Natural Language Understanding Using Word Type Disambiguation and Semantic Networks

Pierre Innocent, Member, IEEE
Tsert.com
contact@tsert.com
http://www.tsert.com/

Abstract - Our approach [patent pending] to natural language understanding and content analysis of unstructured text in alphabet and ideogram based languages (e.g. latin, slavic, germanic, arabic, chinese, etc.) is anchored on the process of word-type disambiguation. The process itself, is based on the statistical analysis of source text written according to the normal usage of a language – how the language is used by native speakers, the same analysis must be done for jargon, and specialized domain languages such as legalese.

The statistical analysis is performed to extract probabilities of appearance of word types in a sequence of word tokens. Once the statistical analysis is performed, a rules set is created. The rules set is then used to improve the process of phrase structure analysis, content analysis, and translation of the unstructured source text.

The analysis, for ideogrammic languages, is done not only for sequences; but also for juxtaposition and combination of glyphs; or glyph sets if a language is known. All languages are glyph based.

Semantic networks (knowledge bases) and Natural Language Processing (NLP) based heuristics are used to weigh the word tokens, that were extracted from the source text, in order to build a network of semantically linked words giving the user some notion about the content of the text.

Our engine is, in summary, a linguistic engine based on probabilities.

The relevance or our approach in building Web Search Engines is also discussed; as well as its possible application to deciphering languages.

The word content is used to mean the concepts and notions extracted from unstructured text. The words meaning, semantic information, and content are interchangeable in this paper.

Index Terms – Semantic Network, Phrase Structure, PS, Content Analysis, Natural Language Processing. NLP, Formal Gramar.

I. Introduction

THE statistics-based approach [***CETE Algorithm & Process***, ***patent pending***] described in this document is used to extract content from unstructured text.

Our primary goal was to improve accuracy where information retrieval, translation, and content analysis is concerned. Our secondary goal was to base the technology, used to perform the above mentioned tasks, on the same Natural Language Processing (NLP) engine.

Other approaches to performing these tasks rely on technology such as word clustering (Internet search-engines, IBM's [Web Fountain], and semantic lattices [NITLE]. Mathematics is the primary tool in these technologies. The NITLE approach, with semantic networks built using relationships between keywords based on co-occurence/clustering statistics, does not work as well as our approach.

Our uniform approach based on **word-type** disambiguation, semantic networks, and a natural language processing engine is believed to be more accurate -- based on our results[1]. Natural language understanding, not mathematics, is used to improve accuracy, by getting the NLP engine to understand text as a human reader, with semantic networks as knowledge bases.

II. Problems

A. Word Clustering

Word clustering is a technique for partitioning sets of words into subsets of semantically similar words (EAGLES Central Secretariat). It relies on statitistics referred to as distributional evidence of words within a particular segment of wors called cluster. Because word clustering is primarily based on mathematical constructs, using statistics and probabilities; its accuracy where informational retrieval is concerned cannot be as high as a system, which is built to extract content from the processed text.

The other limitation to word clustering is the definition of a cluster, as an arbitrarily sized segment or sequence of word tokens; whereas a person understands clusters and segments to simply mean either a sentence or a paragraph, which in themselves convey content o meaning. The way a writer ends a sentence or paragraph, and starts a new one contains semantic information or content. The way punctuation is used also conveys content.

B. Statistics Based Translation Systems

New approaches to translation, based on statistics, have been developed lately, which supposedly improve the resulting text. The systems, built using these approaches, are mainly *Copy&Paste* translation systems, where previously translated text is processed and indexed, and any segment/sequence of words that match the sequence in the text being processed, are extracted from the previously translated target text, amd placed in the resulting translated text. There are several limitations to the approach used in these systems:

1. As previously mentioned, punctuation conveys content; the way, it is used, may change the actual sense of a sentence. This is not taken into account in these systems.
2. The segment, to be translated, is not found in the previously translated text database; therefore results, if not using our approach will be the same as in old tranlation systems.
3. Structural ambiguities, in a particular sequence of words, may not be resolved through old approaches to phrase structure analysis. The statistics used in these systems do not resolve these ambiguities -- see Appendix-C.

III. Solution

A. Basics

We based our approach on the recognition of the basic ambiguities, which are at the root of all languages, both in structure and meaning. Structure, as in phrase structure which is related to grammar and syntax [3]; and meaning, as in word-sense and in understanding content where knowledge bases (semantic networks) need to be accessed.

Statistics, dealing with word type distribution were extracted from randomly downloaded text; and a rules set, based on these statistics, was created. The rules set was then used to parse text, written in the same language, using normal phrase structure analysis. Every subsequent step, such as translation and content extraction is based on the disambiguation of word types and its use in phrase structure analysis **[*patent pending*]**.

[1]See http://www.tsert.com/content-analysis.htm

B. Word-Type Disambiguation

When it comes to phrase structure, word sense disambiguation [Bri95] is what most systems rely on, when processing text. These systems use word clustering as their primary tool. Word type disambiguation which is the typing of the word itself, before any attempt at word-sense disambiguation is made, is what our system relies on, for understanding and extracting content from the source text **[** ***patent pending*** **]**.

Statistics of word type distribution, that is the probability of appearance of a word type in juxtaposition to other word types, is extracted. The statistics that are used, are statistics regarding the distribution of one word type in juxtaposition with one forward and one backward word type; it is a 3-word window of cluster. Accuracy could be improved by increasing the cluster size to 5; but the complexity of the statistical analysis and of the creation of the rules set would be increased exponentially.

The ***CETE*** algorithm is based on the understanding of language, as seen by the computer science field of formal grammars. The word types themselves are the usual grammar based word types such as **nouns**, **verbs**, **adjectives**, **adverbs**, **prepositions**, etc.. For improved accuracy, punctuation is also treated as a word type in our approach – see **Appendix-C**.

C. Rules Set

The rules set, which is created using the extracted statistics, is biased towards the way a human reader (native speaker of the language) would understand a particular segment or section of a sentence.

The statistics themselves may say that the probabilities that a word, following the one under examination, may be a **noun** or a **verb** are equal; our system therefore needs ways to weigh one choice, more than another. Following are some of these heuristics **[** ***patent pending*** **]**:

1. Biasing the system as mentioned above.
2. Accessing a semantic network or knowledge base for additional information about the word being examined.
3. Examining the semantic context in which the word is positioned; that is, first the sentence, and then the paragraph, page, chapter, book or document, which are the only clusters of any sgnificance to our system.

See **Appendix-A** for an example of a rule.

The rules set is used to build a finite state machine. Sequences of words are then fed to the machine. The result is a sequence of typed words that is then analyized with regards to phrase structure. A given sentence is seen as a series of typed words. Some sequences of typed words, when their word-sence is added, are easily recognizable; such as proverbs and colloquialisms. Such sequences are kept in easily accessible databases. Such sequences retain the same semantic information, even when their word-sense is replaced by similar ones. An example, is the sequence '***elementary, my dear watson***', see **Appendix-D**.

An ***evolutionary*** approach will be taken to generate a rules set for a given language. The evolutionary technique will be made possible by using the difference algorithm, provided by our translation engine. The process is straightforward; it consists in feeding language syntax, grammar, and dictionaries to our engine; which after incorporating them; starts translating an english language template into the target language and back to english. The amount of differences between the template and the re-translated version is measured to evaluate the accuracy of the initial rules set **[** ***patent pending*** **]**. The process is repeated until the level of similarity reaches 95%.. After processing a huge amount of text; the rules set is pruned, by removing all the rules which were not exercised.

D. Phrase Structure Analysis

Phrase structure analysis deals with syntax and regular sentence parsing. A loose set of phrase structure rules are selected, so that badly written text – with regards to grammar, syntax and punctuation – can be processed. See **Appendix-B** for an example of a phrase structure rule.

The set of PS rules is used to build a PS tree, which is essentially a tree of clauses representing ***noun*** phrases, ***verb*** phrases, ***adverbial*** phrases, ***prepositional*** phrases, etc.. It is that tree, which is examined in order to extract the semantic information or content required to give the user an idea of the subject of the source text.

When a valid parse tree cannot be built – the system's PS parser could not reach a final state with the given set of PS rules – the ***word-type*** selection phase is performed a second time. A type re-selection of words which may have been left loosely typed. is perfomed. Other words which were ***strongly***[2] typed, such as, ***progressive*** verbs, may be re-typed forcedly as nouns (e.g. *being*, *carrying*, *bearing*, etc.).

A limitation to our method of word-typing and phrase structure analysis is when a verb may have a qualifier, that is out of the 3-word window or cluster -- this is especially true in the English language, where adverbs such as ***up***, ***down***, ***on***, ***off***, etc., may serve as qualifiers to verbs. We resolve this limitation, first, by keeping the

[2]A strongly typed word means that no other word-type can be selected.

qualifier (*adverb*) loosely typed; and while building the parse tree, by having the *qualifier* and the qualified *verb*, at the same level in the parse tree.

An advantage of our approach is the partial resolution of the understanding and translation of ambiguous sentences, sometime referred to as ***gobbledegook***; a problem that copy&paste systems do not resolve **--See Appendix-C D**.

E. NLP Heuristics

Heuristics based on phrase structure analysis and natural language understanding are used to generate mathematical weights (numbers) which are used to rank the extracted words.

The most simple heuristic is to simply distinguish a ***subject*** from an ***object*** in a ***sentence*** clause – which is ***elementary*** grammar, see **Appendix-B**. Additional heuristics, such as recognizing and weighing ***prepositional/conjunctive/adverbial*** subjects and objects, as well as ***noun*** and ***adjective*** linkage, and ***verb*** and ***adverb*** linkage, is also done.

The significance of ***prepositional subject*** and ***object*** is based on language understanding. For example, in a sentence, such as '*Our **Lady** of **Watsonville** is a foot-high image of the Virgin Mary seen in the bark of an oak tree in Watsonville, California.*'; the words L***ady*** and ***Watsonville*** have a semantic relation to each other; one, as the *subject* of the ***prepositional*** clause, and the other as its *object*. Language constructs which convey the concept of ***subject***, such as prepositions like ***of***, ***about***, etc. are significant; they may lead to the extraction of the subject of the source text under analysis [heuristics ***patent pending***].

These heuristics [***patent pending***] – ***subject*** and ***object***, ***verb*** and ***adverb***, ***prepositional, conjunctive, and adverbial*** subjects and objects – are specifically targeted towards the extraction of the particular concept/subject of a given piece of unstructured text. See http://www.tsert.com/content-analysis.htm for real-world examples.

Semantic content, extracted from just the structure of the parse tree – Phrase Structure (PS) Notion -- may not be enough to accurately point to the subject/concept of the source text. A knowledge base (***semantic network***), need to be accessed to modify the weights that were originally obtained. The ranking of the extracted words may then change according to the number and weight of the semantic links that the extracted words have to each other.

A simple example would be where the source text speaks of an engineer and his or her invention. The name of the engineer may be mentioned in every single sentence in the text, even though the subject of the page is the engineer's invention. With just a phrase structure analysis , the engineer's name, as a subject in many sentences, gets the highest weight [heuristics ***patent pending***]. But, by subsequently accessing a semantic network, the engineer's invention may be seen as having more links to other words in the source text, thereby increasing its weight [heuristsics ***patent pending***].

F. Semantic Networks

*Semantic relations were introduced in generative grammar during the mid-1960s and early 1970s ([Fil68], [Jac72], [Gru67]) as a way of classifying the arguments of natural language predicates into a closed set of participant types which were thought to have a special status in grammar(**Eagles Central Secretariat**).*

A semantic network is a collection of words that are linked together by examining the relations that they have with each other, and adding weights to these relations.

Artificial Intelligence (AI) Systems use sets of inference rules which are essentially a semantic network captured as a set of programmatic statements, usually written in LISP or PROLOG.

Our version of a semantic network is kept as a graph, consisting of vertices with weighted and typed links capturing a semantic notion. In our system, semantic networks are used to improve results of word-type and word-sense disambiguation.

Semantic notions can be classified in many forms depending on the area of endeavor, Natural Language Processing (NLP) and Translation being some of these areas. These classifications deal with semantic roles such as ***agent*** or ***actor***, ***patient***, ***theme, etc.,*** [Jac90], [Dow89], [San92b], [San93b]. The formalisms we selected are used in natural language understanding and translation.

G. Web Search Engine

When it comes to search engines, such as Web search engines; the use of our approach can ensure accurate results on practically every single query. By building a semantic network with relationships between keywords that are based in natural language, we avoid the statistics-based problems having to do with keywords co-occurence, as stated in the problems section.

H. CETE Search Engine [*patent pending*]

After building a semantic network with natural language relationships between keywords; our search engine will do the following:

Indexing

1. Index filename of documents.
2. Index keywords found in pages and documents.
3. Content analyze the unstructured text in the pages and documents, using our ***NLP*** approach.
4. Build signatures of every set of extracted keywords and their relationships. These signatures are called ***semantic signatures [patent pending]***.
5. Build signatures of the path traversed, by every set of extracted keywords, in the semantic network. These signatures are called ***network path signatures [patent pending]***.
6. Associate ***semantic*** and ***network path*** signatures with scanned pages and documents ***[patent pending]***.

Search Queries

1. Make searches using ***path-spec queries [patent pending]***.
2. Make searches using ***keywords*** only (i.e. clustering legacy way).
3. Make searches by traversing the semantic network looking for relationships between keywords.
4. Build a graph of the relationships between keywords (***semantic signatures***).
5. Extract ***network path signatures*** from these semantic network traversals.
6. Sort ***network paths and semantic signatures [patent pending]*** for retrieval of scanned pages and documents.
7. Track user behaviour (desktop clicking, voice, eye movement, etc) to ***modify*** the ***strength*** of these ***network paths***.
8. Return results by comparing the query signatures with the stored network path and semantic signatures ***[patent pending].***
9. Return results just with the ***strenghtened network paths [patent pending]*** that refer to files that were deemed to satistfy users.
10. Build semantic ***network path overlays [patent pending]*** using the extracted paths for visual feedback; for example, a different color (e.g. heat-coded), depending on how satisfied, users were with the results of the query; or how strong the relationships between keywords, and the concepts to which they relate, are.
11. Build networks based on ***path-spec*** keywords that can be displayed to users interested, in what the collection of keywords they use, in specifying their file names, look like in a graph ***[patent pending]***.

I. Deciphering languages

Our methodology can be used to decipher unknown languages; by using an iterative process of typing glyphs; to extract glyph sets and glyph modifiers.

The analysis is done on the provided data in order to identify group or set of glyphs, a group being a single glyph or a combination of glyphs. **Assumptions** about the language are left aside; even though a language might seem to be an alphabet based language, or related to a known language.

The statistical analysis is performed to extract positional probabilities, as previously mentioned; but, instead of using already known glyph sets, i.e. **words**; a single unitary glyph is used.. For alphabet-based languages, the unitary glyphs are the letters of the alphabet; for ideogrammic languages, the unitary glyphs are the visual elements constituting an ideogram.

The goal of the analysis is to extract any type of language structure or syntax (formal grammar-type syntax), based on the collection of glyphs, glyph sets, and modifiers. Once a structure has been extracted; then **context** information is **required**. Said context information is taken from the work of anthropologists, or/and by comparing the extracted structure with that of other known languages, which were **similarly analyzed**.

J. DNA Analysis

Our deciphering methodology can also be used in DNA analysis. The same processing is used in order to identify which constitients of DNA represent the glyphs, glyph sets and glyph modifiers of the DNA language. The statistics extracted should be able to say, whether the DNA codons or the bio-chemical components of the codons are the letters of the DNA alphabet. For example, methylated codons can be seen as accented letters or distinct letters. If the bio-chemical components of the codons are the letters; then, the codons themselves can be considered as words in the DNA language.

IV. Conclusion

Our approach may not solve all natural language processing problems; but it does simplify and makes more accurate some of the work required to extract content from unstructured text.

Information retrieval, translation, content extraction, web search-engine construction, and natural language uderstanding of ambiguous text, such as jargon and specialized domain languages, are all improved using our approach.

V. Appendix A

Rule

```
<NODE use="PREV" restrict="T_AllVerbs" absent="!T_AllVerbs">
        <RULE use="NEXT" tokens="T_AnyAdverb" value="^(through)$" step="continue"
                curPresent="T_Prog" keepNext="T_AnyAdverb"/>
        <RULE use="CUR" tokens="T_AnyAdverb" value="^(through)$" step="continue"
                keepCur="T_AnyAdverb"/>
        <RULE use="CUR" tokens="T_Adj" attr="num" nextPresent="T_Noun" keepCur="T_Adj"/>
</NODE>
```

Explanation

The above rule types a sequence of words, starting at one, which could be any type of verb, such as **have**, **be**, **modal**, **progressive**, and **non-modal** or **common**). It first examines the last word, and checks whether it is the adverb ***through***, and whether the middle word can be a progressive verb, such as ***being.*** If a match occurs, then the last word is typed only as an **adverb;** if no match occurs, then the next rule is examined, and then the next. Rules can be skipped, so that the same 3-word window or cluster can be examined by several other rules.

VI. Appendix B

Rule

```
<NODE rule="PS_Noun">
   <RULE tokens="PS_NounPhrase, PS_Noun"/>
   <RULE tokens="PS_NounPhrase"/>
   <RULE tokens="PS_PronounPhrase"/>
</NODE>

<NODE rule="PS_NounPhrase">
   <RULE tokens="PS_Np, PS_PreposPhrase"/>
   <RULE tokens="PS_Np, PS_ConjuncPhrase"/>
   <RULE tokens="PS_Np"/>
</NODE>
```

Explanation

The above rules are very simple; they simply specify the structure of a ***Noun*** phrase.
A ***Noun*** phrase or clause is a sequence of words starting with a basic PS tree or clause,
consisting of **noun, adjective,** and **determinant,** followed by a ***conjunctive*** or ***prepositional*** PS tree.

VII. Appendix C

Text

Blue shows light light emitted by doubly-ionized oxygen atoms.
Noun Verb Adj Noun Part Adverb Adj Noun Noun
Adj Noun Verb Noun Part Adverb Adj Noun Noun

Blues light shows light light emitted by doubly-ionized oxygen atoms.
Noun Adj Noun Verb Noun Part Adverb Adj Noun Noun.
Noun Verb Noun Adj Noun Part Adverb Adj Noun Noun.

The first likes like love were wonderful.
Det Adj Noun Adverb Noun Be Adj
Det Noun Verb Adverb Noun Be Adj

He writes still letters ...
Pron Verb Adj Noun (without a network)
Pron Verb Adverb Noun (with a network, if other words follow letters)

He writes still, letters ...
Pron Verb Adverb Noun

House stills as you like it tills with water.
Noun Verb Conj Pron Verb Compl Noun Prepos Noun
Noun Verb Conj Pron Verb Compl Verb Prepos Noun
Noun Noun Conj Pron Verb Compl Noun Prepos Noun
Noun Noun Conj Pron Verb Compl Verb Prepos Noun

House stills , as you like it tills with water.
Noun Noun Conj Pron Verb Compl Noun Prepos Noun
Noun Noun Conj Pron Verb Compl Verb Prepos Noun
Noun Verb Conj Pron Verb Compl Verb Prepos Noun

He will seem surprised still at his gall.
Pron Modal Verb Adj Adverb Prepos Adj Noun
Pron Modal Verb Adj Adj Prepos Adj Noun
Pron Noun Verb Adj Adj Prepos Adj Noun

Explanation

The above examples indicate how a given sentence can be parsed;
based on the extracted statistics that are part of our engine.

The use of elementary school grammar-like parsing leads to a parsed sentence,
which may or may not be correct, The parsing itself ***cannot be used to gauge***,
whether or not, the sentence is a ***correct one semantically***.

VIII. Appendix D

Text

Elementary, my dear Watson.
Easy, my dear emily.
Simple, my sweet elisabeth.
Adj Comma Pronoun Adj Noun

Elementary my dear Watson.
Easy my dear emily.
Simple my sweet elisabeth.
Adj Pronoun Adj Noun

Explanation

The above examples indicate, how a given sequence carries the same semantic information; even though, the actual word sense, associated with each word type, may be different.

When these sequences are ***often*** used; they become part of usual speech.
They are tagged as usual sayings or colloquialisms.

Our ***CETE*** **NLP** engine keeps a database of these sequences, seen as usual sayings, colloquialisms, proverbs, vernacular speech.

Statistical translation systems may not find matches for these particular sequences; since they deal with word-sense and word-sense only.