for indexing the documents.

3.4 Preprocessor

The UNL expressions contain some extraneous information which is not needed by the search engine. Also there are certain peculiarities associated with indexing Compound UWs present in a UNL relation. The preprocessor handles these tasks and converts the UNL Corpus into an intermediate format used by the indexer.

3.5 Indexer

The indexer is entrusted with the task of taking the UNL Corpus and generating an inverted index on it. It parses the UNL expressions and extracts the relation, the two UW’s on which the relation has been formed(UW1 and UW2) and their respective UW-IDs. The UNL index, which is stored as a table in a mySQL database, is then updated with these values. This completes the offline process and the documents indexed are now searchable through the Search Module.

3.6 Search Module

This is the nucleus of the search engine. The query entered by the user is converted to UNL, preprocessed by the same Preprocessor that we used for indexing, and then passed on to the Search Module. The Search Module uses the UNL index created by the indexer to perform a graph-based search on the UNL query. It also ranks the results as per their relevance.

Searching for the documents is a simple task but ranking the documents as per their relevance with respect to the query is an altogether different and complex task. In the next section we explain in detail how ranking of the documents is performed in our search Engine.

3.7 Deconverter

The search module returns the UNL documents that match with the query. Now the task is to convert these UNL documents into the language of the user’s choice. This task is performed by the Deconverter. Like the Enconverter, it also uses a lexicon and a rule-base.

3.8 Post-Processor

Finally, the Post-Processor takes the translated text and the design of the page, extracted and stored earlier by the HTML parser, and merges them to form an HTML page which is presented to the user. Thus the original page along with its design is shown to the user in his native language.

4. Relevance and Ranking

A web search engine not only returns a set of pages in response to the user's query, but it also has the job of assigning relevance scores to each of them. The relevance of a web page is the measure of the web page's importance with respect to a search query. The importance of a web page is inherently a subjective matter, which depends on the reader's interests, knowledge and attitudes. In spite of this there is much that can be said objectively about the relative importance of web pages that are retrieved in response to a search query.

This kind of ranking of web pages is important in many respects. It saves time and efforts of the search engine user, because the user will find the pages that are most likely to be relevant on the top of the search results. The definition and implementation of the relevance directly affects implementation details of many other aspects of the search engine such as indexing and data structures used for representing the processed corpus. Finally, it affects the Precision and Recall of the search engine, the two most important measures used for judging the results of a search engine.

The relevance of a page to a given query is function of the Global page rank and query-specific page rank.

4.1 Global Page Rank

The global page rank measures the relative importance of the web pages. It does not take into account the user's query to calculate the rank of the page; instead the hyperlink structure of the web is used to calculate the page rank. Even without the knowledge of the actual contents of a web page, we can predict a lot about the relative importance of web pages by looking at the overall link structure of the web. Many novel algorithms for assigning an importance measure to web pages based on link structure analysis have been proposed(e.g. HITS[5], PageRank[6]). Generally, highly linked pages are more important than pages with few links. Alternately, web pages which have few links to them but from prestigious pages such as Yahoo are more important than pages which have large number of links from obscure or not so important pages. Thus an outlink from the Yahoo homepage carries more weight than an outlink from a student’s homepage. The current system doesn’t use the Global page rank, but as the corpus size increases in the future, we can easily incorporate the Global page rank with our own page ranking algorithm.

4.2 Query Specific Page Rank

The Global page rank is independent of the search query. Obviously, the relevance of the web page will also depend on the query posed by the user. For example, in a keyword based search engine, if the query X Y is given, where X and Y are keywords, a page containing both the terms X and Y will be more relevant than a page having only X or only Y.

As explained earlier, in our search engine, both the query and the document is converted into UNL before a meaning based search is performed. For each sentence in the document, we will have one UNL graph. Thus, essentially, we will have a collection of UNL graphs (document) and a given UNL graph (query) which needs to match with the document. If the query graph is a subgraph of any sentence graph in the document, then we can say that the sentence is relevant for the query and the document should be retrieved by the search engine. Intuitively, we need to do a subgraph matching between the query graph and the document sentence graphs.

4.3 Complete Matching

From the above discussion, the first and the easiest algorithm for finding the relevant documents will be as follows. After the query graph is found, a subgraph checking is done on every sentence graph in the document. If the query graph is a subgraph of a sentence, that sentence is considered relevant to the query. A document having more proportion of relevant sentences will be more relevant to a query. Mathematically, this can be expressed as:

$$R_q(d) = \frac{S_{s \in S}\, r_q(s)}{|S|}$$

Where $R_q(d)$ is relevance of document d to the query q. S is set of sentences in the document d. $r_q(s)$ is the relevance of sentence s to the query q. As mentioned above $r_q(s) = 1$ if the query is a subgraph of the sentence graph, 0 otherwise.

To find out the results for a query q, we need to look at each and every sentence of all the documents! Clearly this is very time consuming and impractical as number of documents on the Internet is huge. Thus we need indexing to reduce the time consumed in finding the results for a given query.

The most obvious and relevant indexing scheme will be to index the whole corpus of UNL documents on the edges of the UNL graph. An edge of the UNL graph is a triplet

$$r(U_1, U_2)$$

where r is the relation-label and U_1 and U_2 are Universal Words.

For each edge present in the corpus, we will store the (document id, sentence number) pairs in the index. Now it will be easy to find what all (d,s) pairs have all the edges of the query by taking intersection of sets of (d,s) for each edge of the query.

This approach has a serious problem associated with it. It may sometimes return a document as relevant to a query, even when none of the sentences in the document has the query as its subgraph. To understand this more clearly, consider a UNL graph G. The vertices of this graph will be the UWs and the edges denote the relations between them.

Let vertex $v_1 \in V(G)$. Consider a query, $Q = G \cup r_1(v_1, v_2)$ as shown in figure 3.1. Now consider a document having a sentence graph H(figure 3.2), with $G \subset H$ and having an edge $r_2(v_1 , v_3)$ instead of $r_1(v_1 , v_2)$ as is present in the query. Also, H has an edge $r_1(v_1 , v_2)$ inside it which is not directly linked with G. Here the UW v_1 present in r_1 and that present in r_2 refer to two different instances of the same concept. This is represented in the UNL expression by having two different UW-ID's for each one of them. Clearly, if we apply the above algorithm, this document will also be returned even when it does not contain the query subgraph.

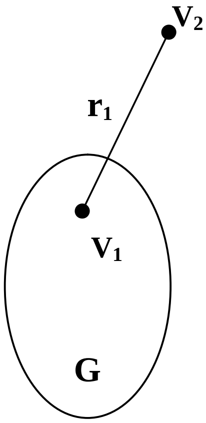


Figure 3.1 Query Graph

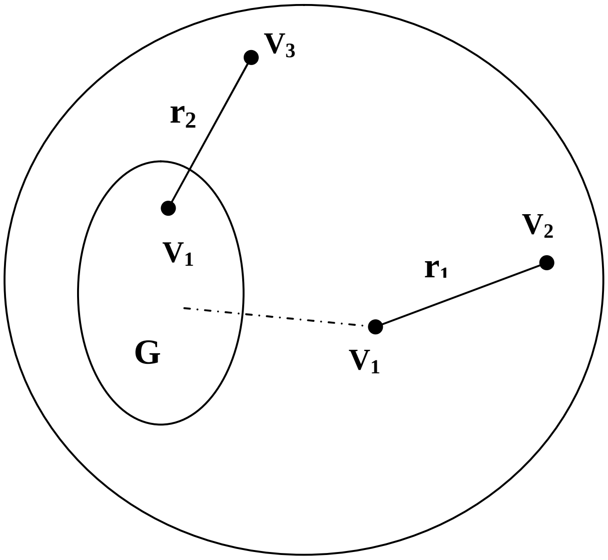


Figure 3.2 Sentence Graph

To overcome this problem, some processing is done after finding the relevant sentences to make sure that the edges found inside the sentence have the same connections as they have in the query. Whether two UWs (vertices) are two different instances of the same concept is indicated by their UW-IDs. In order to positively establish the connectivity of two edges $r_1(u_1 , u_2)$ and $r_2(u_2 , u_3)$, we only need to make sure that UW-ID of u_2 in first relation is same as that of u_2 in the second relation.

4.4 Partial Matching

Another drawback of the complete matching approach is that it is a one-or-none matching approach i.e. a sentence will be either relevant to a query or it won't be. There is no concept of *partial* or *approximate* matching. Undoubtedly, complete matching approach will lead to high precision but low recall. To overcome this problem, we can introduce a partial matching scheme which has lower precision but higher recall.

Two different occurrences of the same UW in two different edges are said to be linked if the UW-IDs of these two occurrences are same. In the document graph, we say that a link between two occurrences of the same UWs, is a *correct* link if there is link corresponding to this link, between the UWs in the query graph also. A correct link is defined only for the common edges in the document and the query.

As we did in complete matching, we will index the documents on their edges. The relevance of a document $R_q(d)$ is same as given in the complete matching scheme. But for partial matching the $r_q(s)$ is now defined as:

$$r_q(s) = \alpha \frac{n}{N} + (1-\alpha)\frac{l}{L}$$

where, *n* is number of relation edges (of the query) found in the sentence. N is the total number of relation edges in the query. *l* is the number of *correct* links in the sentence and L is the total number of links between all UWs in the query. α is a empirical constant.

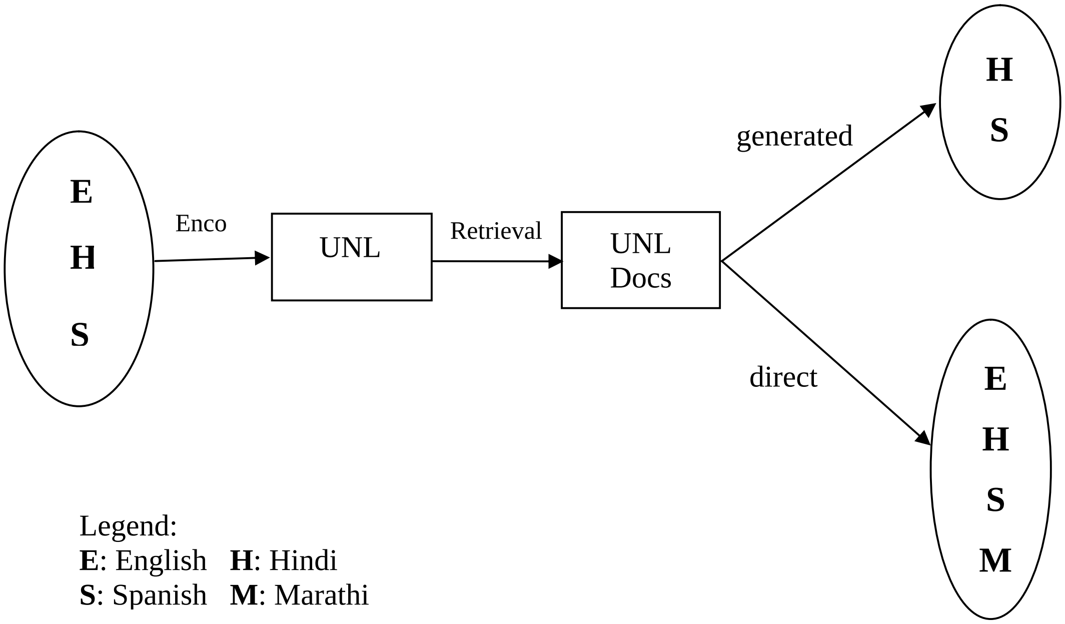
In both of these approaches we have assumed that all the sentences have equal importance. But keyword based search engines, give more weight to terms if they are present in the title or have relatively bigger font size than the rest of the document. We can do the same by giving more weight to those $r_q(s)$ which correspond to sentences having bigger font size or title sentence(s).

5. Results:

Here we shall show the screenshots of the system for some representative queries to demonstrate how the system performs meaning based search as well as to show the multilingual aspect of the system. Currently the system has 7 documents belonging to the agriculture domain picked up from the web. They have been converted into UNL through semi-automatic enconversion assisted by humans. These seven documents have been indexed into the system. Currently the system is capable of accepting queries in English, Hindi and Spanish. Work is under way for including Marathi as well. The entire system has been made UNICODE-Compliant[7]. These UNL documents can also be deconverted online into Hindi or Spanish. Figure 4. shows the current capability of the system.

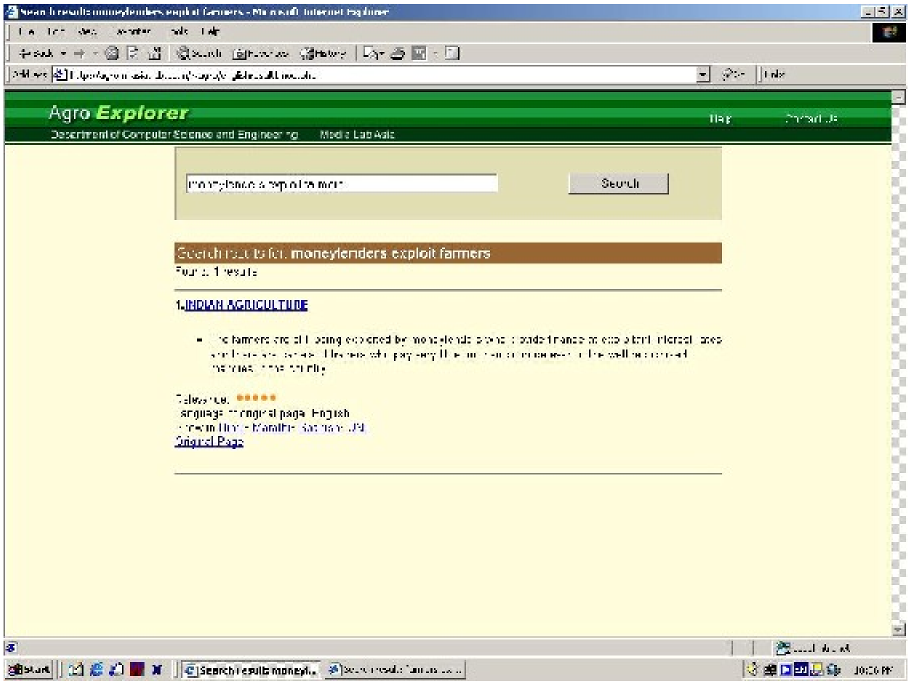
Figures 5.1 and 5.2 illustrate the Meaning-Based Search. As can be seen from the results, the system has the capability of distinguishing between two queries having the same keywords but entirely different meanings. For “moneylenders exploit farmers” the system returns a document but for “farmers exploit moneylenders” the system cannot find a document which matches the query. Given these two queries to Keyword-Based search engines, they would return almost the same results.

Figure 5.3 shows the Multilingual aspect of the system. Here the input query to the system is given in Hindi. The document matched is in English and the user now has the option of viewing the document in Hindi. Similarly Figure 5.4 shows the output of the system for a Spanish query.



Legend:
E: English **H**: Hindi
S: Spanish **M**: Marathi

Figure 4. Current Capability of the System.



moneylenders exploit farmers

Search results for: **moneylenders exploit farmers**

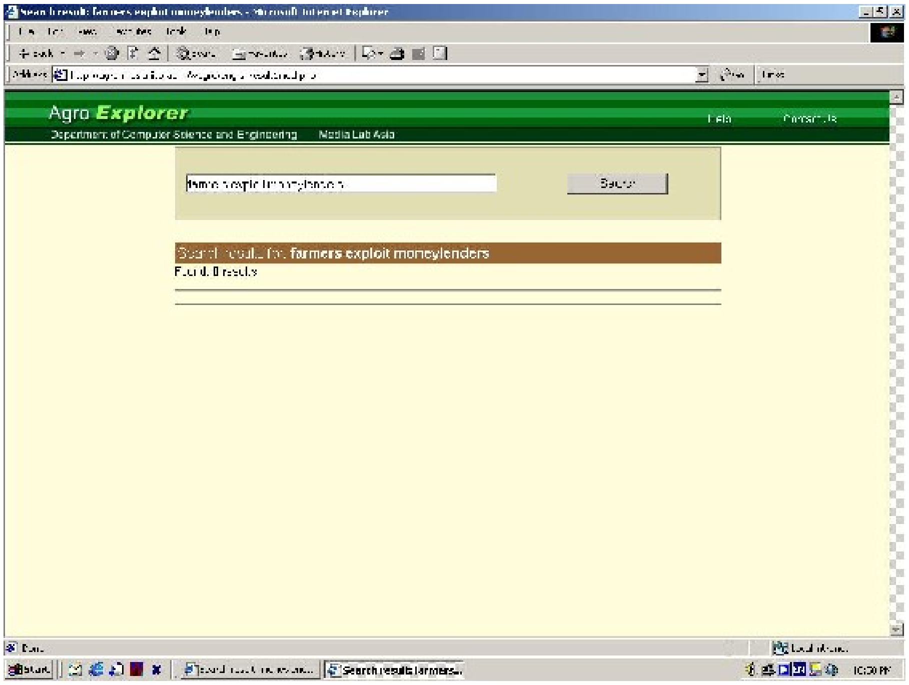
Found 1 results

1. INDIAN AGRICULTURE

- The farmers are still being exploited by moneylenders [illegible] interest rates [illegible]

Relevance: ●●●●●
Language of original page: English
View in Hindi Marathi Spanish UNL
Original Page

Figure 5.1 search results for "moneylenders exploit farmers"

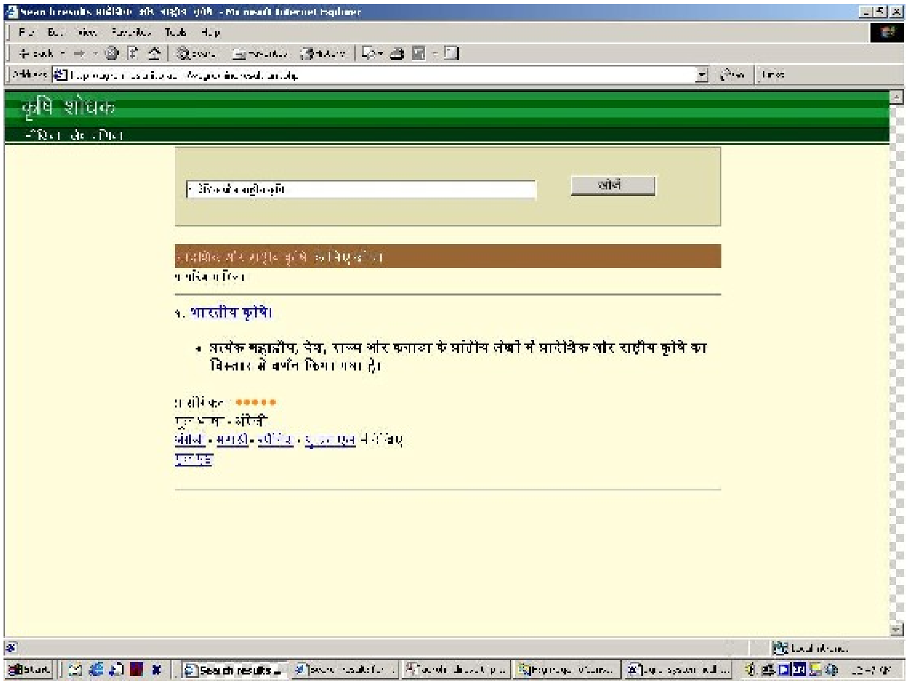


farmers exploit moneylenders

Search results for: **farmers exploit moneylenders**

Found 0 results

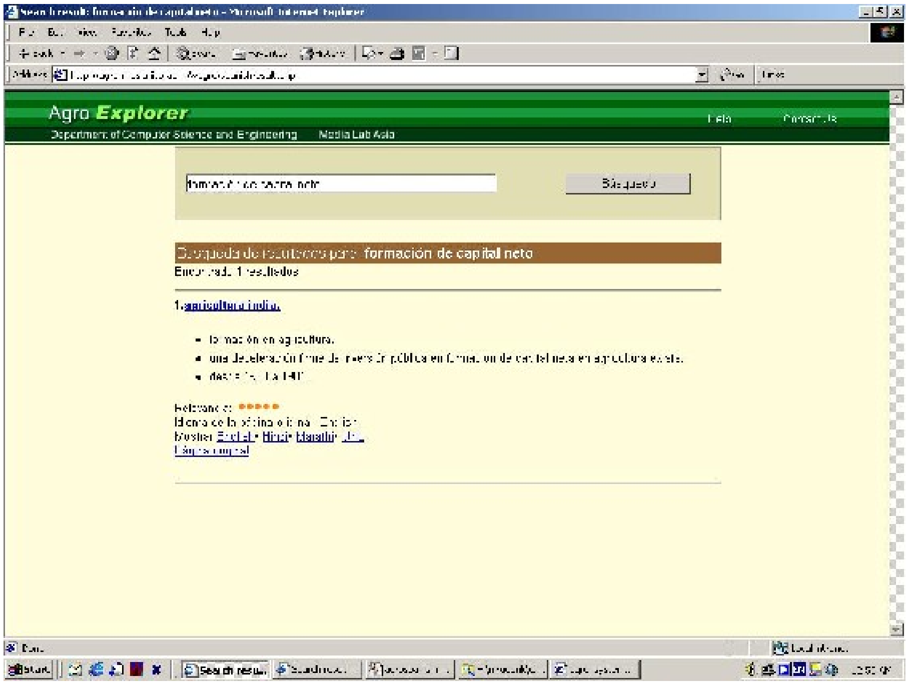
Figure 5.2 search results for "farmers exploit moneylenders"



१. भारतीय कृषि।

- प्रत्येक महाद्वीप, देश, राज्य और कनाडा के प्रांतीय लेखों में प्रादेशिक और राष्ट्रीय कृषि का विस्तार में वर्णन किया गया है।

Figure 5.3 search results for Hindi Query



Búsqueda de resultados para formación de capital neto
Encontrado 1 resultados

1. agricultura india.

- formación en agricultura.
- una deceleración firme de inversión pública en formación de capital neto en agricultura existe.
- desde [illegible] 1980

Relevancia: ●●●●●
Idioma de la página original: English
Mostrar English Hindi Marathi UNL
Página original

Figure 5.4 search results for Spanish query

Conclusion

In this paper, a different approach to the problem of meaning based search engine and multilinguality is presented. We believe that meaning based search on the web is becoming the need of the hour. For the search results to be realistic and meaningful, they must encompass the typical user's requirements and specifications.

The model in this paper is an amalgamation of two independent features. We integrated the user's language requirement with the relative importance of knowledge the user seeks. This has been possible by using the UNL as an intermediary language. Because of UNL both multi-lingual and meaning based properties can be incorporated together rather than using separate language translators in search engines.

The scheme admits itself to Integration of multiple languages in a seamless, scalable manner. The UNL encoding of the query, the graph-based Search, indexing on expressions and Concepts rather than just keywords are the main contributions of this work.

Future work involves more efficient encoding of a large document base in the UNL form. One possibility is to use parsing and chunking tools and lexical resources for preprocessing of the documents prior to encoding in the form of UNL graphs. The Focused Crawler also needs to be augmented with enhanced learning capability.

References

[1] **The Universal Networking Language (UNL) Specifications**, Version 3.0, UNL center, UNDL Foundation, 2001.
http://www.unl.ias.edu/unlsys/unl/UNL%205specifications.html.

[2] **Enconverter Specifications**, Version 3.1, UNL center, UNDL Foundation, 2001.

[3] **Deconverter Specification**, Version 2.5, UNL center, UNDL Foundation, 2001.

[4] S. Chakrabarti, M. Van den Berg, B. Dom.
Focused Crawling: a new approach to topic-specific Web resource discovery. In *Proceedings of the Eighth International World Wide Web Conference*(WWW8), 1999.

[5] Jon M. Kleinberg.
Authoritative Sources in a Hyperlinked Environment. In *Journal of the ACM* 1999.

[6] Sergey Brin and Lawrence Page.
The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the the Seventh International World Wide Web Conference*(WWW7), 1998.

[7] Cathy Wissink.
Indic Script Support on the Windows Platform. In *Proceedings of the 23rd Internationalization and Unicode Conference*, 2003

[8] http://www.unicode.org

[9] http://www.google.com

[10] http://www.altavista.com

[11] http://www.yahoo.com