

# A Cognitive approach to Metaphor Detection

Internship Project Report <sup>1</sup>

May - July 2016

by

**Sheetal Shalini**

National Institute of Technology Surathkal

under the guidance

of

**Prof. C. E. Veni Madhavan**

Informatics and Security Laboratory  
Computer Science and Automation  
Indian Institute of Science  
Bangalore – 560 012 (INDIA)

---

<sup>1</sup>under the Internship Programme, Informatics Laboratory, CSA, IISc 2016

## Acknowledgements

I would like to express my sincere gratitude to and respectfully acknowledge Prof. Veni Madhavan for his constant support and guidelines which have enabled me to showcase immense dedication required for the successful completion of this project. He has been the ideal mentor, who bolstered my ideas and paved the way for their implementation.

I would like to thank Siddhartha Sir for the valuable machine learning lectures that kept me ever engrossed.

I also thank my teammate Aparna for her cooperation which enhanced my productivity to a great extent. My other teammate Akhila has supported me throughout.

Last but not the least, I want to thank my family and friends for their encouragement and motivation, without which this project would have remained a dream.

# Abstract

In an ongoing research programme called DIAMETERS, we aim to aid in the development of a Cognitive Markup Language (CML) to solve various types of Natural Language Processing (NLP) tasks such as metaphor detection, sarcasm detection, irony detection, Anaphora Resolution and named entity recognition. DFS Labelling of the stanford parse tree for identifying chunks, along with its Concreteness-Abstractness measure was performed and evaluated. Dependency tags for each sentence were generated and this helped in determining phrasal relationships. Detailed discussions on machine learning techniques were held, and their applications in NLP tasks through deep learning and neural networks were demonstrated.

**Key words:** Cognitive Science, Metaphor Identification, Sarcasm Detection, Machine Learning.

# Contents

|   |           |
|---|-----------|
| Acknowledgements  | i         |
| Abstract  | ii        |
| <b>1 Introduction</b>   | <b>1</b>  |
| <b>2 Background Study</b>                                     | <b>3</b>  |
| 2.1 Cognitive Parsing . . . . .                               | 3         |
| 2.2 Anaphora Resolution . . . . .                             | 4         |
| 2.3 Metaphors . . . . .                                       | 4         |
| 2.4 Dialogues, Expansion and Rewriting . . . . .              | 5         |
| 2.5 Irony Detection . . . . .                                 | 6         |
| <b>3 Implementations</b>                                      | <b>8</b>  |
| 3.1 DFS Labelling Of Parse Tree . . . . .                     | 9         |
| 3.2 Dependency Tagging . . . . .                              | 10        |
| 3.3 Concreteness-Abstractness (C/A) . . . . .                 | 10        |
| 3.3.1 Stemming . . . . .                                      | 13        |
| 3.4 Tolerance and C/A Ambiguity Analysis . . . . .            | 14        |
| <b>4 Machine Learning and NLP</b>                             | <b>15</b> |
| 4.0.1 Empirical Risk Minimization . . . . .                   | 16        |
| 4.0.2 Probably Approximately Correct (PAC) Learning . . . . . | 16        |
| 4.0.3 Convex Learning Problems . . . . .                      | 17        |
| <b>5 RDF and OWL</b>  | <b>18</b> |
| <b>6 Conclusion and Future Work</b>                           | <b>19</b> |
| References  | 20        |

# A Cognitive approach to Metaphor Detection

Sheetal Shalini

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of humancomputer interaction. It is broadly classified into two aspects:

- (i) Natural Language Understanding (NLU), which includes enabling computers to derive meaning from human or natural language input and,
- (ii) Natural Language Generation (NLG), which deals with the production of natural language in the form of meaningful phrases by the computer from some internal representation.

There are three levels of understanding any natural language:

- (i) Syntactic level, which deals with the study of the combinatorics of units of a language, without referring to its meaning,
- (ii) Semantic level, which deals with the meaning of a sentence and, (iii) Cognitive level, which deals with the understanding of the sentence as a whole based on intellectual and factual knowledge.

Cognitive Markup Language (CML), which is a  $\langle \text{tag}, \text{val} \rangle$  system for better understanding of natural language, will be developed as a part of the project. It involves deriving those aspects of language which require real world knowledge.

## 1 Introduction

Natural language processing (NLP) and Natural language understanding (NLU) are two endeavours in which humans perform well. Machine processing of language, generally called NLP, is carried out through various foundational and advanced methods. In our generic programme of research in NLP and NLU, we plan to synthesize approaches from cognitive science, computational linguistics and formal theories of grammar. Towards this, an approach termed cognitive markup language (CML) has been initiated by Prof. Veni Madhavan [?]. In an ongoing pro-

programme of research termed DIAMETERS [[?], the efficacy of the cognitive approach would be compared in relation to other standard approaches to solving certain types of NLP tasks. The acronym of the programme indicates an ambitious attempt to handle the canonical NLP tasks of Dialogs, Metaphors, Translation, Expansion, Rewriting and Summarization.

NLP consists usually of the main stages of syntactic parsing, certain types of stylistic ambiguity resolution, and semantic resolution. These are typically handled using (i) representation of lexical and syntactic information structures in the form of parts-of-speech (POS) tags and parse trees, (ii) probabilistic context free grammar (PCFG) together with production rules indicating agreement constraints, and (iii) first-order logic forms.

The resources for carrying out the three stages are organized in the form of various lexical and statistical information, grammar production rules, predicate calculus formulation together with the algorithms for handling these pieces of information [[?]anning and Schutze, Juraffsky and Martin, Allen].

The two foundational ideas from linguistic theories that support the above formulations are the *constituency grammar* of Chomsky and the *dependency grammar* of Tesniere. Machine processing of natural languages are, in general, based on these ideas. In particular, for the English language, the widely used Stanford parser gives useful information in the form of POS tags, the parse tree and typed dependencies. The first two give syntactic and lexical information and the third gives certain amount of semantic information. In our approach, we plan to integrate these items of information with certain types of *cognitive chunks* using the proposed CML.

Typically these cognitive chunks will correspond to subtrees of the parse tree rooted at internal nodes labelled as *phrases* (NP, VP, PP, RP). These provide some informal description of answers to cognitive questions such as *who*, *what*, *when*, *where*, *why*, *how*, *which* etc. Usually, these answers are held in a loosely coupled, distributed cognitive network in the mind. Our ongoing work is on developing a synthesis of the syntactic, semantic and cognitive information. We distinguish the above cognitive approach from two other major contemporary approaches which we term (i) the corpora based machine learning approach, and (ii) domain dependent knowledge ontologies. We have carried out studies on certain NLP tasks such as text simplification [[?] and sarcasm detection [[?] using a combination of lexical, syntactic and semantic features. We plan to extend the scope of these to the tasks mentioned above.

This project on *A Cognitive approach to Metaphor Detection* is in the spirit of the above programme. It is expected to provide analysis and guidelines in support of the programme.

## 2 Background Study

As a part of this research internship, we did an intense study of research papers in order to attain detailed knowledge in this field, as well as to strengthen our concepts so that we can engender creative ideas and implement them with perfection. Our background study included topics like Cognitive Parsing, Anaphora Resolution, Metaphors, Dialogues, Expansion, Rewriting and Irony Detection.

### 2.1 Cognitive Parsing

During the course of this internship, I studied research papers and articles on the use of psycholinguistics for cognitive parsing. Parsing is the assignment of words in a sentence to their appropriate linguistic categories to allow understanding of what is being conveyed by the speaker. It is not simply the assignment of words to simple diagrams or categories, but also involves evaluating the meaning of a sentence according to the rules of syntax drawn by inferences made from each word in the sentence. This evaluation of meaning is what makes parsing such a complex process. When speech or text is being parsed, each word in a sentence is examined and processed to contribute to the overall meaning and understanding of the sentence as a whole. However, parsing cannot just rely on simple grammatical rules as quite often, these thematic categorical components can be assigned to multiple categories or take on multiple meanings that drastically change the meaning of a sentence. This is part of what causes parsing to be so complex as it must go past the basic grammatical understanding of a word or a sentence and apply the correct meaning to it. Parsing allows the reader to make these decisions, based on cues obtained from the words previously read in the sentence and the conclusions that can be drawn from these words. It takes the meaning drawn from what was read previously to allow understanding of what is currently being read. This parsing continues from word to word, sentence to sentence and paragraph to paragraph. Cognitive parsing has a long research history in the fields of machine learning and natural language processing. The state-of-the-art techniques for tackling this task are mostly based on statistical language learning. How human parsing sentences is also an important research topic attracting research efforts for decades in the field of cognitive psychology. Some behavioristic experiments have convinced that the interactionist approach is rational and effective to simulate human parsing mechanism.

## 2.2 Anaphora Resolution

Anaphora Resolution is a term which refers to Pronoun Resolution. This indicates the difficulty in figuring out which pronoun refers to which noun in the sentence.

For Example: The Empress hasn't arrived yet but she should be here any minute.

She → Anaphor

The Empress → Antecedent

Coreferent → Both The Empress and she refer to the same REAL WORLD ENTITY.

Catafora refers to when the anaphor precedes the antecedent.

For Example : Because she was going to the post office, Julie was asked to post a small parcel.

Anaphora Resolution(AR) is the process of determining the antecedent of an anaphor. It is used in Natural Language Interfaces, Machine Translation, Automatic Abstracting, and Information Extraction.

When the anaphor and more than one of the preceding (or following) entities (usually noun phrases) have the same referent and are therefore pairwise coreferential, they form a coreferential chain.

For Example : Sophia Loren says that she will be grateful to Bono. The actress revealed that the U2 singer helped her to calm down when she became scared by a thunderstorm while travelling in a plane.

she → Sophia Loren

the actress → Sophia Loren

the U2 singer → Bono

her → Sophia Loren

she → Sophia Loren

The process of Anaphora Resolution includes identifying anaphors which have noun phrases (NP), verb phrases (VP), clauses, sentences or even paragraphs/discourse segments as antecedents. Most of the AR systems deal with identifying anaphors which have noun phrases as their antecedents. All NP's preceding an anaphor are initially regarded as potential candidates for antecedents. Most approaches look for NP's in the current and preceding sentence. Antecedents which are 17 sentences away from the anaphor have already been reported!

## 2.3 Metaphors

Metaphor is a figure of speech in which a word or phrase is applied to an object or action to which it is not literally applicable. There are three algorithms for automatic metaphor identification, the three algorithms are variations of a single algorithm and they outperform the



state-of-art algorithm by 71% precision. It indicates that one possible approach to metaphor identification is to consider a common-sensical use of the phrase and its violation as an indication of metaphorical use.

It uses the Mutual Information (MI) greater than or equal to 3 as the minimum criterion to indicate any significant statistical association between the words.

$$MI(w1,w2) = \log_{10}((AB*SizeCorpus)/(A*B*span))/\log(2)$$

A → frequency of word w1

B → frequency of word w2

span → span of words

SizeCorpus → Size of the corpus

There is a state-of-the-art algorithm for metaphor identification by measuring the abstractness level of the noun. The more relatively abstract the noun, the more likely is that the phrase functions as a metaphor. But this Concrete-Abstract algorithm is limited and fails in cases like Broken heart where heart is pretty much a concrete noun, so it takes it in the literal sense. The Concrete Category Overlap Algorithm outperforms the Concrete-Abstract Algorithm as it considers along with the abstractness of the noun, the selectional preferences.

Metaphors arise when one concept is viewed in terms of the properties of the other. In other words it is based on similarity between the concepts. Similarity is a kind of association implying the presence of characteristics in common. The phenomenon of metaphor is addressed by identifying a set of linguistic cues indicating it. The criterion chosen in order to determine whether a phrase is metaphorical, is how close the words sense is to its embodied origins. Embodied origin is the way the concept is grounded in sensorimotor experience. This is why 'Big house' is a literal phrase and a 'Big issue' is a metaphorical phrase. In its literal sense, 'Big' points to a spatially large object but its metaphorical extended sense is 'important'. A literal phrase can be traced to its embodied source while a metaphorical phrase is its extension.

## 2.4 Dialogues, Expansion and Rewriting

The ultimate goal of natural language processing (NLP) is the ability to use natural languages as effectively as humans do. As computers play a larger role in the preparation, acquisition, transmission, monitoring, storage, analysis, and transformation of information, endowing them with the ability to understand and generate information expressed in natural languages becomes more and more necessary. The major challenges to NLP systems are reading and writing

text, translation of documents or spoken language, and interpretation of interactive dialogs. They perform functions like analysis (or interpretation) of the input, mapping it into an expression in some meaning representation language (MRL), reasoning about the interpretation to determine the content of what should be produced in response to the user and generation of a response, perhaps as a natural-language utterance or text. Deletion of semantically irrelevant words, substitution of words by standard synonymous words or expressions, transformation of complex forms into simple normal forms, and changing the order of phrases into a predefined normal order are the main rules of rewriting. Each application of a rewriting rule results in a semantically equivalent sequence. An input sentence is submitted to a transformation system that applies the rules in the order in which they are specified. A sequence of words  $S_0$  is transformed into  $S_1$  using rule  $r_1$ ; then  $S_1$  is transformed into  $S_2$  using rule  $r_2$  and so on until no rule can be applied anymore, finally resulting in a sequence  $S_n$  which is the semantic form of  $S_0$ .

There are two different query expansion techniques:

1. Automatic
2. Manual

The automatic expansion algorithm uses title description and narrative fields. The query text is replaced by the output of an algorithm that linked words which appeared in the same sentence at less than 3 words of each other using the phrase operator.

In manual expansion algorithm, the initial natural language topic statement is submitted to a standard retrieval engine via a Query Expansion Tool (QET) interface. The statement is converted into an internal search query and run against the database.

## 2.5 Irony Detection

Irony is the use of words to convey a meaning that is the opposite of its literal meaning. It is a technique of indicating, as through character or plot development, an intention or attitude opposite to that which is actually or ostensibly stated.

The three main types of irony are

1. Verbal Irony : This is the contrast between what is said and what is meant. For Example, sarcasm.
2. Dramatic Irony : This is the contrast between what the character thinks to be true and what we (the reader) know to be true. Sometimes as we read we are placed in the

position of knowing more than what one character knows. Because we know something the character does not, we read to discover how the character will react when he or she learns the truth of the situation.

3. Situational Irony : This is the most common in literature. It is the contrast between what happens and what was expected (or what would seem appropriate). Because it emerges from the events and circumstances of a story, it is often more subtle and effective than verbal or dramatic irony.

The echoic mention approach to irony provides an answer to the question of whom the speakers attitude is directed towards. It determines the victim of irony.

The irony detection algorithm collates an evaluation corpus of 40,000 tweets, which is divided into four parts, comprising one self-described positive set and three other sets that are not so tagged, and thus assumed to be negative. Each set contains 10,000 different tweets (though all tweets may not be textually unique). We assume therefore that our corpus contains 10,000 ironic tweets and 30,000 largely non-ironic tweets. In order to estimate the overlap between the ironic set and each of the three non-ironic ones, the Monge Elkan distance was employed.

$$sim(s, t) = \frac{1}{k} = \sum_{i=1}^K \max_{j=1}^L sim'(A_i, B_j)$$

The irony model consists of four conceptual features and their corresponding dimensions listed below

1. Signatures
  - (a) Pointedness
  - (b) Counter-factuality
  - (c) Temporal compression
2. Unexpectedness
  - (a) Temporal imbalance
  - (b) Contextual imbalance
3. Style
  - (a) Character-grams (c-grams)
  - (b) Skip-grams (s-grams)

(c) Polarity skip-grams (ps-grams)

#### 4. Emotional scenarios

(a) Activation

(b) Imagery

(c) Pleasantness

The representativeness of a given document  $d$  (e.g. a tweet) is computed separately for every dimension of each feature according to Formula

$$\delta_{ij}(d_k) = \frac{fdf_{ij}}{|d|}$$

where  $i$  is the  $i$ -th feature ( $i = 1, \dots, 4$ ),

$j$  is the  $j$ -th dimension of  $i$  ( $j = 1, \dots, 2$  for the unexpectedness feature, and  $1, \dots, 3$  otherwise),

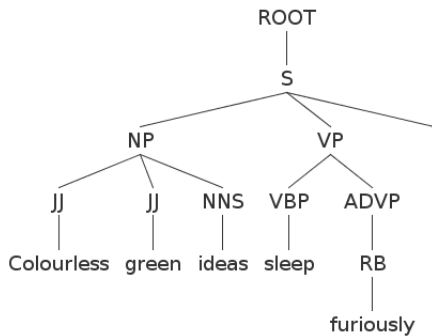
$fdf$  (feature dimension frequency) is the frequency of the dimension  $j$  of the feature  $i$ ,

and  $d$  is the length (in terms of tokens) of the  $k$ -th document  $d$ .

## 3 Implementations

Having done valuable study and research on the vital aspects of NLP, we then proceeded with implementations, which led us a step forward in the direction of cognitive parsing. We started by downloading Stanford Parser along with its POS (Parts-Of-Speech) tagger, and ran it to produce the output as a parse tree and POS tagging of each word in a sentence.

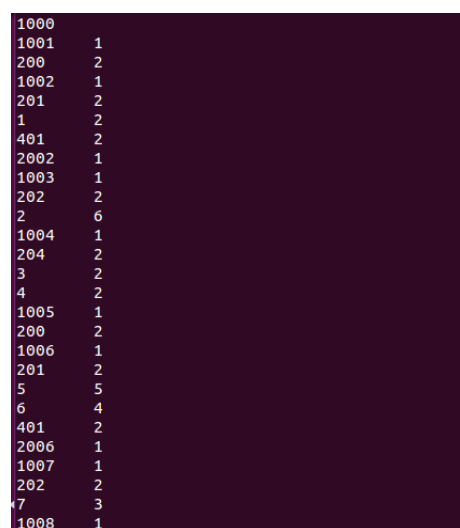
Parse tree is a diagrammatic representation of the parsed structure of a sentence or string. It is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. They are usually constructed based on either the constituency relation of constituency grammars (phrase structure grammars) or the dependency relation of dependency grammars.



### 3.1 DFS Labelling Of Parse Tree

We perform a Depth First Search (DFS) traversal of the parse tree of every sentence included in a file, and represent the conclusions in the following manner :

- There are 2 columns. Column 1 represents the POS tags and Column 2 represents the length factor.
- Column 1 is labelled as follows :
  - Declarative statements (S), Noun Phrases (NP), Verb Phrases (VP), Prepositional Phrases (PP) and Subordinating conjunctions (SBAR) are indicated with starting tags as 200-204 and ending tags as 400-404.
  - Other POS tags are represented by the tags 100-140 and 163.
  - Left and right angular brackets are represented by tags of 1000's and 2000's respectively.
  - Words are represented by their position in the sentence.
  - The beginning of every sentence is represented by a 1000.
- Column 2 is labelled as follows :
  - Words are represented by their lengths.
  - POS tags are represented by 2, and brackets by 1.



|      |   |
|------|---|
| 1000 |   |
| 1001 | 1 |
| 200  | 2 |
| 1002 | 1 |
| 201  | 2 |
| 1    | 2 |
| 401  | 2 |
| 2002 | 1 |
| 1003 | 1 |
| 202  | 2 |
| 2    | 6 |
| 1004 | 1 |
| 204  | 2 |
| 3    | 2 |
| 4    | 2 |
| 1005 | 1 |
| 200  | 2 |
| 1006 | 1 |
| 201  | 2 |
| 5    | 5 |
| 6    | 4 |
| 401  | 2 |
| 2006 | 1 |
| 1007 | 1 |
| 202  | 2 |
| 47   | 3 |
| 1008 | 1 |

## 3.2 Dependency Tagging

The Stanford typed dependencies representation was designed to provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people without linguistic expertise who want to extract textual relations. In particular, rather than the phrase structure representations that have long dominated in the computational linguistic community, it represents all sentence relationships uniformly as typed dependency relations. We have implemented the Dependency tagging of each sentence included in the input file. Each dependency relation is mapped to a number, and the output of the program is represented by 6 columns. These columns indicate the dependency relation, the length of the relation, the Governor, the length of the governor, the Dependent, and the length of the dependent, respectively.

```
[main] INFO edu.stanford.nlp.parser.lexparser.LexicalizedParser - Loading parser from se
CFG.ser.gz ...
done [1.2 sec].
0      0      0      0      0      0
529    3      2      6      1      2
550    3      0      4      2      6
524    3     10      5      3      2
525    3      3      2      4      2
506    3      6      4      5      5
529    3     10      5      6      4
515    3     10      5      7      3
518    3     10      5      8      8
500    3     10      5      9      3
503    3      2      6     10      5

[main] INFO edu.stanford.nlp.parser.lexparser.LexicalizedParser - Loading parser from se
CFG.ser.gz ...
done [0.8 sec].
0      0      0      0      0
529    3      8      5      1      4
529    3     10     11      1      4
500    3      1      4      3      5
500    3      6      7      4      2
500    3      6      7      5      3
500    3      3      5      6      7
550    3      0      4      8      5
511    3      8      5      9      3
500    3      8      5     10     11
506    3     16      8     11     10
500    3     11     10     13      8
506    3     16      8     13      8
511    3     11     10     14      3
500    3     11     10     15      8
506    3     16      8     15      8
520    3      8      5     16      8
```

## 3.3 Concreteness-Abstractness (C/A)

Abstract nouns are words that name things that are not concrete. Our physical senses cannot detect an abstract noun we can't see it, smell it, taste it, hear it, or touch it. In essence, an abstract noun is a quality, a concept, an idea, or maybe even an event. Something that is abstract exists only in the mind, while something that is concrete can be interacted with in a physical way. Concrete verb refers to a verbal aspect in verbs of motion that is unidirectional

(as opposed to multidirectional), a definitely directed motion, or a single, completed action (instead of a repeated action or series of actions). Concrete verbs may be either imperfective or perfective. On the other hand, abstract verbs are always imperfective in aspect, even with prefixes that are normally associated with the perfective aspect.

We formed a base list of around 500 verbs and nouns, and expanded it to a list of 15000 verbs and nouns using the wordNet synset operation.

The inputs to this program were :

- File containing a list of test sentences.
- File containing the extended Noun List.
- File containing the extended Verb List.

We then analyzed each sentence to check for concrete and abstract verbs and nouns, and found out the probability of each of the 4 pairs :

- Concrete Verb acting on Concrete Noun (C-C)
- Concrete Verb acting on Abstract Noun (C-A)
- Abstract Verb acting on Concrete Noun (A-C)
- Abstract Verb acting on Abstract Noun (A-A)

```
Count C-C (Concrete Verb-Concrete Noun) = 129
Count C-A (Concrete Verb-Abstract Noun ) = 77
Count A-C (Abstract Verb-Concrete Noun) = 95
Count A-A (Abstract Verb-Abstract Noun) = 95
Total count = 396
Probability C-C (Concrete Verb-Concrete Noun) = 32.57575757575758
Probability C-A (Concrete Verb-Abstract Noun ) = 19.444444444444443
Probability A-C (Abstract Verb-Concrete Noun) = 23.98989898989899
Probability A-A (Abstract Verb-Abstract Noun) = 23.98989898989899
```

We also labelled each sentence with its corresponding pairs, for easier understanding and depiction.

```

C-C she had your dark suit in greasy wash water all year .
C-C she had your dark suit in greasy wash water all year .
C-A she had your dark suit in greasy wash water all year .
C-C do n't ask me to carry an oily rag like that .
C-C do n't ask me to carry an oily rag like that .
A-C do n't ask me to carry an oily rag like that .
C-C bright sunshine shimmers on the ocean .
C-C bright sunshine shimmers on the ocean .
C-C carl lives in a lively home .
C-C alimony harms a divorced man 's wealth .
C-A alimony harms a divorced man 's wealth .
NA although always alone we survive .
NA most young rise early every morning .
C-C coconut cream pie makes a nice dessert .
C-A coconut cream pie makes a nice dessert .
A-A biblical scholars argue history .
A-C the eastern coast is a place for pure pleasure and excitement .
A-C the eastern coast is a place for pure pleasure and excitement .
A-A the eastern coast is a place for pure pleasure and excitement .
A-A the eastern coast is a place for pure pleasure and excitement .
A-C the prowler wore a ski mask for disguise .
A-A the prowler wore a ski mask for disguise .
NA challenge each general 's intelligence .
C-A upgrade your status to reflect your wealth .
C-A upgrade your status to reflect your wealth .
A-A upgrade your status to reflect your wealth .

```

We then computed the number of concrete and abstract verbs and nouns in a sentence, the total of which gives us the concreteness and abstractness of that sentence. We represented it as a 3-tuple set and the corresponding computation was done in the following 3 steps :

1. The number of concrete and abstract nouns in the *Subject* (first Level-1 Noun Phrase (NP) of the parse tree) of a sentence was calculated and displayed as the first 2 columns of the output.
2. The number of concrete and abstract verbs in the *Predicate* (first Level-1 Verb Phrase (VP) of the parse tree) of a sentence was calculated and displayed as the next 2 columns of the output.
3. The number of concrete and abstract nouns in the *Object* (first Level-2 Noun Phrase (NP) inside the first Level-1 Verb Phrase (VP) of the parse tree) of a sentence was calculated and displayed as the last 2 columns of the output.



```

0 0 1 0 2 1 she had your dark suit in greasy wash water all year .
0 0 1 1 0 0 do n't ask me to carry an oily rag like that .
1 0 1 0 1 0 bright sunshine shimmers on the ocean .
0 0 1 0 1 0 carl lives in a lively home .
0 0 1 0 1 1 alimony harms a divorced man 's wealth .
0 0 0 0 0 0 although always alone we survive .
1 0 0 0 0 0 most young rise early every morning .
1 0 1 0 0 1 coconut cream pie makes a nice dessert .
0 0 0 1 0 1 biblical scholars argue history .
1 0 0 1 1 2 the eastern coast is a place for pure pleasure and excitement .
1 0 0 0 0 1 the prowler wore a ski mask for disguise .
0 0 0 0 0 1 challenge each general 's intelligence .
0 0 1 0 0 2 upgrade your status to reflect your wealth .
2 0 0 1 0 1 cliff 's display was misplaced on the screen .
0 0 0 1 0 0 it 's impossible to deal with bureaucracy .
0 1 1 1 1 0 good service should be rewarded by big tips .
0 0 0 1 0 0 flying standby can be practical if you want to save money .
0 1 0 1 1 0 the thinker is a famous sculpture .
0 0 0 1 0 0 masquerade parties tax one 's imagination .
0 0 1 0 2 0 birthday parties have cupcakes and ice cream .
1 0 0 1 1 0 his scalp was blistered from today 's hot sun .
0 0 0 0 0 0 the cartoon features a muskrat and a tadpole .
0 0 1 0 1 0 clasp the screw in your left hand .
1 0 0 1 1 0 a screwdriver is made from vodka and orange juice .
0 1 1 0 2 0 publicity and notoriety go hand in hand .
1 0 0 0 0 0 the willowy woman wore a muskrat coat .
0 1 0 0 0 0 correct execution of my instructions is crucial .
1 0 1 0 1 0 the water contained too much chlorine and stung his eyes .
1 0 0 1 0 0 the drunkard is a social outcast .
0 0 0 1 2 0 they remained lifelong friends and companions .
0 0 0 0 0 0 the diagnosis was discouraging however he was not overly worried .

```

### 3.3.1 Stemming

In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word-form. There are several types of stemming algorithms which differ in respect to performance and accuracy. A simple stemmer looks up the inflected form in a lookup table. The advantages of this approach is that it is simple, fast, and easily handles exceptions. The disadvantages are that all inflected forms must be explicitly listed in the table: new or unfamiliar words are not handled, even if they are perfectly regular (e.g. iPads v iPad), and the table may be large.

In our program, we use a stemming algorithm which falls into the class called Suffix stripping algorithms. Suffix stripping algorithms do not rely on a lookup table that consists of inflected forms and root form relations. Instead, a typically smaller list of rules is stored which provides a path for the algorithm, given an input word form, to find its root form. Some examples of the rules include:

- if the word ends in ed, remove the ed
- if the word ends in ing, remove the ing
- if the word ends in ly, remove the ly

Suffix stripping approaches enjoy the benefit of being much simpler to maintain than brute force algorithms. However, the algorithm fails when dealing with exceptional relations (like ran and run). The solutions produced by suffix stripping algorithms are limited to those lexical

categories which have well known suffixes with few exceptions. After stemming is performed on the word, it is checked again in the hashmap of words. If the value is not null, the count is incremented else the count continues to be 0.

### 3.4 Tolerance and C/A Ambiguity Analysis

The main hypothesis is on the co-occurrence of incongruous or anomalous word pairs. A pair of words  $P = \langle w_1, w_2 \rangle$  under consideration will be co-located either adjacent to each other or separated by a few words. They could both have the same or different POS tags. They could be used in same or different senses. In the first version of the proposed heuristic, we ascribe concreteness/ abstractness values to the two words of  $P$ , separately. In a proposed augmentation of this scheme, we will ascribe a concreteness/ abstractness value to the pair of words together. While a power law distribution (Zipf’s law) is followed by single word frequencies, it is of interest to utilize the properties of distributions of bi-grams and possibly higher k-grams for small k. We build a feature vector for a collection of labelled sentences,  $S$ , in which the co-ordinate values represent various concreteness/ abstractness scores of the constituent k-grams. Then we use  $S$  as a training set for a two-class supervised learning algorithm. In particular, the learning algorithm could be a simple classification such as k-means clustering algorithm.

The generation of CA scores for a word is as follows: For a word  $w$ , let  $N(w) = \{w \cup \sigma(w)\}$  denote the words in the synset  $\sigma(w)$  together with the word  $w$ . We ascribe the value  $|N(w)|$ —denote the size of the set  $N(w)$ , as the CA score of all elements in  $N(w)$ . We first build the table of scores for a collection  $C$  starting from a small set of baseline words  $B$ . The words in the set  $B$  are given scores by human annotators. For concrete words, we use positive integers, and for abstract words, we use negative integers.

|     |       |       |              |
|-----|-------|-------|--------------|
| 10  | 10    | 0     | cleavage     |
| -10 | -10   | 0     | poverty      |
| 10  | 10    | 0     | sweater      |
| -10 | -10   | 0     | goodness     |
| 10  | 7.78  | 2.22  | jaw          |
| 10  | 10    | 0     | ketch        |
| 10  | 10    | 0     | turkey       |
| 10  | 10    | 0     | trolley      |
| 10  | 10    | 0     | tractor      |
| -10 | -5    | -5    | patience     |
| -10 | -10   | 0     | grace        |
| 10  | 10    | 0     | ferryboat    |
| 10  | 10    | 0     | psychologist |
| 10  | 10    | 0     | meteor       |
| -10 | -10   | 0     | calm         |
| -10 | -10   | 0     | strength     |
| 10  | 10    | 0     | castle       |
| 10  | 10    | 0     | cans         |
| 10  | 10    | 0     | mouse        |
| 10  | 10    | 0     | mother       |
| -10 | -10   | 0     | worry        |
| 10  | 10    | 0     | epee         |
| 10  | 10    | 0     | knives       |
| 10  | 10    | 0     | telephone    |
| -10 | -10   | 0     | difficulty   |
| -10 | -10   | 0     | name         |
| 10  | 10    | 0     | crow         |
| -10 | -6.67 | -3.33 | novelty      |
| 10  | 10    | 0     | book         |
| 10  | 10    | 0     | truck        |
| 10  | 10    | 0     | councilman   |
| 10  | 0     | 10    | curio        |
| 10  | 10    | 0     | gloves       |
| 10  | 10    | 0     | footstool    |
| 10  | 10    | 0     | bracelet     |

In this representation, we have listed abstract and concrete words as -10 and 10 respectively. We have then calculated the Concreteness or Abstractness score by generating synsets of various words, and listed them in the second column. Also, the tolerateness (the difference between the perfect value and the evaluated score) is calculated and listed in the third column.

## 4 Machine Learning and NLP

Machine Learning (ML) is programming computers so that they can learn from input available to them. The input to a learning algorithm is training data, representing experience, and the output is some expertise, which usually takes the form of another computer program that can perform some task. In a basic statistical learning setting, a learner has access to

**Domain set** : A set  $X$  which is the set of objects that we wish to label.

**Label set** : Let  $Y$  denote our set of possible labels.

**Training data** : The set  $S$  defined by  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$  is a finite sequence of pairs in  $X \times Y$  : that is, a sequence of labeled domain points.

**The learners output** : The learner outputs a prediction rule,  $h : X \rightarrow Y$  . This function is also called a hypothesis.

**A data generation model** : Each pair in the training data  $S$  is generated by first sampling a point  $x_i$  according to  $D$  and then labeling it by  $f$  .

**Measures of success** : The error of  $h$  is the probability to draw a random instance  $x$ , according to the distribution  $D$ , such that  $h(x)$  does not equal  $f(x)$ .

*The Bayes Optimal Predictor*

Given any probability distribution  $D$  over  $X$   $\{0,1\}$ , the best label predicting function from  $X$  to  $\{0,1\}$  will be

$$f_d(x) = \begin{cases} 1, & \text{if } P[y = 1|x] > 1/2 \\ 0, & \text{otherwise} \end{cases}$$

For every probability distribution  $D$ , the Bayes optimal predictor  $f_D$  is optimal, in the sense that no other classifier,  $g : X \rightarrow \{0,1\}$ , has a lower error. That is, for every classifier  $g$ ,  $L_D(f_D) < L_D(g)$ .

### 4.0.1 Empirical Risk Minimization

Empirical risk minimizer (ERM) finds  $h_s$  that minimizes the error with respect to the unknown  $D$  and  $f$ . Since the learner does not know what  $D$  and  $f$  are, the true error is not directly available to the learner. A useful notion of error that can be calculated by the learner is the training error the error the classifier incurs over the training sample:

$$L_s(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

### 4.0.2 Probably Approximately Correct (PAC) Learning

**PAC Learnability:** A hypothesis class  $H$  is PAC learnable if there exists a function  $m_H : (0, 1)^2 \rightarrow \mathbb{N}$  and a learning algorithm with the following property: For every element that belongs to  $(0, 1)$ , for every distribution  $D$  over  $X$ , and for every labeling function  $f : X \rightarrow \{0, 1\}$ , if the realizable assumption holds with respect to  $H, D, f$ , then when running the learning algorithm on  $m > m_H(\epsilon, \gamma)$  i.i.d. examples generated by  $D$  and labeled by  $f$ , the algorithm returns a hypothesis  $h$  such that, with probability at least  $1 - \gamma$  over the choice of the examples. Support Vector Machine (SVM) is a linear classifier that chooses the line which maximizes the minimum margin. AdaBoost is an algorithm that has access to a weak learner and finds a hypothesis with a low empirical risk. The weak learner returns a weak hypothesis  $h_t$ , whose 1 error is at most  $\gamma$ . AdaBoost assigns a weight for  $h_t$ . The output of the AdaBoost algorithm is a strong classifier that is based on a weighted sum of all the weak hypotheses.

### 4.0.3 Convex Learning Problems

**Convexity:** A set  $C$  in a vector space is convex if for any two vectors  $u, v$  in  $C$ , the line segment between  $u$  and  $v$  is contained in  $C$ . A convex function may or may not have additional properties as Lipschitzness, boundedness and smoothness.

Implementing the ERM rule on a convex function, in general, is easy and accurate. Convex functions can be learnt efficiently as gradient descent algorithm converges to the global minima of the function as opposed to the non-convex functions where we might end up at a local minima. That is where surrogate losses come into picture. One way to deal with non-convex functions is to upper bound the nonconvex loss function by a convex surrogate loss function and then applying the ERM just as a convex function. E.g. the surrogate loss function for SVM is the hinge loss.

Maximum likelihood estimation returns the Gaussian function that results in the given distribution with maximum probability.

## 5 RDF and OWL

Resource Description Framework (RDF) is a framework for describing identified resources and how they are related to each other. It is composed of three different elements:

- Resources the things being described
- Properties the relationships between the things
- Classes the buckets used to group things

Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies. It is used for developing ontologies compatible with the World Wide Web and has a class of properties associated with itself that allow the system to make sense of first order logic.

## 6 Conclusion and Future Work

We classified the synset words in wordnet successfully and used them to attach a 3 tuple value denoting the CA of the sentence. We carried out a DFS labelling of the stanford tree for identifying chunks and for better readability for machines. Also, for verbs and nouns we assigned to each word a quantity called the CA Score which will be used for future work elaborated below. The main hypothesis is on the co-occurrence of incongruous or anomalous word pairs. A pair of words  $P = \langle w_1, w_2 \rangle$  under consideration will be co-located either adjacent to each other or separated by words. They could both have same or different POS tags. They could be used in same or different senses. In the first version of the proposed heuristic we ascribe concreteness/ abstractness values to the two words of  $P$ , separately. In a proposed augmentation of this scheme, we will ascribe a concreteness/ abstractness value to  $P$ .

While a power law distribution (Zipfs law) is followed by single word frequencies, it is of interest to utilize the properties of distributions of bigrams and possibly higher k-grams for small k. We build a feature vector for a collection of labelled sentence,  $S$ , in which the coordinate values represent various concreteness/ abstractness (CA) scores of the constituent k-grams. Then we use  $S$  as a training set for a two-class supervised learning algorithm. In particular, the learning algorithm could be simple classification scheme such as k-means clustering algorithm.

The three coordinates will represent the first noun in the head NP, the first verb in the succeeding VP and the first noun in the final NP. We treat missing values as 0. The basis for this scheme is the standard production rules, at the highest level of the parse tree, namely  $S \rightarrow \langle NP, VP, NP \rangle$ . We take the first element in each phrase for our initial studies, since most well formed sentences will have this simple and direct form. The idea is to extend the algorithm and the cognitive approach from unigram models to k-gram models.

## References

- [1] Manning Chris and Hinrich Schtze, Foundations of Statistical Natural Language Processing
- [2] Jurafsky Daniel and James H. Martin Speech and Language Processing
- [3] Allen James F. Natural Language Understanding
- [4] anerjee Siddhartha, Nitin Kumar and C E Veni Madhavan. Text Simplification for Enhanced Readability
- [5] Nagwanshi Prateek, CE Veni Madhavan. Sarcasm Detection using Sentiment and Semantic Features
- [6] Davidov Dmitry, Oren Tsur and Ari Rappoport (2010). Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon. Proceedings of the Fourteenth Conference on Computational Natural Language Learning
- [7] Collins, Michael; Singer, Y. 1999. Unsupervised Models for Named Entity Classification. In Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.
- [8] Rau, Lisa F. 1991. Extracting Company Names from Text. In Proc. Conference on Artificial Intelligence Applications of IEEE.
- [9] Thielen, Christine. 1995. An Approach to Proper Name Tagging for German. In Proc. Conference of European Chapter of the Association for Computational Linguistics. SIGDAT.
- [10] Gaizauskas, Robert.; Wakao, T.; Humphreys, K.; Cunningham, H.; Wilks, Y. 1995. University of Sheffield: Description of the LaSIE System as Used for MUC-6. In Proc. Message Understanding Conference.
- [11] Ruslan Mitkov, ANAPHORA RESOLUTION: THE STATE OF THE ART.