G.PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY,KURNOOL
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IV B.TECH I SEM (R13)

INFORMATION RETRIEVAL SYSTEMS

UNIT-3

UNIT-III
Semantic Networks

Semantic networks are based on the idea that knowledge can be represented by concepts which are linked together by various relationships. A semantic network is simply a set of nodes and arcs. The arcs are labelled for the type of relationship they represent. Factual information about a given node, such as its individual characteristic (color, size, etc.), are often stored in a data structure called a frame. The individual entries in a frame are called slots A frame for a rose can take the form:
(rose

(has-color red)

(height 2 feet)

(is-a flower)

)

Here the frame rose is a single node in a semantic network containing an is-a link to the node flower. The slots has-color and height store individual proper-ties of the rose.Natural language understanding systems have been developed to read hu-man text and build semantic networks representing the knowledge stored in the text turns out that there are many concepts that are not easily represented (the most difficult ones are usually those that involve temporal or spatial reasoning). Storing information in the sentence, "A rose is a flower.", is easy to do as well as to store, "A rose is red", but semantic nets have difficulty with storing this information: "The rose grew three feet last Wednesday and was taller than anything else in the garden." Storing in-formation about the size of the rose on different dates, as well as, the relative location of the rose is often quite difficult in a semantic network. For a detailed discussion see the section on "Representational Thoms" about the large-scale knowledge representation project called eye .

Despite some of the problems with storing complex knowledge in a semantic network, research was done in which semantic networks were used to improve information retrieval. This work yielded limited results and is highly language specific, however, the potential for improvement still exists. Semantic networks attempt to resolve the mismatch problem in which the terms in a query do not match those found in a document, even though the document is relevant to the query. Instead of matching characters in the query terms with characters in the documents, the semantic distance between the terms is measured (by various measures) and incorporated into a semantic network. The premise behind this is that terms which share the same meaning appear relatively close together in a semantic network. Spreading activation is one means of identifying the distance between two terms in a semantic network There is a close relationship between a thesaurus and a semantic network. From the standpoint of an information retrieval system, a thesaurus attempts to solve the same mismatch problem by expanding a user query with related terms and hoping that the related terms will match the document. A semantic network subsumes a thesaurus by incorporating links that indicate "is-a-synonym-of" or "is-related-to," but a semantic

network can represent more complex information such as an is-a hierarchy which is not found in a thesaurus.

One semantic network used as a tool for information retrieval research is WorldNet. WorldNet is publicly available and contains frames specifically designed for words (some semantic networks might contains frames for more detailed concepts such as big-and-hairy-person). WorldNet can be found on the Web at: www.cogsci.princeton.edurwn.WordNet contains different entries for the various semantic meanings of a term. Additionally, various term relationships are stored including: synonyms, antonyms (roughly the opposite of a word), hyponyms (lexical relations such as is-a), and metonyms (is a part-of). Most nouns in WorldNet are placed in the is-a hierarchy while antonyms more commonly relate adjectives. Interestingly, less commonly known relations of entailment and troponyms are used to relate verbs. Two verbs are related by entailment when the first verb entails the second verb. For example, to buy something entails that you will pay for it. Hence, buy and pay are related by entailment. A troponym relation occurs when the two activities related by entailment must occur at the same time (temporally co-extensive) such as the pair (limp, walk). Software used to search WordNet is further described in .It is reasonable to assume that WordNet would help effectiveness by expanding query terms with synsets found in WordNet. Initial work done by Voorhees however, failed to demonstrate an improvement in effectiveness. Even with manual selection of synsets, effectiveness was not improved when queries were expanded. A key obstacle was that terms in queries were not often found in WordNet due to their specificity-terms such as National Rifle Association are not in WordNet. Also, the addition of terms that have multiple meanings or word senses significantly degrade effectiveness. More recent work, with improvements to WordNet over time has incorporated carefully selected phrases and showed a small (roughly five percent) improvement Semantic networks were used to augment Boolean retrieval and automatic relevance ranking. We describe these approaches in the remainder of this section.

3.1 Distance Measures

To compute the distance between a single node in a semantic network and another node, a spreading activation algorithm is used. A pointer starts at each of the two original nodes and links are followed until an intersection occurs between the two points. The shortest path between the two nodes is used to compute the distance. Note that the simple shortest path algorithm does not apply here because there may be several links that exist between the same two nodes. The distance between nodes a and b is: Distance(a,b) = minimum number of edges separating a and b

3.1.1 R-distance

The problem of measuring the distance between two sets of nodes is more complex. Ideally the two sets line up, for example "large rose" and "tall flower" is one such example where "large" can be compared with "tall" and "rose" can be compared with "flower." The problem is that it is difficult to align the concepts such that related concepts will be compared. Hence, the R-distance takes all of the individual entries in each set and averages the distance between all the possible combinations of the two sets. If a document is viewed as a set of terms that are "AND"ed together, and a query is represented as a Boolean expression in disjunctive normal form, then the R-distance identifies a measure of distance between the Boolean query and the document. Also, a

NOT applied to a concept yields the distance that is furthest from the concept. Hence, for a query Q for terms ((a AND b AND c) OR (e AND f)) and Document D with terms (tl AND t2), the similarity is computed below.

$$c_1 = \frac{d(a,t_1) + d(a,t_2) + d(b,t_1) + d(b,t_2) + d(c,t_1) + d(c,t_2)}{6}$$

$$c_2 = \frac{d(e,t_1) + d(e,t_2) + d(f,t_1) + d(f,t_2)}{4}$$

SC(Q,D) is computed now as the MIN(CI, C2). Essentially, each concept represented in the query is compared to the whole document and the similarity measure is computed as the distance between the document and the closest query concept.

Formally, the R-distance of a disjunctive normal form query Q, and a document D with terms (tl' t2, ... ,tn) and Cij, indicates the jf.h term in concept I is defined as:

$$\begin{aligned} SC(Q,D) &= min\left(SC_1(c_1,D), SC_1(c_2,D), \ldots, SC_1(c_m,D)\right) \\ SC_1(c_i,D) &= \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} d(t_i,c_{ij}) \\ SC(Q,D) &= 0, \texttt{ if } Q = D \end{aligned}$$

3.1.2 K·distance

A subsequent distance measure referred to as the K-distance was developed. This measure incorporates weighted edges in the semantic network. The distance defined between two nodes is obtained by finding the shortest path between the two nodes (again by using spreading activation) and then summing the edges along the path. More formally the distance between terms ti and tj is obtained by:

$$d_{ij} = w_{t_i,x_1} + w_{x_1,x_2} + \ldots + w_{x_n,t_j}$$

where the shortest path from ti to tj is: ti, Xl, X2, ... , tj. The authors treat NOT as a special case. The basic idea is to dramatically increase the weights of the arcs that connect the node that is being referenced with a NOT (referred to as separation edges). Once this is done, any paths that include this node are much longer than any other path that includes other terms not referenced by a NOT. To obtain the distance between two sets, A and B, of nodes with weighted arcs, the K-distance measure computes the minimum of the distances between each node in set A and set B. These minimum distances are then averaged. Since the weights on the arcs may not be equivalent in both directions, the distance measure from A to B is averaged with the distance from B to A. For our same query Q: «a AND b AND c) OR (e AND 0)

Assume document D has only two terms: (tl AND t2), the similarity is computed below.

$$c_1 = \frac{min(d(a,t_1),d(a,t_2)) + min(d(b,t_1),d(b,t_2)) + min(d(c,t_1),d(c,t_2))}{3}$$

$$c_2 = \frac{min(d(e,t_1),d(e,t_2)) + min(d(f,t_1),d(f,t_2))}{2}$$

SC(Q,D) is still the min(q, C2). The value of SC(D,Q) would then be obtained, and the two coefficients are then averaged to obtain the final similarity measure. The K-distance of a disjunctive normal form query Q and a document D with terms (tl' t2, ... , tn) is defined as:

$$SC(Q,D) = \frac{SC_1(Q,D) + SC_1(D,Q)}{2}$$

$$SC_1(Q,D) = min\left(SC_2(c_1,D), SC_2(c_2,D), \ldots, SC_2(c_m,D)\right)$$

$$SC_2(c_i,D) = \frac{1}{n}\left(\sum_{j=1}^{n} min\left(d(c_{ij},t_j)\right)\right)$$

$$SC(Q,D) = 0, \ \mathtt{if} \ \ Q = D$$

The R-distance satisfies the triangular inequality such that r-dist(a,c) is less than or equal to r-dist(a,b) + r-dist(b,c). The K-distance does not satisfy this inequality but it does make use of weights along the edges of the semantic network.

3.1.3 Incorporating Distance

Lee, et aI., incorporated a distance measure using a semantic network into the Extended Boolean Retrieval model and called it-KB-EBM for Knowledge Base-Extended Boolean Model . The idea was to take the existing Extended Boolean Retrieval model and modify the weights used to include a distance between two nodes in a semantic network. The Extended Boolean model uses a function F that indicates the weight of a term in a document. In our earlier description we simply called it Wi, but technically it could be represented as F(d, ti). Lee, et al., modified this weight by using a semantic network and then used the rest of the Extended Boolean model without any other changes. This cleanly handled the case of NOT. The primitive distance function, d(ti, tj), returns the length of the shortest path between two nodes. This indicates the conceptual closeness of the two terms. What is needed here is the conceptual distance, which is inversely proportional to the primitive distance function. Hence, the new F function uses:

$$distance^{-1}(t_i,t_j) = \frac{\lambda}{\lambda + distance(t_i,t_j)}$$

First, the function F is given for a document with unweighted terms. The new function, F(d, ti), computes the weight of term ti in the document as the average distance of ti to all other nodes in the document. The new function F is then:

$$F(d,t) = \frac{\sum_{i=1}^{n} distance^{-1}(t_i, t)}{1 + \frac{\lambda}{\lambda+1}(n-1)}$$

For existing weights for a term in a document, F is modified to include weights Wi. This is the weight of the i^th term in document d.

$$F(d,t) = \frac{\sum_{i=1}^{n} distance^{-1}(t_i, t) w_i}{1 + \frac{\lambda}{\lambda+1}(n-1)}$$

3.1.4 Evaluation of Distance Measures

All three distance measures were evaluated on four collections with nine, six, seven, and seven documents, respectively. Precision and recall were not measured, so evaluations were done using comparisons of the rankings produced by each distance. In some cases MESH was used-a medical semantic network-in other cases, the Computing Reviews Classification Scheme (CRCS) was used. Overall, the small size of the test collections and the lack of precision and recall measurements made it difficult to evaluate these measures. They are presented here due to their ability to use semantic networks. Most work done today is not focused on Boolean requests. However, all of these distance measures are applicable if the natural language request is viewed as a Boolean OR of the terms in the query. It would be interesting to test them against a larger collection with a general semantic network such as WordNet.

3.2 Developing Query Term Based on "Concepts"

Instead of computing the distance between query terms and document terms in a semantic network and incorporating that distance into the metric, the semantic network can be used as a thesaurus to simply replace terms in the query with "nearby" terms in the semantic network. Vectors of "concepts" can then be generated to represent the query, instead of term-based vectorsan algorithm was given that described a means of using this approach to improve an existing Boolean retrieval system. Terms in the original Boolean system were replaced with "concepts". These concepts were found in a semantic network that contained links to the original terms. The paper referred to the network as a thesaurus, but the different relationships existing between terms meet our definition of a semantic network. The system described used an automatically generated semantic network. The network was developed using two different clustering algorithms. The first was the standard cosine algorithm, while the second was developed by the authors and yields asymmetric links between nodes in the semantic net. Users were then able to manually traverse the semantic network to obtain good terms for the query, while the semantic nets were also used to find suitable terms to manually index new documents.

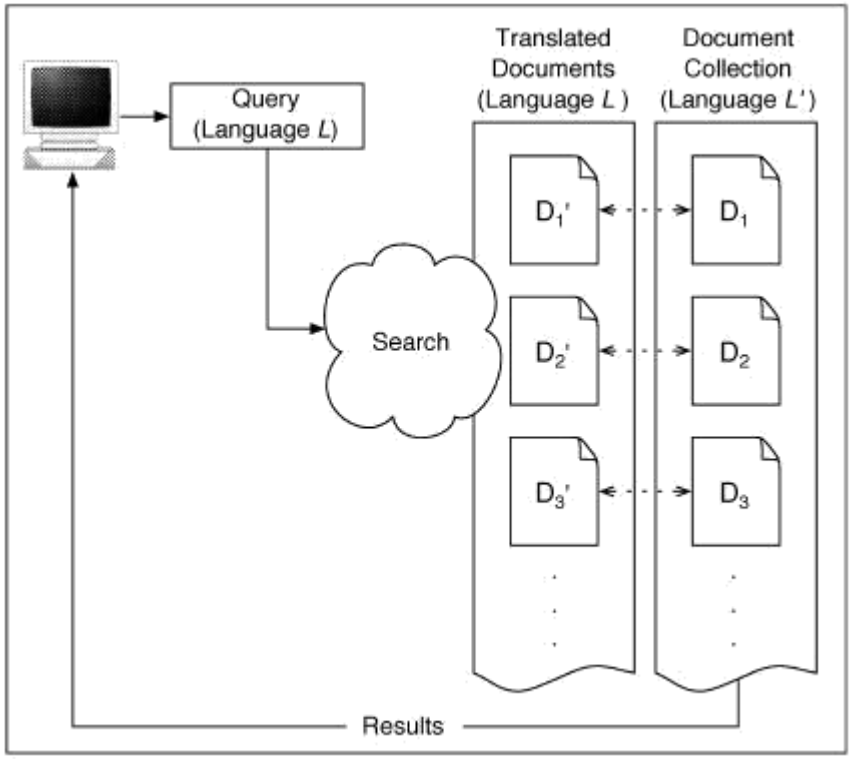

Fig :Translate the Documents

3.3 Ranking Based on Constrained Spreading Activation

Two interesting papers appeared that are frequently referenced in discussions of knowledge-based information retrieval. These describe the GRANT system in which potential funding agencies are identified based on areas of research Retrieval Utilities 139 interest. A manually built semantic network with 4,500 nodes and 700 funding agencies was constructed with links that connect agencies and areas of interest based on the topics agencies are interested in. Given a topic, the links emanating from the topic are activated and spreading activation begins. Activation stops when a funding agency is found. At each step, activation is constrained. After following the first link, three constraints are used. The first is distance. If the path exceeds a length of four ,it is no longer followed. The second is fan-out, if a path reaches a node that has more than four links emanating from it, it is not followed. This is because the node that has been reached is too general to be of much use and it will cause the search to proceed in many directions that are of little use. The third type of constraint is a rule that results in a score for the link. The score is considered an endorsement. Ultimately, the results are ranked based on the accumulation of these scores. An example of one such endorsement occurs if a researcher's area of interest is a subtopic or specialization of a general topic funded by the agency it gets a

positive endorsement. An agency that funds research on database systems will fund research in temporal database systems. More formally: request-funds-for-topic(x) and IS-A(x,y) --t request-funds-for-topic(y) A negative endorsement rule exists when the area of research interest is a generalization of a funding agency's areas of research. An agency that funds database systems will probably not be interested in funding generic interest in computer science. A best-first search is used such that high-scoring endorsements are followed first. The search ends when a certain threshold number of funding agencies are identified. The GRANT system was tested operationally, and found to be superior to a simple keyword matching system that was in use. Searches that previously took hours could be done in minutes. More formal testing was done with a small set of twenty-three queries. However, the semantic network and the document collection were both relatively small so it is difficult to generalize from these results. Overall, the GRANT system is very interesting in that it uses a semantic network, but the network was constrained based on domain specific rules.

3.4 Parsing

The ability to identify a set of tokens to represent a body of text is an essential feature of every information retrieval system. Simply using every token encountered leaves a system vulnerable to fundamental semantic mismatches between a query and a document. For instance, a query that asks for information about computer chips matches documents that describe potato chips. Simple single-token approaches, both manual and automatic, are described. Although these approaches seem crude and ultimately treat text as a bag of words, they generally are easy to implement, efficient, and often result in as good or better effectiveness than many sophisticated approaches measured at the Text Retrieval Conference (TREC).

A step up from single-term approaches is the use of phrases in document retrieval. Phrases capture some of the meaning behind the bag of words and result in two-term pairs (or multi-term phrases, in the general case) so that a query that requires information about New York will not find information about the new Duke of York.
More sophisticated approaches are based on algorithms commonly used for natural language processing (NLP). These include part-of-speech taggers, syntax parsers, and information extraction heuristics. We provide a brief overview of the heuristics that are available and pay particular attention only to those that have been directly incorporated into information retrieval systems. An entire book could be written on this section as the entire field of natural language processing is relevant. Overall, it should be noted that parsing is critical to the performance of a system. For complex NLP approaches, parsing is discussed in great detail, but to date, these approaches have typically performed with no significant difference in performance than simplistic approaches

3.4.1 Single Terms

The simplest approach to search documents is to require manual intervention and to assign names of terms to each document. The problem is that it is not always easy to assign keywords that distinctly represent a document. Also, when categorizations are employed-such as the Library of Congress subject headings-it is difficult to stay current within a domain. Needless to say, the manual effort used to categorize documents is extremely high. Therefore, it was learned

early in the process that manually assigned tokens did not perform significantly better than automatically assigned tokens.

Once scanning was deemed to be a good idea in, the next step was to try to normalize text to avoid simple mismatches due to differing prefixes, suffixes, or capitalization. Today, most information retrieval systems convert all text to a single case so that terms that simply start a sentence do not result in a mismatch with a query simply because they are capitalized. Stemming refers to the normalization of terms by removing suffixes or prefixes. The idea is that a user who includes the term "throw" in the query might also wish to match on "throwing", "throws", etc. Stemming algorithms have been developed for more than twenty years. The Porter and Lovins algorithms are most commonly used. These algorithms simply remove common suffixes and prefixes. A problem is that two very different terms might have the same stem. A stemmer that removes -ing and -ed results in a stem of r for terms red and ring. KSTEM uses dictionaries to ensure that any generated stem will be a valid word. Another approach uses corpus-based statistics (essentially based on term co-occurrence) to identify stems in a language-independent fashion These stemmers were shown to result in improved relevance ranking over more traditional stemmers.Stop words are terms deemed relatively meaningless in terms of document relevance and are not stored in the index. These terms represent approximately forty percent of the document collection. Removing these terms reduces index construction, time and storage cost, but may also reduce the ability to respond to some queries. A counterexample to the use of stop word removal occurs when a query requests a phrase that only con-tains stop words (e.g., "to be or not to be"). Nevertheless, stop word lists are frequently used, and some research was directed solely at determining a good stop word list. Finally, we find that other parsing rules are employed to handle special characters. Questions arise such as what to do with special characters like hyphens, apostrophes, commas, etc. Some initial rules for these questions are given , but the effect on precision and recall is not discussed. Many TREC papers talk about cleaning up their parser and the authors confess to having seen their own precision and recall results improved by very simple parsing changes. However, we are unaware of a detailed study on single-term parsing and the treatment of special characters, and its related effect on precision and recall.

3.4.2 Simple Phrases

Many TREC systems identify phrases as any pair of terms that are not separated by a stop term, punctuation mark, or special character. Subsequently, infrequently occurring phrases are not stored. In many TREC systems, phrases occurring fewer than 25 times are removed. This dramatically reduces the number of phrases which decreases memory requirements. Once phrases are employed, the question as to how they should be incorporated into the relevance ranking arises. Some systems simply add them to the query, while others do not add them to the query but do not include them in the computation of the document length normalization. The reason for this is that the terms were already being considered. Tests using just phrases or terms were performed on many systems. It was found that phrases should be used to augment, not replace the terms. Hence, a query for New York should be modified to search for new, york, and New York. Phrases used in this fashion are generally accepted to yield about a ten percent improvement in precision and recall over simple terms.

3.4.3 Complex Phrases

The quest to employ NLP to answer a user query . In fact, NLP systems were often seen as diametrically opposed to information retrieval systems because the NLP systems were trying to understand a document by building a canonical structure that represents the document. The goal behind the canonical structure is to reduce the inherent ambiguity found in language. A query that asks for information about walking should match documents that describe people who are moving slowly by gradually placing one foot in front of the other. A NLP system stores information about walking and moving slowly with the exact same canonical structure-it does this by first parsing the document syntactically-identifying the key elements of the document (subject, verb, object, etc.) and then building a single structure for the document. Simple primitives that encompass large categories of verbs were proposed such as PTRANS (physically transport), in which John drove to work and John used his car to get to work both result in the same simple structure John PTRANS work. Progress in NLP has occurred, but the reality is that many problems in knowledge representation make it extremely difficult to actually build the necessary canonical structures. The CYC project has spent the last fifteen years hand-building a knowledge base and has encountered substantial difficulty in identifying the exact means of representing the knowledge found in text . A side effect of full-scale NLP systems is that many tools that do not work perfectly for full language understanding are becoming quite usable for information retrieval systems. We may not be able to build a perfect knowledge representation of a document, but by using the same part-of-speech tagger and syntactic parser that might be used by an NLP system, we can develop several algorithms to identify key phrases in documents.

3.4.3.1 Use of POS and Word Sense Tagging

Part-of-speech taggers are based on either statistical or rule-based methods. The goal is to take a section of text and identify the parts of speech for each token. One approach incorporates a pretagged corpus to identify two measures: the frequency a given term is assigned a particular tag and the frequency with which different tag sequences occur . For example, duck might appear as a noun (creature that swims in ponds) eighty percent of a time and a verb (to get out of the way of a ball thrown at your head) twenty percent of the time. Additionally, "noun noun verb" may occur ten percent of the time while "noun noun noun" may occur thirty percent of the time. Using these two lists (generated based on a pretagged training corpus) a dynamic programming algorithm can be obtained to optimize the assignment of a tag to a token for a given step. Rule-based taggers in which tags are assigned based on the firing of sequences of rules are described. Part-of-speech taggers can be used to identify phrases. One use is to identify all sequences of nouns such as Virginia Beach or sequences of adjectives followed by nouns such as big red truck. Another use of a tagger is to modify processing such that a match of a term in the query only occurs if it matches the same part-of-speech found in the document. In this fashion, duck as a verb does not match a reference to duck as a noun. Although this seems sensible, it has not been shown to be particularly effective. One reason is that words such as bark have many different senses within a part of speech. In the sentences A dog's bark is often stronger than its bite and Here is a nice piece of tree bark, bark is a noun in both cases with very different word senses. Some initial development of word sense taggers exists. This work

identifies word senses by using a dictionary based stemmer. Recent work on sense disambiguation for acronyms is found.

3.4.3.2 Syntactic Parsing

As we move along the continuum of increasingly more complex NLP tools, we now discuss syntactic parsing. These tools attempt to identify the key syntactic components of a sentence, such as subject, verb, object, etc. For simple sentences the problem is not so hard. Whales eat fish has the simple subject of Whales, the verb of eat, and the object of fish. Typically, parsers work by first invoking a part-of-speech tagger. Subsequently, a couple of different approaches are employed. One method is to apply a grammar. The first attempt at parsers used augmented transition networks (ATNs) that were essentially non-deterministic finite state automata in which: subject-verb-object would be a sequence of states. The problem is, that for complex sentences, many different paths occur through the automata.
Also, some sentences recursively start the whole finite state automata (FSA), in that they contain structures that have all the individual components of a sentence. Relative clauses that occur in sentences such as Mary, who is a nice girl that plays on the tennis team, likes seafood. Here, the main structure of Mary likes seafood also has a substructure of Mary plays tennis. After ATN s, rule-based approaches that attempt to parse based on firing rules, were attempted.

Other parsing algorithms, such as the Word Usage Parser (WUP) by Gomez, use a dictionary lookup for each word, and each word generates a specialized sequence of states .In other words, the ATN is dynamically generated based on individual word occurrences. Although this is much faster than an ATN, it requires substantial manual intervention to build the dictionary of word usages. Some parsers such as the Apple Pie Parser, are based on light parsing in which rules are followed to quickly scan for key elements of a sentence, but more complex sentences are not fully parsed. Once the parse is obtained, an information retrieval system makes use of the component structures. A simple use of a parser is to use the various component phrases such as SUBJECT or OBJECT as the only components of a query and match them against the document. Phrases generated in this fashion match many variations found in English. A query with American President will match phrases that include President of America, president who is in charge of America, etc. One effort that identified head-modifier pairs (e.g., "America+president") was evaluated against a patent collection and demonstrated as much as a sixteen percent improvement in average precision. On the TREC-5 dataset, separate indexes based on stems, simple phrases (essentially adjective-noun pairs or noun-noun pairs), head-modifier pairs, and people name's were all separately indexed. These streams were then combined and a twenty percent improvement in average precision was observed. To date, this work has not resulted in substantial improvements in effectiveness, although it dramatically increases the run-time performance of the system.

3.4.3.3 Information Extraction

The Message Understanding Conference (MUC) focuses on information extraction-the problem of finding various structured data within an unstructured document. Identification of people's names, places, amounts, etc. is the essential problem found in MUC, and numerous algorithms that attempt to solve this problem exist. Again, these are either rule-based or statistical

algorithms. The first step in many of these algorithms is to generate a syntactic parse of the sentence, or at the very least, generate a part-of-speech tag. Details of these algorithms are found in the MUC Proceedings. More recently the Special Interest Group on Natural Language Learning of the Association for Computational Linguistics held a shared task on Language-Independent Named Entity Recognition. All of the proceedings may be found at. In this task, language independent Retrieval Utilities 145 algorithms were used to process standard test collections in English and German. Named entity taggers identify people names, organizations, and locations. We present a brief example that we created with a rule-based extractor from BBN Corporation to obtain this new document. This extractor works by using hundreds of hand-crafted rules that use surrounding terms to identify when a term should be extracted. First, we show the pre-extracted text-a paragraph about the guitarist Allen Collins.

<TEXT>
Collins began his rise to success as the lightning-fingered guitarist for the Jacksonville band by a group of high school students. The band enjoyed national fame in the 1970's with such hits as "Free Bird," "Gimme Three Steps," "Saturday Night Special" and Ronnie Van Zant's feisty "Sweet Home Alabama."

</TEXT>
The following output is generated by the extractor. Tags such as PERSON and LOCATION are now marked.
<TEXT>

<ENAMEX TYPE="PERSON">Collins<IENAMEX> began his rise to success as the lightning-fingered guitarist for the <ENAMEX TYPE="LOCATION">Jacksonville<IENAMEX> bandformed in <TIMEX TYPE="DATE"> 1 966<ITIMEX> by a group of high school students. The band enjoyed national fame in the <TIMEX TYPE="DATE">1970s <lTIMEX> with such hits as "Free <ENAMEX TYPE="PERSON"> Bird <IENAMEX>," "Gimme Three Steps," "Saturday Night Special" and <ENAMEX TYPE="PERSON">Ronnie Van Zant<IENAMEX>'s feisty "Sweet Home <ENAMEX TYPE="LOCATION">Alabama<IENAMEX>."

</TEXT>

In this example, and in many we have hand-checked, the extractor performs welL Many extractors are now performing at much higher levels of precision and recall than those of the. However, they are not perfect. Notice the label of PERSON being assigned to the term "Bird" in the phrase "Free Bird." Using extracted data makes it possible for a user to be shown a list of all person names, locations, and organizations that appear in the document collection.These could be used as suggested query terms for a user.The simplest use of an extractor is to recognize key phrases in the documents. An information retrieval system could incorporate extraction by increasing term weights for extracted terms. Given that extractors are only recently running fast enough to even consider using for large volumes of text,research in the area of using extractors for information retrieval is in its infancy.

3.5 Cross-Language Information Retrieval

Cross-Language Information Retrieval (CUR) is quickly becoming a mature area in the information retrieval world. The goal is to allow a user to issue a query in language L and have that query retrieve documents in language *L'*. The idea is that the user wants to issue a single query against a document collection that contains documents in a myriad of languages. An implicit assumption is that the user understands results obtained in multiple languages. If this is not the case, it is necessary for the retrieval system to translate the selected foreign language documents into a language that the user can understand.

3.5.1 Introduction

The key difference between CUR and monolingual information retrieval is that the query and the documents cannot be matched directly**.** In addition to the inherent difficulty in matching the inherent style, tone, word usage, and other features of the query with that of the document, we must now cross the language barrier between the query and the document. focuses on the core problems involved in crossing the language barrier..

3.5.1.1 Resources

Numerous resources are needed to implement cross-language retrieval systems. Most approaches use bilingual term lists, term dictionaries, a comparable corpus or a parallel corpus. A *comparable corpus* is a collection of documents in language L and another collection about the same topic in language *L'*. The key here is that the documents happen to have been written in different languages, but the documents are not literal translations of each other. A news article in language L by a newspaper in a country which speaks language L and an article in language *L'* by a newspaper in a country which speaks language *L'* is an example of comparable documents. The two newspapers wrote their own article; they with comparable corpora are that they must be *about the same topic.* A book in French on medicine and a book in Spanish on law are not comparable. If both books are about medicine or about law they are comparable. We will discuss CUR techniques using a comparable corpus. A *parallel corpus* provides documents in one language L that are then direct translations of language *L'* or vice versa. The key is that each document is in language L is a direct translation of a corresponding document in language *L'*. Hence, it is possible to align a parallel corpus at the document level, the paragraph level, the sentence level, the phrase level, or even the individual term level. Legislative documents in countries or organizations that are required to publish their proceedings in at least two languages are a common source of parallel corpora. In general, a parallel corpus will be most useful if it is used to implement cross-language retrieval of documents that are in a similar domain to the parallel corpus. Recent work shows that significant effectiveness can be obtained if the correct domain is selected. We discuss parallel corpus CUR techniques and also note that even within a single language such as Arabic, there are many different character sets. Language processing resources exist to not only detect a language but also to detect a character set. Cross-language systems often struggle with intricacies involved in working with different character sets within a single language. Unicode was developed to map the character representation for numerous scripts into a single character set, but not all electronic documents are currently stored in Unicode.

3.5.1.2 Evaluation

Different measures are used to evaluate the performance of cross-language information retrieval systems. The most obvious is simply to compute the average precision of the cross-language query.
Another approach is to compute the percentage of monolingual performance. This can occasionally be misleading because the techniques used to achieve a given monolingual performance may be quite different than those used for cross-language performance. Straightforward techniques typically result in 50% of monolingual performance, but the CUR literature contains results that exceed 100% because of the inherent query expansion that occurs when doing a translation. We note that queries with relevance judgments exist in Arabic, Chinese, Dutch, Finnish, French, German, Italian, Japanese, Korean, Swedish and Spanish. These have been used at various evaluations at To cross the language barrier, we must answer four core questions:

• What should be translated? Either the queries may be translated, the documents, or both queries and documents may be translated to some internal representation.

• Which tokens should be used to do a translation (e.g.; stems, words, phrases, etc.)?

• How should we use a translation? In other words, a single term in language L may map to several terms in Language Lt. We may use one of these terms, some of these terms, or all of these terms. Additionally, we might weight some terms higher than other terms if we have reason to believe that one translation is more likely than another.

• How can we remove spurious translations? Typically, there are spurious translations that can lead to poor retrieval. Techniques exist to remove these translations.As with monolingual retrieval, various strategies and utilities exist for cross language retrieval. As with the monolingual retrieval we organize the chapter according to strategies and utilities. Again, a strategy will take a query in language L and identify a measure of similarity between the query and documents in the target language L'. A utility enhances the work of any strategy.