Extraction of Syntactically Similar Sentences from Huge Corpus for Language Research

Sanjay Kumar [1], Sandhya Umrao[2]
School of Computing Science & Engineering, Galgotias University, India
Galgotia College of Engineering and Technology, Greater Noida, India,

Abstract

The Corpus Based and statistical approaches exploits several heuristics to determine the summary-worthiness of sentences. It actually uses statistical appearances of words, words-pairs and noun phrases to calculate sentence weights and then extract the highest scoring sentences. The purpose of this research is to build a tool for Extraction of Syntactically similar sentences from huge corpus for language research. To discuss its design, use and implementation. The proposed tool is based on a logical approach to computational corpus linguistics where sentences of logic are used to express statements about texts and logical inference is used to manipulate these sentences in order to analyze the texts.

The research based on functionalities needed in a corpus system can be implemented when based upon adequate means of representing, querying and reasoning. The proposed system implements hand coding, searching and parsing.

Apart from being interesting from a practical point of view, the development of such a system raises intriguing philosophical and methodological questions: What is corpus texts? What is a corpus theory? What is the link between the truth of such a tool and its usefulness for natural language processing purposes? These and related questions are discussed in the research. The system exist in a prototype implementation and the research contains numerous examples from this implementation in action.

Key Words: *Corpus Linguistics, Corpus tools, Grammar, Grammar development, Logic programming.*

I. INTRODUCTION

The body of written a particular subject is an important way of basic data for language research. While numbers of syntactically syntax analysis tools exist such as software of alphabetical list of word in the text, but there is also need of more observation in create well-formed sentences in a language? MT software may need some more specific syntax criteria for particular rules for developed system. There are numerous algorithms and tools are available for statistical analysis of the on-line text but

- Researchers are increasingly interested in syntactic investigations of larger corpora.
- Machine translation software developers need syntactically grouped corpus for rule formation.
- MT software evaluators require verity of sentences of different syntactic patterns to evaluate the grammatical coverage.
- Language learners may require the different examples of same structure.

Keeping the above in view the research for automatic extraction of syntactically similar sentences from huge corpus is chosen.

The corpora are an important source of empirical data for language research. While numerous syntactic analysis tools exist such as concordance software but there is also a need for syntactic investigations. MT software may require the sentences of some specific syntactic criteria to create the rule or to evaluate a developed system. A language learner needs various examples of same syntactic pattern. Therefore, the objective is to design an algorithm for the selection of sentences by giving syntactic criteria or the grouping of syntactically similar sentences from text corpora when these corpora contain no prior syntactic markup.

II. NATURAL LANGUAGE PROCESSING

The process of computer analysis of input provided in a human language (Natural Language), and conversion of this input into a useful form of representation. The field of NLP is primarily concerned with getting computers to perform useful and interesting tasks with human language. The field of NLP is secondarily concerned with helping us come to better understanding of human language. To build a program that understands spoken language, we need all the facility of a written language understand as well as enough additional knowledge to handle all the noise and ambiguity of the audio signal. Thus it is useful to divide the entire language – processing problem in to two tasks.

- ✓ Written Text
- ✓ Speech

- Processing written text, using lexical, syntactic knowledge of the language as well as the required real world information.
- Processing spoken language, using all the information needed above plus additional knowledge about phonology as well as enough added and information to handle the further ambiguities that arise in speech.

A. Information Extraction

There are many NLP systems that can process arbitrary text. Several general –purpose linguistic capabilities are characteristic for this type of systems: part-of-speech tagging, parsing, word-sense disambiguation, higher level (semantic) understanding, dialog systems, natural language interfaces and queries.etc. We are interest in NLP systems that are built with a pre-specified task over a well-defined domain of interest. These systems are information extraction systems.

B. Generic IE System

In the generic IE system, there are no assumptions on the input format. The Tokenization and Tagging phase tokenizes the txt- divides it into sentences with words. Some systems can eventually disambiguate or tag for parts of speech (POS) or semantic class. The sentence analysis stage parses the sentences for simple constructs (verb, noun, prepositional and other phrases). This stage could also involve parsing for higher level constructs and even label semantic entities in the text and transform them to normalized form. Up to this point, the two stages of the system are not necessary domain-specific.

Extraction is the first stage where the processing is tied to the domain. Here relevant parts are extracted, but it is important to point out that this stage is not just extraction of text fragments. In addition, this stage annotates relations between parts of the text, and additional information that describes the triggers of the extraction. The merging stage compares the entities extracted in the previous stage and deduces whether they refer to the same information. This comparison filters the extracted information and in the place where separate extraction results are combined into one when they refer to the same piece of information. The final, template generation stage produces the output. This stage considers only extracted information relevant to the output format.

C. Components of NLP

1. Natural Language Understanding

Mapping the given input in the natural language into a useful representation different level of analysis required.

1. Morphological Analysis
2. Syntactic Analysis
3. Semantic Analysis
4. Discourse Analysis

2. Natural Language Generation

Producing output in the natural language from some internal representation different level of synthesis required.

1. Deep Planning
2. Syntactic Generation.

D. Knowledge of Language

1. **Phonology:** Concern how words are related to the sounds that realize them.
2. **Morphology:** Concerns how words are constructed from more basis meaning units called morphemes. A morpheme is the primitive unit of meaning in a language.
3. **Syntax:** Concerns how can be put together to from correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of other phrases.
4. **Semantics:** Concerns what words mean and how these meaning combine in sentences to form sentence meaning. The study of context-independent meaning.
5. **Pragmatics:** Concerns how sentences are used in different situations and how use affects the interpretation of the sentence.
6. **Discourse:** Concerns how the immediately preceding sentence affects the interpretation of the next sentence.
7. **World Knowledge**: Includes general knowledge about the world. What each language user must know about the others beliefs and goals.

E. Semantic Analysis

Semantic analysis must do following important things.

- It must map individual words into appropriate objects in the knowledge base or database.
- It must create the correct structures to correspond to the way the meaning of the individual words combine with each other.
- Assigning meaning to the structures created by syntactic analysis.
- Mapping words and structures to particular domain objects in way consistent with our knowledge of the world.
- Semantic can play an important role selecting among competing syntactic analysis and discarding illogical analysis.

F. Syntactic Analysis

Syntactic analysis must exploit the result of morphological analysis to build a structural description of the sentence. The goal of this process, called parsing, is to convert the flat list of words that forms the sentences in to a structure that defines the units that are represented by the flat list.

1. **Parsing:** Converting a flat input sentence into a hierarchical structure that corresponds to the

units of meaning in the sentence. There are different parsing formalisms and algorithms. Most formalisms have two main components:

- **Grammar:** A declarative representation describing the syntactic structure of sentences in the language.
- **Parser:** An algorithm that analyzes the input and output its structural representation consistent with the grammar specification.

2. **Context Free Grammar:** CFG are in the center of many of the parsing mechanisms, but they are complemented by some additional features that make the formalism more suitable to handle natural languages.

- **Rewrite Rules:** Specify what tree structures are allowable.
- A grammar whose rewrite rules have a single symbol.
- Powerful enough to describe most of the structure in natural language.
- Restricted enough to develop efficient parsers.

III. RELATED WORK

A. A Syntactic Approach for Searching Similarities within Sentences

Textual data is the main electronic form of knowledge representation, sentences meant as logic units of meaningful word sequences, can be considered its backbone. This method based on purely syntactic approach for searching similarities within sentences. This process being very time consuming, efficiency in retrieving the most similar parts available in large repositories of textual data is ensured by making use of new filtering techniques.

B. Finding Syntactic Structure in Unparsed Corpus

The Gsearch Corpus Query System: The Gsearch system allows the selection of sentences by syntactic criteria from text corpora, even when these corpora contain no prior syntactic markup. This is achieved by means of chart parser, which takes as input a grammar and a search expression specified by the user. Gsearch features a modular architecture that can be extended straight forwardly to give access to new corpora. The Gsearch architecture also allows interfacing with external linguistic resources. Gsearch system is a tool designed to facilitate the investigation of lexical and syntactic phenomena in unparsed corpora. Gsearch permits users to search for linguistic structures by processing a query based on a user definable grammar. Gsearch is intended as a flexible tool for scientists wishing to study corpora, and is not intended for accurate unsupervised parsing. The nature of lexical and syntactic ambiguity means that Gsearch will often return a parse, which while strictly correct with respect to the supplied grammar and query is inappropriate for a sub string. The current version of Gsearch only supports lexical markup in the input corpus. Gsearch relies on a Uniform Input Format (UIF) for the corpora it accesses. For each corpus supported by Gsearch, a filter is required that translates the specific format the corpus is encoded into the UIF. Gsearch also supports a pipeline architecture where the input is taken from the output of a tagger running on a previously untagged corpus. The output of Gsearch query is encoded in a Uniform Output Format (UOF), which is passed through an output filter that transforms it into the

Specific format required for a given output module. Gsearch comes with filters for a number of corpora that are widely used.

C. Annotation of Syntax and Morphology:

For an annotation of grammatical categories to word from tokens in our corpus, the commercial tagger machines syntax by connexor was employed. This tagger is a rule –based, robust syntactic parser available for several languages and based on Constraint Grammar and Functional Dependency Grammar. It provides morphological, surface syntactic and functional tags for each word form and a dependency structure for sentences and besides is able to process and output

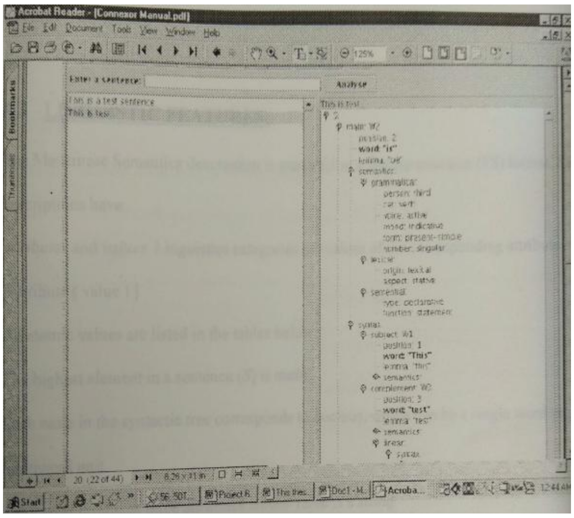
IV. PROPOSED WORK

Machinese Semantics contains a custom lexicon mechanism. This feature enables developers to add their own words to the parser. These words can be domain-specific vocabularies, multi-word terms, names and places etc. this way developer can influence how the parser analyses texts.

Machinese linguistic analyzers support the following output formats.

1. XML output
2. Visual feature tree output- Machinese Semantics Viewer

Machinese Semantics XML output form and Machinese Semantics Viewer shows the same linguistic information, but the formatting is obviously different.



This is a test sentence
This is test

```
This is test
S
  main: W2
    position: 2
    word: "is"
    lemma: "be"
    semantics
      grammatical
        person: third
        cat: verb
        voice: active
        mood: indicative
        form: present-simple
        number: singular
      lexical
        origin: lexical
        aspect: stative
      sentential
        type: declarative
        function: statement
    syntax
      subject: W1
        position: 1
        word: "This"
        lemma: "this"
        semantics
      complement: W3
        position: 3
        word: "test"
        lemma: "test"
        semantics
        linear
          syntax
```

Table 4.1: Machinese Semantics Features

A. Linguistic Features

The Machinese Semantics description is provided in a Feature-Structure (FS) format. FS descriptions have *attributes and value*. Linguistics categories are values of the corresponding attribute. The highest element in a sentence is **main.** Each node in the syntactic tree corresponds to nucleus, which can be single word or a multiword unit. The parser produces three attributes whose value is a string.

- **Word** is the running text token.
- **Lemma** is the base form of the nucleus.
- **Head** is printed only in such multi-word units where it is different from the lemma.

B. Main Level Index

Sentences divided in two parts.

1. Syntax

There are different categories of the sentences at syntax level.

Category	**Explanation**
Main	Main nucleus of the sentence
Subject	Notional subject
Theme	Grammatical subject in existential construction and extra position
Object	Notional direct object
Complement	Complement of the notional subject
Attribute	Attribute
Coordination	Topology
Determiner	Articles
Quantifier	Quantifying determining
Reason	Adjunct for reason
Purpose	Adjunct for purpose

Table 4.2: List of different category of Sentences at syntax level.

2. Semantics

Semantics are further categorized into three levels.

2.1. Grammatical Semantics

Grammatical sentences can be further divided into the following categories:

Category	**Tag**
Noun	N
Adjective	A
Verb	V
Numeral	NUM
Determiner	DET
Adverb	ADV
Coordinator	CC
Pronoun	PRON
Interjection	INTERJ
Preposition	PREP
Subordinator	CS

Table 4.3: Main Part-of-Speech category

Grammatical (Morphological) Number

Category	Tag
Singular	SG
Plural	PL

Table 4.4: List of Numbers

Grammatical Case

Category	Tag
Nominative	NOM
Accusative	ACC
Gentive	GEN

Table 4.5: List of Grammatical Case

Grammatical Person

Category	Tag
First	SG1.PL1
Second	SG2,PL2
Third	SG3,PL3

Table 4.6: List of Grammatical Person

Grammatical Gender

Category	Tag
Male	MASC
Female	FEM

Table 4.7: List of Grammatical Gender

Grammatical Voice

Category	Tag
Active	%VA
Passive	%VP

Table 4.8: List of Grammatical Voice

Grammatical Mood

Category	Tag
Indicative	______________________
Subjunctive	SUBJUNCTIVE
Conditional	…………………………………
Imperative	IMP

Table 4.9: List of Grammatical Mood

Grammatical Degree

Category	Tag
Absolute	ABS
Comparative	CMP
Superlative	SUP

Table 4.10: List of Grammatical Degree

2.2. Sentential Semantics

It refers to the properties of clauses and sentences.

✓ Sentence Function
✓ Sentence Type
✓ Sentence Modality

2.3. Lexical Semantics

It refers to the features of a given lexeme.

✓ State
✓ Proper (Noun)
✓ Lexical Semantic Class
✓ Refining
✓ Location

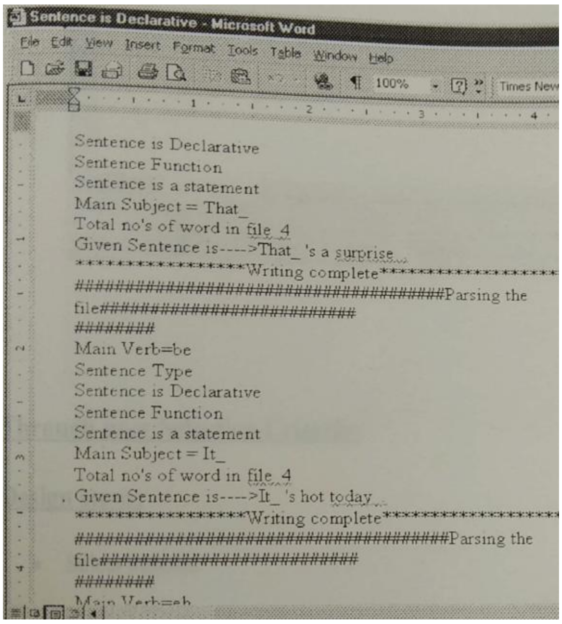
V. DESIGN AND EXPERIMENT RESULT

A. Types of Input

1. Through Text file extract the sentence

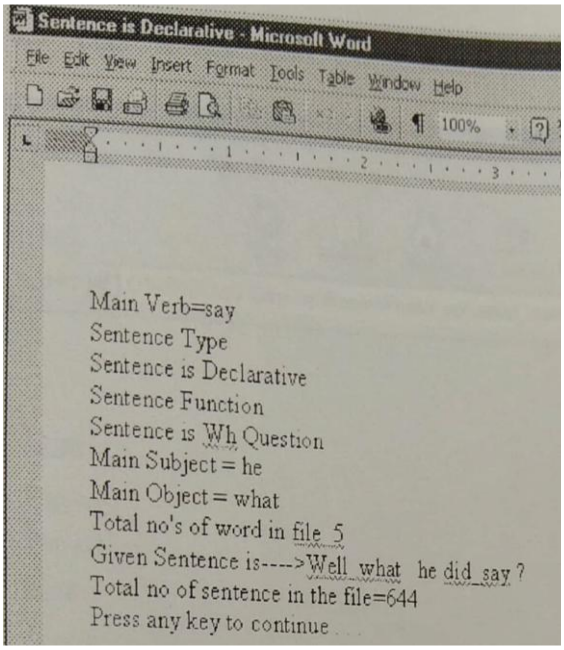
Design Step:

- Find the syntax name
- Find out each word id and position according to syntax name and store in the array. Find out the semantics (Grammatical, Sentential and Lexical) of the word and put in the database.
- Create a database of unique sentence
- Find out the total number of words in the sentence.


Sentence is Declarative - Microsoft Word
File Edit View Insert Format Tools Table Window Help
100%
Times New

```
Sentence is Declarative
Sentence Function
Sentence is a statement
Main Subject = That_
Total no's of word in file 4
Given Sentence is---->That_ 's a surprise
******************Writing complete*******************
##########################################Parsing the
file##############################
#########
Main Verb=be
Sentence Type
Sentence is Declarative
Sentence Function
Sentence is a statement
Main Subject = It_
Total no's of word in file 4
Given Sentence is---->It_ 's hot today
******************Writing complete*******************
##########################################Parsing the
file##############################
#########
Main Verb=eh
```

Figure 5.1: Type of Sentence


Sentence is Declarative - Microsoft Word
File Edit View Insert Format Tools Table Window Help
100%

```
Main Verb=say
Sentence Type
Sentence is Declarative
Sentence Function
Sentence is Wh Question
Main Subject = he
Main Object = what
Total no's of word in file 5
Given Sentence is---->Well what he did say ?
Total no of sentence in the file=644
Press any key to continue
```

Figure 5.2: Total Number of Sentence in the File

2. Select the type of sentence by the user
Design Step:

- Select Indexes
- Select property within the selected index
- Generate the query for selected property
- Process the given query in the database
- Display the output.

V. CONCLUSION

We are able to find out the type of a particular sentence, what are main subject, main verb, type and function. In this implementation we can also find the total number of sentences in the particular file.

REFERENCES

[1] Stean Corley, Martin Corley, Frank Keller, Matthew W, “Finding Syntactic Structure in Unparsed Corpora” The Gsearch Corpus Query system.

[2] Aho,A.V. and J.D Ullman “The theory of parsing, translation and compiling” 1972

[3] Federica Mandreoli, Riccardo Martoglia “A Syntactic Approach for Searching Similarities within Sentences”

[4] Sinha, RMK “Machine translation an Indian perspective” in Language Engineering Conference 2002. Procedding,13-15Dec2002 Pages:181-182

[5] Jain, Sinha, RMK, “Role of examples in translation in systems, man, and cybernetics” 1995 , intelligent systems for 21st Century. IEEE international Conference on , Volume 2,22-25 Oct 1995, Pages: 1615-1620

[6] Vilares Ferro, M; Alonso Pardo,M; Grana Gil,J; Cabrero Souto,D “ Tabular DCG parsing for natural language” In proceeding of the First workshop on Tabulation in Parsing and Deduction(TAPD-1998) , Paris, PP 44-51