CONSTRUCTION OF AUTOMATED SYSTEM FOR INFORMATION EXTRACTION AND TEXT CATEGORIZATION

Abdul Kareem M. Radhi
Al-Nahrain University, College of Information Engineering.
E-Mail: kareem-m-radhi@yahoo.com.

Abstract

This paper presents a research on the field of AI via studying machine learning for natural language understanding. One important part of the process of understanding a text consists on apprehending its underlying interrelations of concepts [1]. Learning is to gain knowledge or understanding or skill in by study instruction or experience and modification of behavioral tendency by experience. We might say, very broadly that a machine learns when it changes its structure, or program, or data (based on its inputs or in response to external information) in such a manner that its expected future performance improves.

The proposed system aims to extracts concepts from text written in English natural language text. In spite of the complexity of English language the proposed system offer intelligent user interactive interface that create structured query and complete the concepts relations before extracting the desired information from one or a lot of documents in specific domain in the form of templates consist a number of slots using inductive logic programming (ILP).

Keywords: (AI: Artificial Intelligence, ILP: Inductive logic programming), (ATN: augmented transition network), (POS: part of speech tagging), (Entropy),

Introduction

In the last decades years there is an explosive growth in the amount of information available on networked computers much of it in the form of natural language documents. Answering many questions about available information requires a deeper "understanding" of natural language. One way of providing more "understanding" is with information extraction.

The extracted information can offer to the reader a global picture describing the concept relations in the domain and then be stored in a database which could then be queried using either standard database query languages or a natural language database interface. Moreover it can be used in educational fields. Systems for this task require significant domain specific knowledge and are time consuming and difficult to build by hand.

There are a large number of applications in which a large corpus of texts must be searched for particular kinds of information and that information must be entered into a database for easier access [8]. In order to construct a system, which can automate learning from a text, we propose to use inductive logic programming technique.

Inductive logic programming (ILP) is the intersection of machine learning and logic programming [3]. It is very applicable in finding embedded relations between different entities in same and different domains. Inductive logic programming (ILP) studies the induction of rules in first order logic. I used rule-based learning, which is one of the approaches in ILP to analyze and classify positive and negative examples for input text to belonging to different documents written in natural language. I try to have little information about the feature vector in order to enforce the applicability of the system to another domain.

Text Processing:

English language like any other natural language is a complex phenomenon, such that the words formatted from different roots and derivate according to some English languages law, besides different ambiguities, which may be conducting the sentences, and the morphology of the word [7]. Understanding natural language requires a large amount of knowledge about morphology, syntax, semantics, and pragmatics as well as general knowledge about the world. Acquiring and

encoding all of this knowledge is one of the fundamental impediments to developing effective and robust language-processing systems [7].

The motivation of this work that the human when searching for a specific fragment of information which is may be embedded in a whole document, he did not read the whole document in detail at all, but he read the whole document first, and then looking carefully for the desired information [5]. From that we first process the text in the documents exploiting any useful information in the syntactic and semantic analysis of the text like part of speech tagging of every word in the sentence and the semantic role of the noun phrases in the text, then proceed with rule based learning in order to construct general concept model describing the entities found and there relations.

I apply and test my approach for the first module on the following biological kingdom text:

{"A predator is an animal that eats other animals. For example, lions eat gazelles and zebras. These are the preys; Humans are predators that can be preys too.
Vegetarian animals are usually preys, while predators are obviously carnivores.
Humans are carnivores, because they eat both animals and vegetables.
Another difference between these two groups is that predators are small and fast, while the preys are bigger and slower."}

Processing the whole text must begin with defining the entities and some knowledge about the domain. We will discuss the proposed system structure which involving in general three stages as follows:

Applied algorithm

1. Providing the system with text that contains relevant features of the target domain.
2. Segment the text according to discourse analysis constraints.
3. The text is tagged using a Word Net database to find all parts of speech to which a word may belong and to classify words in the parsing process.
4. Obtain lexical classification of the words in the parsing process.
5. Morphological analysis via pipeline morphemes. Morphology is the study of ways words are built from smaller meaning bearing units, called morphemes.
6. Syntactic analysis according to augmented grammar.
7. Semantic analysis for the successfully syntactic analysis sentences using ATN techniques.
8. Defining the seed concepts to find the relevant sentences belonging to the domain.
9. Extract the primitive facts and relations between the defined concepts.
10. Using inductive logic programming (ILP) to complete the relations between the concepts .This is done via interactive dialogue between the user and the machine or the system. So we need interface module system.
11. Find hierarchal concept schema.

The Word-Net or System lexicon is classified generally according to words noun and verbs syntactic and semantic features:

- *Verb (stem,voice,tense[subjectgender,object gender],number).*
- *Noun (stem, definition, gender, number, adjectivability).*

Filtering stage

This module uses superficial techniques to filter out the sentences that are likely to be irrelevant; thus turning the text into a shorter text that can be processed more quickly.

NLP Techniques can be used to enrich the information given to machine learning algorithm, or to filter the input. For example part of speech tags may be included in the sentence word relation, and may be also be used as filtering mechanism. The isa(Class, Class) relation of the ontology can be used as background theory. The second resource that can be used is a mapping between a concept and the form of words that it may be associated with in a text. In this stage there are small another steps aiming to segment the English text to distinguished sentences, and these sentences then segmented to its components (words). The segmentation takes

into account the punctuations and separated characters. In order to define the roles of every word in the sentence, I found that it is necessary to look up at the part of speech of tagging (POS) of every word in the sentence.

For example a predator is noun and eats are a verb. This step will feed the morphological analyzer stage. So in order to define the source of every word we will analyze and drive its roots by auxiliary morphological pipeline (containing the suffixes and prefixes), which could be introduced to English words. Before parsing every sentence in the text we will define a specific augmented grammar for English language, which can cover the formalism of its generation. The proposed system uses augmented transition network (ATN) techniques to analyze these sentences syntactically to feed the case role mapping of every noun in the noun phrase and define the main verb in these sentences. The ambiguities, which could arise in the English language, may change the meaning of the word in the sentence. One word may have different meaning, and different words may have the same meaning. There is another ambiguous case in the sentence, which could arises and affect the meaning of it directly. So we process the sentence in another stage to disambiguate it to reach the correct meaning of the context. Therefore we saw that word sense disambiguation and anaphora resolution another stages which they enforce the processing to reach the meaning or semantic phase of the text. In this point we apply decision tree to satisfy our goal.

Anaphora resolution

Traditional text disambiguation through anaphora resolution is essentially founded on a model of anaphora resolution based on history lists [1]. A history list is a list of discourse entities generated from the preceding sentences. An anaphor is an expression, which can not have independent reference, but refer to another expression, the so-called antecedent. Identifying multiple phrases that refer to the same entity is another difficult language-processing problem. Anaphora resolution can be treated as a categorization problem by classifying pairs of phrases as either co-referring or not. Given a corpus of texts tagged with co-referring phrases, positive examples can be generated as all co-referring phrase pairs and negative examples as all phrase pairs within the same document that are not marked as co-referring. Both decision-tree and instance-based methods have been successfully applied to resolving various types of anaphora. For example:

"*A predator is an animal <u>that</u> eats other animals."*

"<u>That</u>" is relative clause, which refer to the noun {predator}.

"Lions eat gazelles and zebras. <u>These</u> are the preys".
<u>These</u> refer to preys.

Computational linguistic system that learns to transform natural language sentences into semantic representations has important practical applications in building natural language interfaces. There is a long tradition of representing the meaning of natural language statements and queries in first order logic [3]. A modern approach semantic parsing refers to the task of mapping a natural language sentence into a detailed semantic representation or logical form.

Inductive logic programming is appropriate for this learning task for several reasons. ILP provides a natural representation of the relations to be learned and background knowledge can easily be represented.

Semantic grammar

The first technique for combining syntactic and semantic processing involves collapsing them into a single uniform framework either a context–free grammar except that it uses semantic categories for terminal symbols [4]. The proposed system use ATN techniques to combine syntactic and semantic knowledge in order to analyze the sentences in the text. A sentence like "a predator eats the prey" can be processed by this technique:

"Predator is a living – entity type animate. Eats is present transitive verb-type patient – subclass stomach. So the prey is living entity also animate."

"*A predator eats the prey."*

Rule: s-mod (Declaration),

First-NP NP1,

Det(the),

Noun(type:animate,sub_type(animal),gender(predator),

Verb:eats;tens-present;

Type-transitive,

Sem-ingest;stomach

Agent(a predator),

Ref-number(singular),

Theme-eating,

Object:NP2,

Patient- animate,living_entity,type-animal,sub_type:food,

Sem_type(edible),

Obj(the prey).

Rule Induction

Rule induction is viewed as part of semi automated process, which necessarily includes human involvement. A domain expert, and /or a knowledge engineer, may need to understand the suggested extraction rules [20]. They may also wish to refine the suggested rules. Alternatively, the rules can be learned and applied incrementally; with the human having the role of correcting the derived facts /markup ones an initial set of rules is learned.

Possible solutions for instances in the document can be translated into a set of rules by creating a separate rule for each instance like nodes in decision trees. However, rules can also be directly induced from training data using a variety of algorithms. The general goal is to construct the smallest rule-set (the one with the least number of symbols) that is consistent with the training data. The standard approach is to use a form of greedy set covering, where at each iteration; a new rule is learned that attempts to cover the largest set of examples of a particular category without covering examples of other categories. These examples are then removed, and additional rules are learned to cover the remaining examples of the category. Explanation based learning system are known for their ability to produce complete concept representation from a single training instance [5]. This is in contrast to inductive learning techniques that incrementally build a concept representation in response to multiple training instances. A fact between two concepts specifies an existent relation between them [1].

"A predator is an animal that eats other animals".

I deduce from this that all predators will eat other animals which are the preys.

Generally, this selection is identified by the main verb in the sentence. And where can we find the *binary predicate* concepts, the subject represents a concept whose relation or property, is passed in a sentence. Thus, an object, or qualifier, shall be considered as the second concept in a binary predicate. One of the main jobs of discourse analysis is to determine logical relationships between domain objects according to domain specifications of what relationships are reportable. Rule induction is viewed as part of a semi automated process, which necessarily includes human involvement. A domain expert, and\or a knowledge engineer, may need to understand the suggested extraction rules. They may also wish to refine the suggested rules. Alternatively, the rules can be learned and applied incrementally, with the human having the role of correcting the derived facts once an initial set of the rules is learned [16].

Fig. (1) represents a sample domain comprising positive and negative examples distributed randomly according to the embedded relations and facts belonging to each of them.

Concept Model Construction

Analyzing the examples, which represented by the entities in the documents and its relation is the first step in defining the concepts of the domain. We represent these examples via the first order logic which be considered as a powerful form realizing the structure and its contents. We use rule-based learning to define the concepts in the domain. Defining a concept must be proceeded by defining the *relevant* sentences in the text. These relevant sentences could be representing

as positive examples. Then we must classify relevant and non-relevant sentences.

Inductive logic programming aim to construct concepts of the entities found in the documents by specifying attributes generated to specific entity progressing in forward manner from specific to general as shown in Figs. (1, 2, 3, and 4). The standard approach is to use a form of greedy set covering, where at each iteration; a new rule is learned that attempts to cover the largest set of examples of a particular category without covering examples of other categories. These examples are then removed, and additional rules are learned to cover the remaining examples of the category.

Constructing concept model for the desired domain will be useful in the future when we consulting with another text for the same domain.

Interacting with the user the system can extract the facts and rules, some of them are:

isa(predator, animal).
isa(prey, animal).
eats(predator, prey).

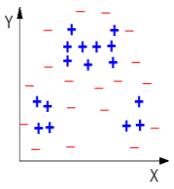

Fig. (1)

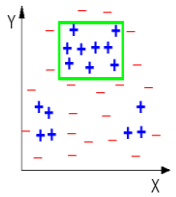

Fig. (2)

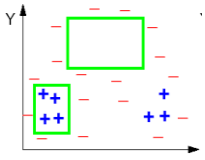

Fig. (3)

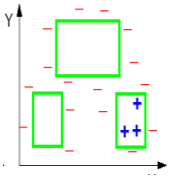

Fig.(4)

can_be(predator, human).
property(predator, bigger).
property(prey,slower).
eat(lions, gazelles).

Evaluation

One obvious method to evaluate a computational theory would be to run the program to see how well it performs, for example if the program is meant to answer questions about a database of facts, you might ask questions to see how good it at producing the correct answers [4]. We can evaluate the results with the following metric functions:

Entropy (impurity, disorder) of a set of examples, relative to a binary classification is:

$$\text{Entropy}(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-) \quad \ldots\ldots (1)$$

Where P_+ is the proportion of positive examples in *S* and P_- the proportion of negatives.

If all examples belong to the same category, entropy is 0, Where P_+ is the proportion of positive examples in (S) and P_- is the proportion of negatives.

For multiple category problems with C categories, entropy can be generalized to:

$$\text{Entropy}(S) = \sum_{I=1}^{c} - P_i \log_2(P_i) \quad \ldots\ldots (2)$$

Where P_i is the proportion of category i examples in S.

The information gain of an attribute is the expected as reduction in entropy caused by portioning on this attribute:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values of } A} \frac{|Sv|}{|S|} \text{Entropy}(S_v) \quad \ldots\ldots (3)$$

Where S_v is the subset of *S* for which attribute *A* has value *v* and the entropy of the partitioned data is calculated by weighting the entropy of each partition by its size relative to the original set.

Performance is measured in terms of recall and precision, where recall is the percentage of positive instances that were formed by rule base. Precision measures the percent correct of instances extracted by the rule base [8].

$$\text{Recall} = \frac{\text{Number of correctly predicted entities}}{\text{Number of entities that should have been found}} \quad \ldots\ldots (4)$$

$$\text{Precision} = \frac{\text{Number of correctly predicted entities}}{\text{Number of all entities predicted}} \quad \ldots\ldots (5)$$

System Design And Implementation

An optimize representation of concepts as shown in [semantic network] in which nodes are concepts and arcs are relations. This work is composed of two autonomous modules:

1. Concept extraction.
2. Text categorization.

As shown in Figs. [5] to [14] respectively, the system begins with a text reader that extracts concept relations through the parsing process; then a concept map constructor that uses Machine Learning inspired algorithms to complete the map through dialogue with the user. For the first module we propose to extract, from the parsing process of text files, relations between inherent constitutes. They represent these relations in a Logical Form (LF) and summarize the relevant information in normalized templates that are adaptable to different user needs. On the other hand, the system extracts binary predicates from a text file using syntactic and discourse knowledge. These predicates will feed other stages that infer knowledge based on machine learning. The system doesn't need any preview knowledge about the discussed domain. It receives a text as initial base of the information extraction after tagging a text file using the external lexicon.

It builds a predicates that map relations between two concepts from parsing of sentences. Its practical goal is to be able to extract from utterances, for example "Cows, as well as rabbits, eat only vegetables, while humans eat also meat", the predicates:

{ eat (cow, vegetables),
  eat (rabbit, vegetables),
  eat (human, vegetables),
  eat (human, meat)}

This will form for these relations of the sentences its concept map. As we see later, then the system will be able to produce rules, for example:

"isa(X, vegetarian):-eat(X, vegetables)".

Although we feel the limit of two arguments per predicate as highly restrictive, we imposed ourselves this constraint as a development condition.

The system is responsible for the interactive construction of concept maps. A concept map in this system consists of a set of binary predicates that represent relations between concepts. The system was designed to accept any relation and concept the user inputs, building gradually in parallel the corresponding ontological "isa" tree and learning some particularities of the domain. It applies two different techniques of Inductive Learning in order to extract regularities on the relations and concepts of the map: a best current hypothesis based algorithm to learn the categories of the arguments of each relation; and an Inductive Logic Programming based algorithm to learn the contexts that are recurrent in each relation. The input for both algorithms consists of the binary relations of the concept map, each of them being a new isolated example (positive or negative) to the process of learning. Sometimes, finding concepts in a dependent sentence isn't clear for an automatic tool. If some ambiguity arises in this process, the system shall apply Anaphora Resolution and/or Context-Dependent Analysis at the Co-Reference Disambiguation Module. Traditional text disambiguation through Anaphora Resolution is essentially founded on a model of Anaphora Resolution based on History Lists.

Lexicon gives the system only a lexical classification of the words in the parsing process, leaving out other Lexicon information such as antonyms, hyponyms, hyponyms, and meronyms. Lexicon is just used to supply lexical verification of words present in sentences. Since, in real world, concepts in a text are not named every time in the same way, the system uses synonymy semantic relationship from Lexicon to identify the concepts that were already referred to before with a different name, for example, the concept. It is important to say that Lexicon itself is organized according to synonymy: words are joined together in lots called synsets (essential structures of this database).

The result of this whole process is a list of predicates that represent the concept interrelations the system has detected. This list is then the input to the system, which, through dialogue with the user, will try to clarify as much important points as it can.

The system can use not only affirmative and declarative sentences, but moreover imperative, "yes, no", and question sentences.

In spite of the complexity of parsing raw or novel sentences, such that we could not find a parser that can process more than 75% of these sentences, the proposed work overcomes this problem by combining fragments of the parsing process. An interesting point would be to analyze negative sentences as negative examples of the knowledge base, as well as to include temporal reference to establish a non-monotonic database.

Motivation and Text Processing

Refer to principles of Top-Down design and in order to extract any concepts from natural language text, the design of system is a cascade of transducers or modules that at each step add structure and often lose information, hopefully irrelevant, by applying rules that are acquired manually and/or automatically. As an example, consider the parsing module. The parser is the transducer. The input is the sequence of words or lexical items that constitute the sentence. The output is a parse tree of the sentence. This adds information about predicate-argument and modification relations.

The motivation of our works is that the human when searching for a specific fragment of information which may be embedded in a whole document, he does not read the whole document in detail at all, but he read the whole document first, and then looks carefully for the desired information.

From that we first process the text in the documents exploiting any useful information in the syntactic and semantic analysis of the text like part of speech, tagging of every word in the sentence and the semantic role of the noun phrases in the text, then proceed with rule based learning in order to construct general concept model describing the entities found and their relations.

Understanding natural language requires a large amount of knowledge about morphology, syntax, semantics, and pragmatics as well as general knowledge about the world. Acquiring and encoding all of this knowledge is one of the fundamental impediments to developing effective and robust language-processing systems.

Processing the whole text must begin with defining the entities and some knowledge about the domain. We will discuss the proposed system structure starting with the proposed applied algorithm as follows:-

Algorithm Divide and Conqure

1- Search for the fundamental concepts.
2- Process the “isa-tree”.
3-Find links for each relation for pairs of categories.
4-Match between concepts to find category in the tree.
5-Interact with the user to get his observation of the concepts domain, to point the most general the categories.
6-Construct semantic network of the concepts starting from general concept to the specific concept of the observation. Therefore we have binary predicates that represent the pairs of categories of the arguments that cover the positive examples and avoid the negative ones.

Applied algorithm

1. Provide the system with text that contains relevant features of the target domain
2. Segment the text according to discourse analysis constraints.
3. . The text is tagged using a lexicon to find all parts of speech to which a word may belong and to classify words in the parsing process.
4. Morphological analysis via pipeline morphemes. Morphology is the study of ways words are built from smaller meaning bearing units, called morphemes.
5. Syntactic analysis according to augmented grammar. And obtain lexical classification of the words in the parsing process.
6. Semantic analysis for the successfully syntactic analysis sentences using augmented transition network.
7. Define the seed concepts to find the relevant sentences belonging to the domain.
8. Extract the primitive facts and relations between the defined concepts.
9. Use inductive logic programming (ILP) to complete the relations between the concepts .This is done via interactive dialogue between the user and the machine or the system and according to divide and conquer algorithm. So we need interface module system.
10. Find hierarchal concept schema.

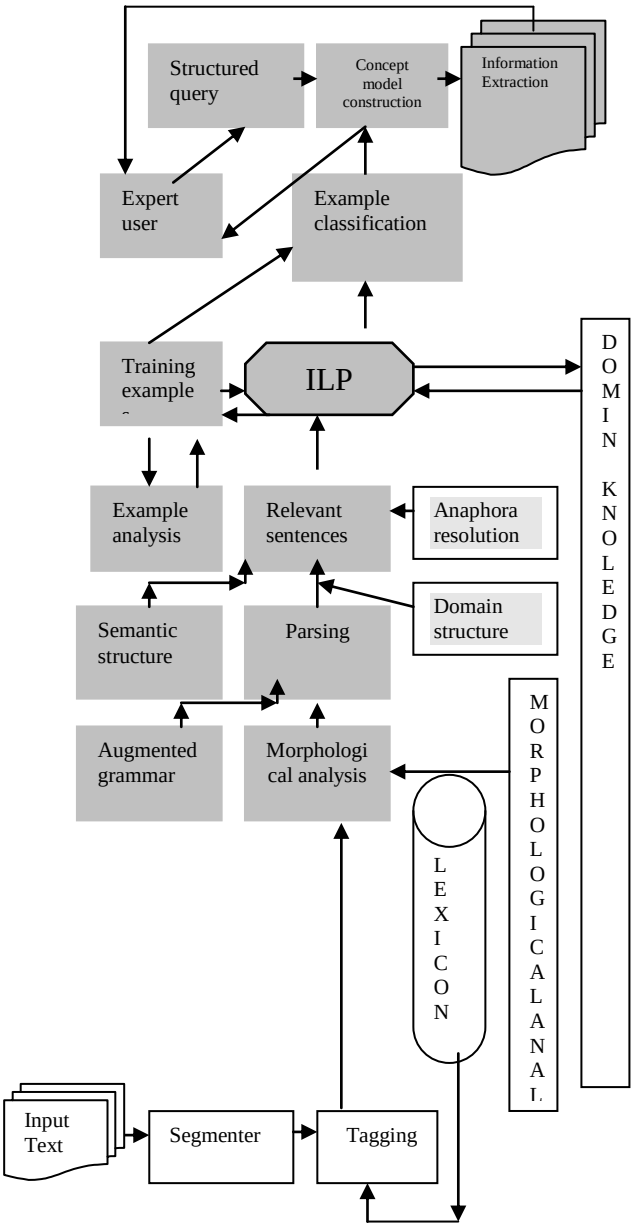

Fig. (5) : Concept model construction.

In this module the system uses superficial techniques to filter out the sentences that are likely to be irrelevant; thus turning the text into a shorter text that can be processed more quickly.

NLP Techniques can be used to enrich the information given to machine learning algorithm, or to filter the input. For example part of speech tags may be included in the sentence word relation, and may be also be used as filtering mechanism. The *isa(Class, Class)* relation of the ontology can be used as background theory. The second resource that can be used is a mapping between a concept and the form of words that may be associated with in a text.

Segmentation

This stage aims to segment the text to distinguish sentences as shown in Fig. (6), these sentences are then segmented to their components (words). The segmentation takes into account the punctuations and separated characters as keywords.

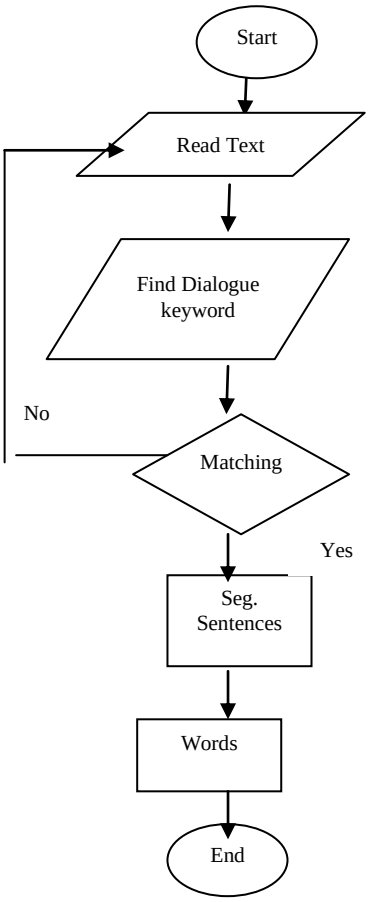

Fig. (2) : Segmentation flowchart.

Tagging

In order to define the roles of every word in the sentence, we find that it is necessary to look up at the part of speech of tagging *(POS)* of every word in the sentence as shown in Fig.(7). For example a predator is noun and eats are a verb.

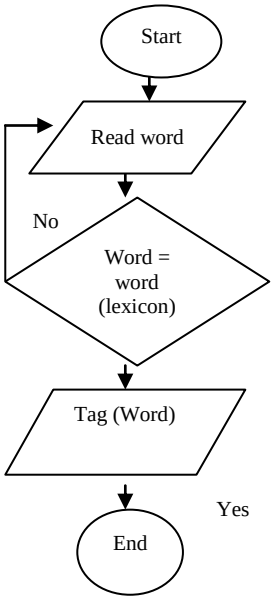

Fig. (7) : Tagging.

Morphological Analysis

The above step will feed the morphological analyzer stage. So in order to define the source of every word we will analyze and drive its roots by auxiliary morphological pipeline (containing the suffixes and prefixes), which could be introduced to English words. In this case we can recognize hyponyms and synonyms of different words; Fig. (8) describes this step.

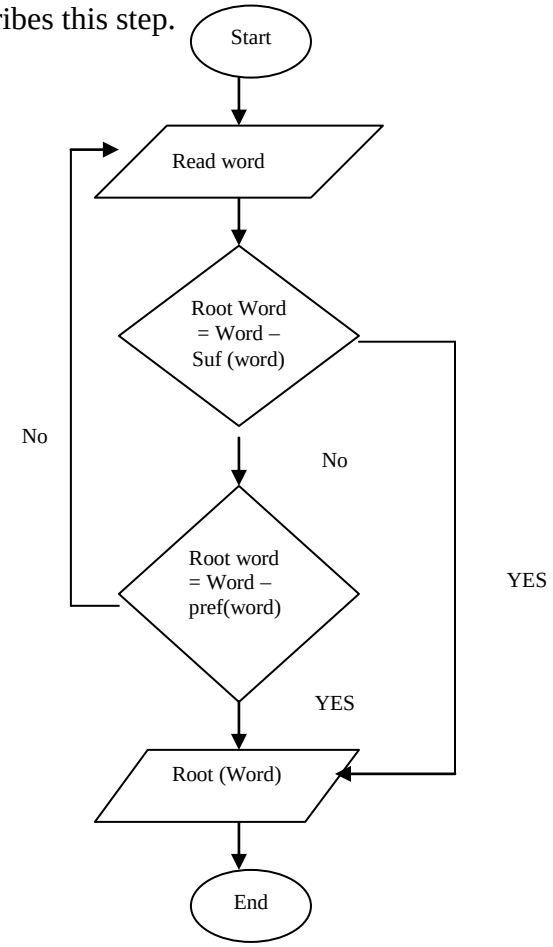

Fig.(8) : Morphological analysis.

Syntax Analysis and Augmented Grammar

Before parsing every sentence in the text we will define a specific augmented grammar for natural language, which can cover the formalism of its generation. Moreover we use rule base and ***ATN*** techniques to analyze these sentences syntactically, feed the case role mapping of every noun in the noun phrase and define the main verb in these sentences, Fig. (9) describes the processes of syntax analysis. According to divide and conquer algorithm, the system does not process the hole sentence directly, but fragments it to a chain of phrases. These phrases will be checked to decide which of it match the rules of augmented grammar. These matching phrases will be joined to form the parsing sentences, such that the last becomes the entries to semantic analysis process. These matching phrases will be joined to form the parsing sentences, such that the last becomes the entries to semantic analysis process.

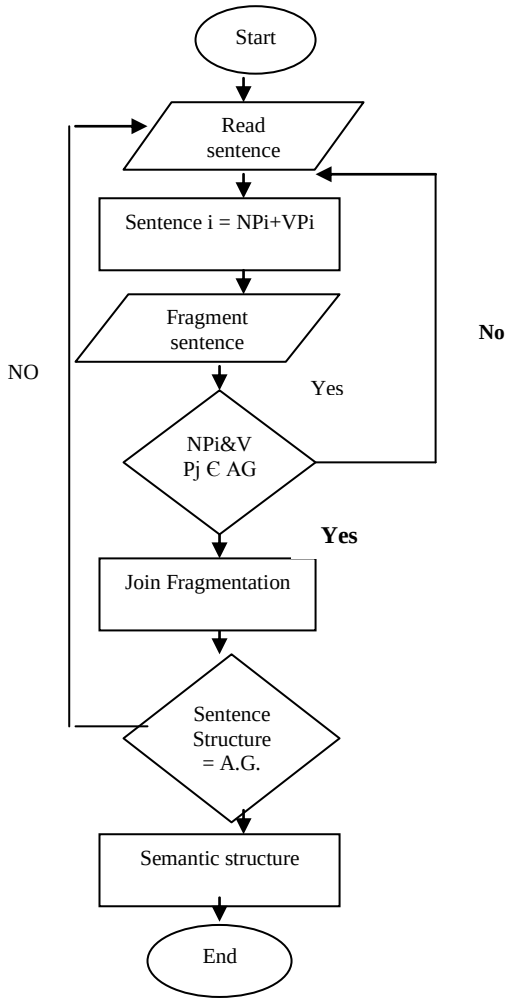

Fig. (9) : Sentence analysis.

The ambiguities, which could arise in the natural language, may change the meaning of the word in the sentence. One word may have different meanings, and different words may have the same meaning. There is another ambiguous case in the sentence, which could arise and affect the meaning of it directly. So we process the sentence in another stage to disambiguate it to reach the correct meaning of the context. Therefore we see that word senses disambiguation and anaphora resolution, another stages, which enforces the processing to reach the meaning or semantic phase of the text. In this point we apply decision tree to satisfy our goal.

Anaphora Resolution

Identifying multiple phrases that refer to the same entity is another difficult language-processing problem. Anaphora resolution can be treated as a categorization problem by classifying pairs of phrases as either co-referring or not. Given a corpus of texts tagged with co-referring phrases, positive examples can be generated as all co-referring phrase pairs and negative examples as all phrase pairs within the same document that is not marked as co-referring. Both decision-tree and instance-based methods have been traditional text disambiguation through anaphora resolution is essentially founded on a model of anaphora resolution based on history lists. A history list is a list of discourse entities generated from the preceding sentences as shown in Fig. (10).

An anaphor is an expression, which can not have independent reference, but refers to another expression, the so-called antecedent. Successfully applied to resolving various types of anaphora. Computational linguistic system that learns to transform natural language sentences into semantic representations has important practical applications in building natural language interfaces. There is a long tradition of representing the meaning of natural language statements and queries in first order logic. A modern approach of semantic parsing refers to the task of mapping a natural language sentence into a detailed semantic representation or logical form. Here in Fig. (10), the history list is a list of all subject sentences, and sf ($subject_i$) is the syntactic, semantic features of sentence subject "i ".

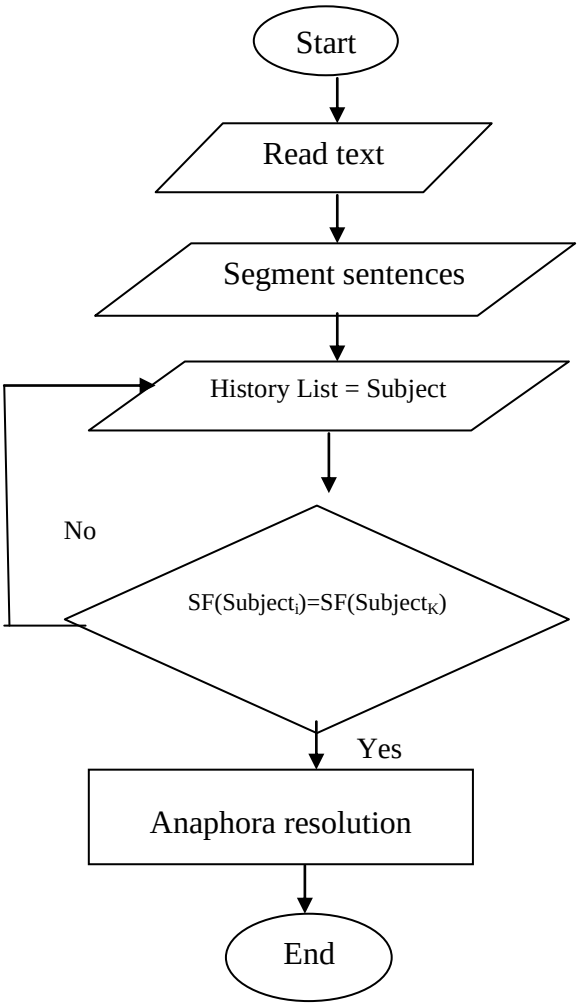

Fig. (10) : Anaphora resolution flow chart.

Semantic Grammar

The first technique for combining syntactic and semantic processing involves collapsing them into a single uniform frame work either a context–free grammar except that it uses semantic categories for terminal symbols. Using ATN to combine syntactic and semantic knowledge in order to analyze the sentences in the text. Fig. (11) describes this technique.

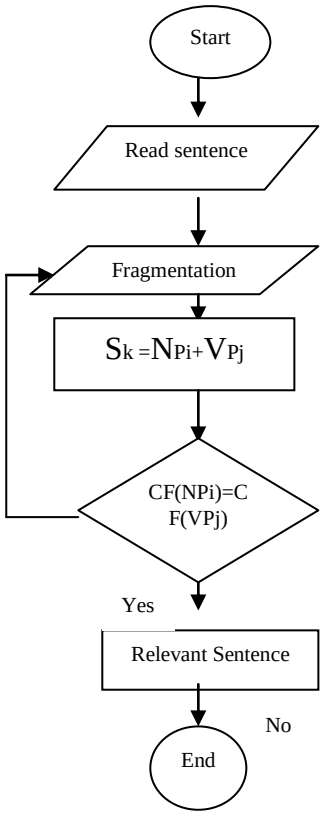

Fig. (11) : Semantic analysi.

Relevant Sentences

This activity consists of selecting adequate linguistic tools and applying them to texts. During the linguistic step, the system has to choose the terms and the lexical relations (hyperonyms, hyponym, and synonyms). Fig. (12) describes this activity.

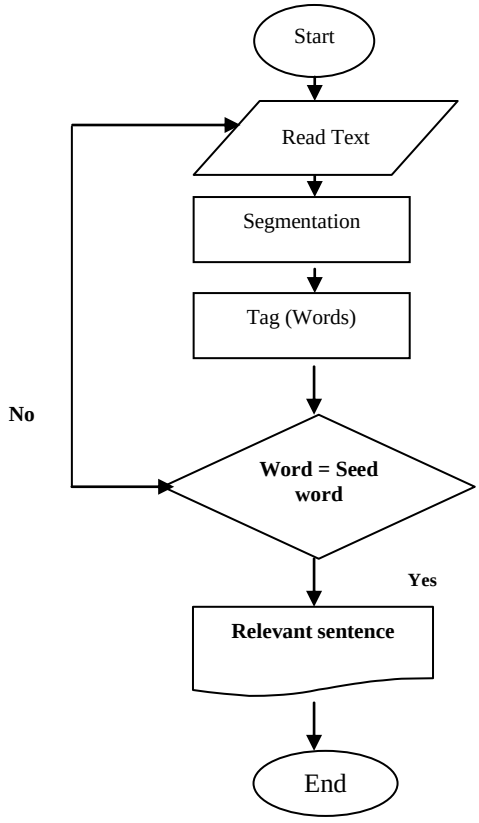

Fig.(12) :Relevant sentences.

Rule Induction

Rule induction is viewed as part of semi automated process, which necessarily includes human involvement. A domain expert, and /or a knowledge engineer, may need to understand the suggested extraction rules [20]. They may also wish to refine the suggested rules. Alternatively, the rules can be learned and applied incrementally; with the human having the role of correcting the derived facts /markup ones an initial set of rules is learned.

Possible solutions for instances in the document can be translated into a set of rules by creating a separate rule for each instance like nodes in decision trees. However, rules can also be directly induced from training data using a variety of algorithms. The general goal is to construct the smallest rule-set (the one with the least number of symbols) that is consistent with the training data. The standard approach is to use a form of greedy set covering, where at each iteration; a new rule is learned that attempts to cover the largest set of examples of a particular category without covering examples of other categories.
These examples are then removed, and additional rules are learned to cover the remaining examples of the category.

Explanation based learning system are known for their ability to produce complete concept representation from a single training instance [5].This is in contrast to inductive learning techniques that incrementally build a concept representation in response to multiple training instances.

A fact between two concepts specifies an existent relation between them [1].
"a predator is an animal that eats other animals".
We deduce from this that all predators will eat other animals which are the preys.

Generally, this selection is identified by the main verb in the sentence. And where can we find the *binary predicate* concepts, the subject represents a concept whose relation or property, is passed in a sentence. Thus, an object, or qualifier, shall be considered as the second concept in a binary predicate.
Fig.(2) represents a sample domain comprising positive and negative examples distributed randomly according to the embedded relations and facts belonging to each of them.

Concept Model Construction

Analyzing the examples, which are represented by the entities in the documents and their relation, is the first step in defining the concepts of the domain. We represent these examples via the first order logic which is considered as a powerful form realizing the structure and its contents. We use rule-based learning to define the concepts in the domain. Defining a concept must be proceeded by defining the relevant sentences in the text. These relevant sentences could be represented as positive examples. Then we must classify relevant and non-relevant sentences. After specifying rules as shown in Fig. (13), the system checks features of every positive and negative training example. The Hoarn-clause definition (Rule induction) covers all examples that match its conditions. Then concept model is a schema of all positive and not negative examples in the desired domain. Generalization is an important feature that characterizes the implemented system, where every induced covering rule is a general rule that match new positive examples of the desired domain in the future. Then the general rule will be induced to cover largest number of positive examples.

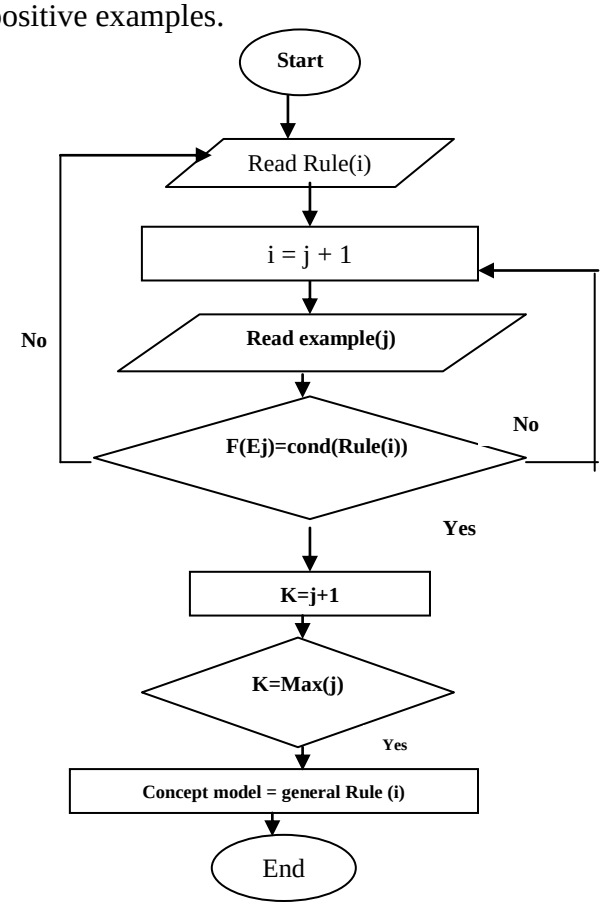

Fig. (13) :Concept model construction.

Inductive Logic Programming

By this technique we aim to construct concepts of the entities found in the documents by specifying attributes generated to specific entity progressing in forward manner from specific to general as shown in Figs. (1, 2, 3, and (4). Fig. (14) describes this technique.

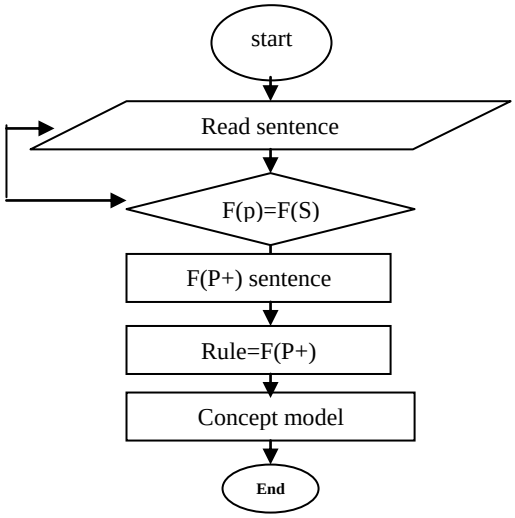

Fig. (14) : Inductive Logic programming.

The standard approach is to use a form of greedy set covering, where at each iteration; a new rule is learned that attempts to cover the largest set of examples of a particular category without covering examples of other categories. These examples are then removed, and additional rules are learned to cover the remaining examples of the category.

Constructing concept model for the desired domain will be useful in the future when we consult with another text for the same domain.

Interacting with the user the system can extract the facts and rules, and relations between entities.

Evaluation*:*

One obvious method to evaluate a computational theory would be to run the program to see how well it performs. For example if the program is meant to answer questions about a database of facts, you might ask questions to see how well it is at producing the correct answers.

We can evaluate the results with the following metric functions:

Entropy (impurity, disorder) of a set of examples, relative to a binary classification is:

$$\underline{\text{Entropy (S)}} = -p_+ \log_2 (p_+) - p_- \log_2 (p_-) \quad \text{.....(6)}$$

where P_+ is the proportion of positive examples in (S) and P_- is the proportion of negatives.

For multiple category problems with C categories, entropy can be generalized to:

$$\underline{\textbf{\textit{Entropy (S)}}} = \sum_{I=1}^{C} - P_i \log_2 (P_i) \quad \text{.....} (7)$$

where $\boldsymbol{P_i}$ is the proportion of category i examples in S.

The information gain of an attribute is the expected as reduction in entropy caused by portioning on this attribute:

$$\underline{\text{Gain (S, A)}} = \text{Entropy (S)} - \sum_{v \in \text{Values of A}} \frac{|Sv|}{|S|} \text{Entropy}(S_v) \quad \text{.....} (8)$$

Where $\underline{\boldsymbol{S_v}}$ is the subset of $\boldsymbol{S}$ for which attribute $\boldsymbol{A}$ has value $\boldsymbol{v}$ and the entropy of the partitioned data is calculated by weighting the entropy of each partition by its size relative to the original set.

Performance

Is measured in terms of recall and precision, where recall is the percentage of positive instances that were formed by rule base.

Precision measures the percent correct of instances extracted by the rule base [23].

$$\text{Recall} = \frac{\text{Number of correctly predicted entities}}{\text{Number of entities that should have been found}} \quad \text{.....} (9)$$

$$\text{Precision} = \frac{\text{Number of correctly predicted entities}}{\text{Number of all entities predicted}} \quad \text{.....} (10)$$

We applied and test our proposed system by training it to different texts types, novel, and newspaper articles. One of this is the following biological kingdom text:

{"A predator is an animal that eats other animals. For example, lions eat gazelles and zebras. These are the preys; Humans are predators that can be preys too.

Vegetarian animals are usually preys, while predators are obviously carnivores.

Humans are carnivores, because they eat both animals and vegetables}.

Another difference between these two groups is that predators are small and fast, while the preys are bigger and slower."}

First of all, the text is segmented according to discourse analysis c the Word-Net or System lexicon is classified generally according to words noun and verbs syntactic and semantic features:-

- Verb(stem,voice,tense[subjectgender,object gender],number).
- Noun (stem, definition, gender, number, adjectivability).
- Pronoun (gender, number).
- Preposition.
- Adjective.
- Aux verb.
- Question mark operators**.**

Looking to previous properties the proposed system tags text words according to such classifications. This step is followed by morphological analysis through stored domain morphological pipeline (suffixes, prefixes, and define roots of noun and verbs).

According to augmented grammar the proposed system constructs rules represented by logical forms to analyze and parse the previous sentences syntactically. The retrieved data are represented by a set of patterns as Noun, Verb phrases, adjective Subjects, and Objects.

$\underline{NP_1+VP_1+NP_2}$: a predator + is + an animal

$\underline{NP_3+VP_2+NP_4}$: that (refer to predator) eats other animals.

Adjective: animal → predator.

$\underline{NP_5+VP_3 + NP_6+\text{conjuction} +NP_7}$: Lions eats gazelles and zebras.

In the processed text, we have two simple anaphora sentences, the system resolves such sentences such that in the sentence:

"A predator is an animal that eats other animals."

"That" is relative clause, which refers to the noun {predator}.

And in the sentence:

"Lions eat gazelles and zebras. These are the preys".

These refer to preys.

Proceeding with implemented design to process the text, the system analyzes combining syntactic and semantic features to define its semantic representation controlled by ATN techniques.

A sentence like "a predator eats the prey" can be processed by this technique:
"Predator is a living – entity type animate. ***Eats*** is present transitive verb-type patient – subclass stomach. So the prey is living entity also animate."
"A predator eats the prey."
Rule: s-mod (Declaration),
First-NP NP1,
Det(the),
Noun(type:animate,sub_type(animal),gender(predator),
Verb:eats;tens-present;
Type-transitive,
Sem-ingest;stomach
Agent(a predator),
Ref-number(singular),
Theme-eating,
Object:NP2,
Patient- animate,living_entity,type-animal,sub_type:food,
Sem_type(edible),
Obj(the prey).

Derived sentences, patterns, and property represent a positive example in terms of inductive logic programming (ILP), which is the proposed technique in learning by examples. Sample of the extracted concepts and facts (relations between concepts) is:

isa(predator, animal).
isa(prey, animal).
eats(predator, prey).
can_be(predator, human).
property(predator, bigger).
property(prey,slower).
eat(lions, gazelles).

Machine learns when the system begins extracting a schema and a hall picture of the desired domain, interacting with the user to extract the facts and rules, and relations between entities. So in order to define the missing relations between concepts and define hierarchy concepts map, the proposed system constructs structured query with the user to complete such facts and relations.

1. Define the concept animal?
2. Complete the relation between predators and lions?
3. Can human be predator?

The system interface module declares these relations, facts and interacts with the user to construct concept module. Defining rules according to these relations and facts (conditions) is the first schema in knowledge base of the machine. The expansion of the knowledge base will be:

isa(predator, animal).
isa(prey, animal).
eats(predator, prey).
can_be(predator, human).
property(predator, bigger).
property(prey,slower).
eat(lions, gazelles).
Can_be(predator,prey).
Can_be(lion,prey).
eat(predator,vegetable).
Can_be(human,prey).

Input (Reading) another text of the same domain will feed and expand machine knowledge base through machine learning from a new knowledge which is beginning from scratch. From the first phase of machine learning to proposed testing text, we can draw the semantic network of partial biological kingdom as follows:

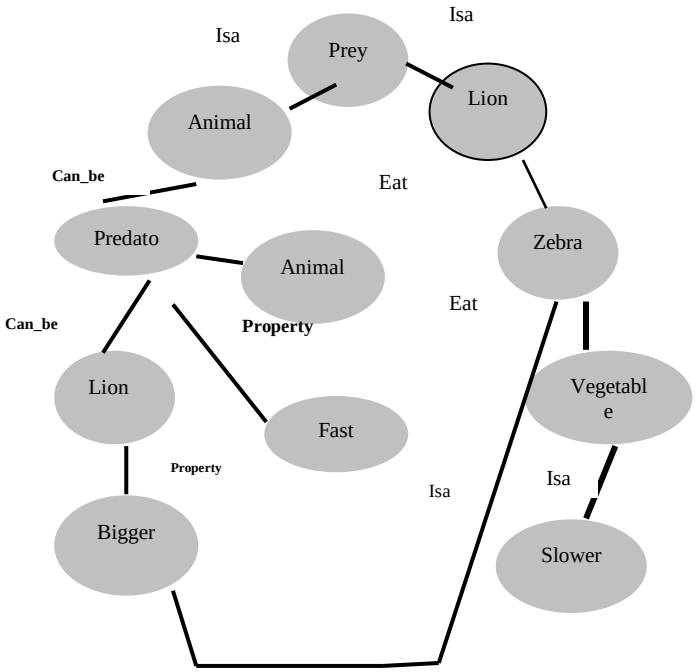

Fig. (15) : Semantic Network.

For the second model (Topic spotting), the proposed system as shown in Fig. (16), builds a classifier for a set of documents through a preclassified documents, such that some of them consist of the same and /or different domains respectively.

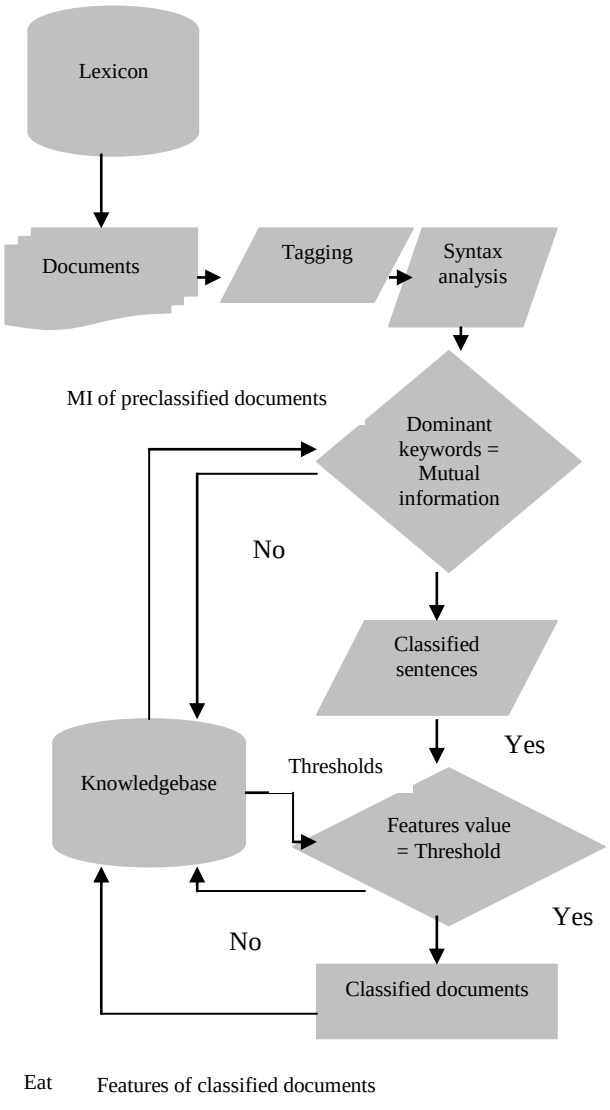

Fig. (16) : Topic spotting with ML approach.

The proposed second model aims to construct a classifier from a classified documents belonging to different domains, through studying the characteristics, terms weights, dominant words, mutual information, and sentence classifications. Learning the machine with the newly documents feature would feed the knowledge base of the proposed system classifier. Section (2.6) describes the first transducer of this model, tagging each word in the document with supported system lexicon, deriving words root to recognize hyponyms, synonyms. Segment sentences then sentence analysis is the second transducer in the proposed system model, as

Depending on mutual information (MI), the system would find successful dominant frequent terms which characterize each Topic, such that machine could learn the properties of the last depending on its ability to understand text features. Fig. (17) describes the flowchart of this technique.

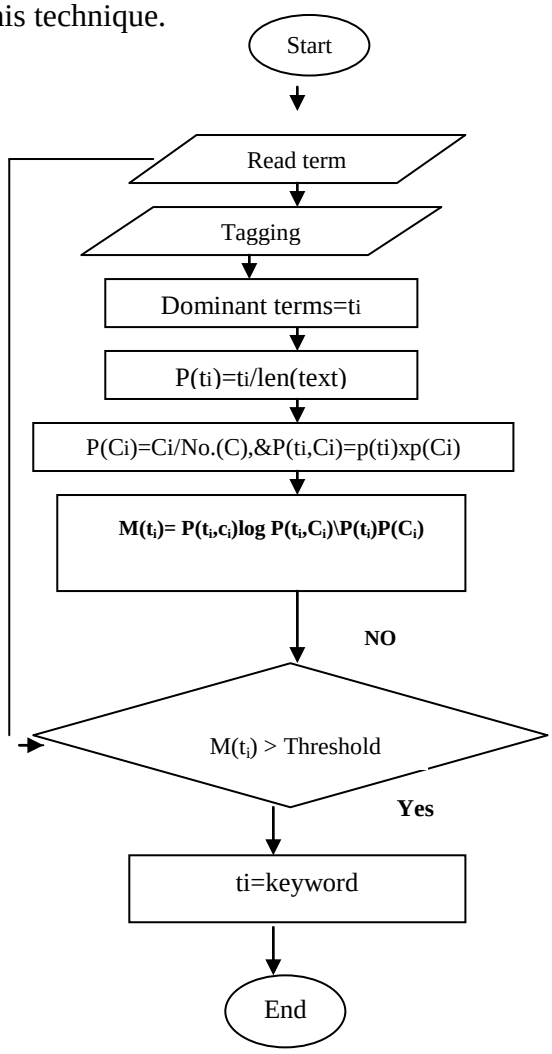

Fig. (17): Mutual information.

The proposed system derives weight of more frequent terms document, as shown in Fig. (18) which are labeled sequentially and stored in machine knowledge base as features of document ($F(d_j)$).

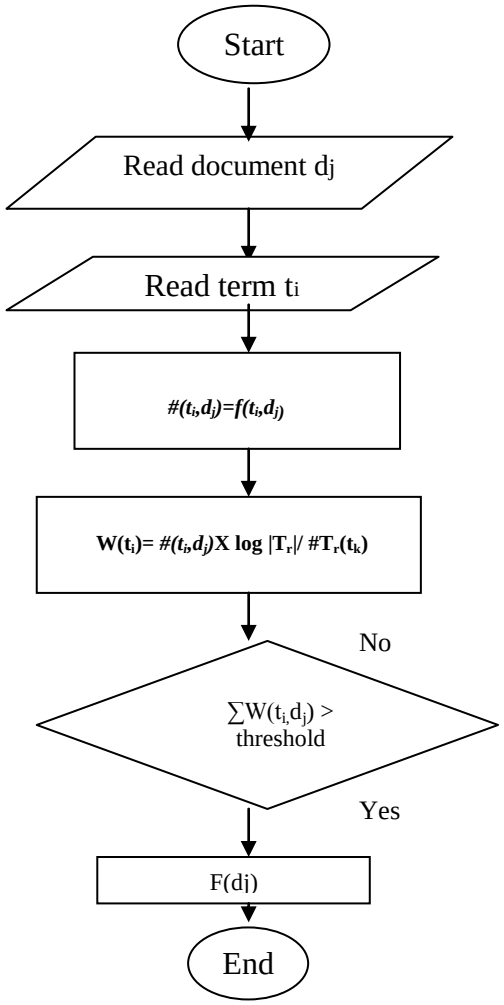

Fig. (18) : Determining weights flow chart.

Table (1)
Some of dominant terms in training documents.

Documents	T1	T2	T3	T4
D1	Biological	Kingdom	Animal	Cnidarians
D2	Football	FIFA	World cups	Ranking
D3	Games	Basketball	Handball	Swimming
D4	Majesty	Government	Prime minister	King
D5	Racing	Team	Games	Goals
D6	President	Government	Congress	_____
D7	President	Government	Congress	_____
D8	European Unions	Presidency	Political	_____
D9	Democracy	Bush administration	Qaeda	_____
D10	Market	Financial		_____
D11	Gulf Arab	United Nations	Gulf Arab Leaders	_____
D12	Government	Election	Weapons	_____

Supervised learning technique is adapted by system aiming for user decision in defining measurements criteria or thresholds features system that assist to the probabilistic of spotting document topic.

The system ranks categories in D with the user assistance according to their appropriateness to C, such that the proposed system would define a threshold τ_i value in this manner due to the flexibility of terms estimation in more than one Topic. For a large set of documents, the proposed system would define generality $g_\Omega(c_i)$ of a category C in terms of test and training sets.Articles in newspapers { THE JORDAN TIMES, Herald Tribune, and The Daily Star } are a set of news articles which been applied to proposed system techniques as training and test sets documents for machine learning of Topic Spotting .

To discuss the results of these techniques, some of them are shown in Table (1), Table (2), and Table (3). The documents are classified to training and testing sets, such that testing set is 3/2 of training set.

Table (2)
Weights and Mutual Information of Some Training Documents Term.

Documents	t_k	d_j	$T_{r(t_k)}$	W	M	$\sum M$
D_1	3	1	2	2.3344	0.0444	0.1332
D_2	2	2	2	1.5563	0.2949	0.8847
D_3	3	2	2	2.3344	0.04779	0.14337
D_4	4	1	1	4.3167	0.10934	0.32802
D_5	2	7	2	1.5563	0.65988	1.97964
D_6	2	7	2	1.5563	0.04340	0.1302
D_7	1	7	7	0.7781	0.02743	0.08229
D_8	2	7	7	2.1583	0.80485	2.41455
D_9	2	7	7	1.5563	0.03734	0.11202
D_{10}	2	1	1	2.1583	0.80485	2.41455
D_{11}	3	1	7	3.2357	0.03883	0.11649
D_{12}	2	1	7	2.1583	0.03883	0.11649

Table (3)
Weights and Mutual Information of Some Test Documents Term.

Documents	t_k	d_j	$T_{r(t_k)}$	W	M	$\sum M$
D_1	2	2	2	0.2498	0.0151	0.0454
D_2	2	2	2	0.6020	0.0093	0.0280
D_3	3	1	1	1.8061	0.3615	1.0487
D_4	3	1	1	0	0.4769	0.4769
D_4	2	2	1	0.6020	0.0132	0.0398

Table (4)
Evaluation of system classifier.

Category C_i		User Decision	
		True	False
System classifier Decision	True	T_{ci} +	F_{ci}+
	False	F_{ci} -	T_{ci} -

Table (5)
Evaluation of Training documents

Category	P_i (precision)	R_i (Recall)
Biological	0.5	1
Political	0.8571	1
Sports	1	0.75
Business	1	1

Table (6)
Evaluation of Test documents

Category	P_i *(precision)*	R_i *(Recall)*
Biological	*1*	*1*
Political	*1*	*1*
Sports	*1*	*0.5*
Business	*1*	*1*

Table (7)
Categorization of Training documents.

Documents	Effectiveness	Domain
D_1	0.6666	Biological
D_2	0.875	Sports
D_3	0.875	Sports
D_4	0.8888	Political
D_5	0.875	Sports
D_6	0.8888	Political
D_7	0.8888	Political
D_8	0.8888	Political
D_9	0.8888	Political
D_{10}	1	Business
D_{11}	0.8888	Political
D_{12}	0.8888	Political

Table (8)
Categorization of Training documents.

Documents	Effectiveness	Domain
D_1	*1*	*Political*
D_2	*1*	*Political*
D_3	*1*	*Business*
D_4	*0.6666*	*Sports*
D_5	*1*	*Political*

Conclusions

The proposed system have provided a survey on symbolic machine learning for understanding text written in English natural language .The main conclusions that the proposed approach can use supervised learning approach to build a machine system which can learn from written English text.

In spite of the complexity of English language the proposed system offer intelligent user interactive interface that create structured query and complete the concepts relations before extracting the desired information from one or a lot of documents.

References

[1] Oliveiria,"Automatic Reading and learning from text".

[2] Nilson j. Nilson "introduction to machine learning", Robotics Laboratory, department of computer science, Stanford University, 1996.

[3] Cynthia A. Thompson," Acquiring Word-Meaning Mappings for Natural Language Interfaces", School of Computing, University of Utah.

[4] Allen, James," Natural language understanding", university of Rochester, 1995.

[5] Siter, an De,"Information extraction via double classification" university of Antwerb, Belgium, 2000.

[6] Luger, George," Artificial Intelligence", 1998.

[7] Mooney, Raymond," Symbolic machine learning for natural language processing." Cornell University, 2003.

[8] Mooney, Raymond," Learning semantic parsers", 2003.

[9] Cardie, A case –based approach to knowledge acquisition for domain specific sentence analysis", university of Massachusetts, 2003.

[10] Hobbs, jerry r.,"a cascaded finite-state transducer for extracting information from natural language text", Menlo Park California.

[11] Soderland, Stephen G,"Learning text analysis rule for domain specific natural language processing", university of massachuatesettes 1997.

[12] Rich, E, "Artificial intelligence ", Mc GRAW Hill, tnc, 1991.
[13] Shapiro," Encyclopedia of Artificial intelligence, volume 1, 1987.
[14] Kufman, William," THE HANDBOOK OF ARTIFICIAL INTELLIGENCE ", Volume 1, 1981.
[15] Michalski, r., "Understanding Natural Language ", Encyclopedia of artificial intelligence, SHAPIRO, S. (ED), VOL1, JOHN WILLY & SONS, New York 1990.
[16] Grankist, Danial,"Natural Language Understanding and Dialogues by Partial Adaptive term Spotting. the Isaac Museum Guide Robot, internet research.
[17] Traum," INTEGRATING NATURAL LANGUAGE UNDERSTANDING AND PALN REASONING IN THE TRAINS"1993, Conversation system, Internet research.
[18] Aitken, James," Learning Information Extraction Rules: An Inductive Logic Programming, Internet Researches.
[19] Soderland, Stephen, "Learning Domain Specific discourse rules for information extraction", Internet research.

الخلاصة

يتناول هذا البحث دراسة أحد حقول الذكاء الاصطناعي من خلال دراسة كيفية تعلم الماكينة لفهمها للغة الطبيعية. إن أهم جزء في عملية فهم أي نص يأتي من خلال أدراك المفاهيم الموجودة داخل ذلك النص وفهم العلاقات فيما بين تلك المفاهيم. إن التعلم هواكتساب المعرفة أو فهمها أو اكتساب المهارات من خلال دراسة الأيعازات و المهارات أو الخبرة وتغيير السلوك من خلال اكتساب تلك الخبرة. و بشكل عام يمكن القول بأن الماكينة قد تعلمت،عندما يتغير تركيب أو مسار البرامج متى ما تغيرت مدخلاتها أو من خلال الاستجابة لتغير البيانات الداخلة أليها من المصدر الخارجي وبحيث يمكن توقع تغير من كفاءتها مستقبلا". يهدف النظام المقترح إلى بناء نموذجين : الأول يسعى الى استخلاص المفاهيم من نص مكتوب باللغة الإنكليزية والثاني : يهدف الى بناء مصنف لتصنيف مجموعة من الوثائق أو النصوص.وعلى الرغم من صعوبة اللغة الطبيعية الإنكليزية فأن النظام المقترح يوفر واجهة تفاعلية ذكية مع المستفيد تمكنه من خلق أسئلة واستفسارات مهيكلة يسهل له الإجابة عنها واستخراج المفاهيم الناقصة والعلاقات التي تربط فيما بينها استكمالا" لاستخلاص المعلومات من النص المراد فهمه .