

# Linguistic and Statistical methods for Sentimental Analysis

Veronica Sharma MT14028

Inshu Kumar Chugh 2011052

# Assignment\_1: Text Polarity based on BOW Approach

- Calculated sentiment at Sentence level.
- Calculated Text Polarity based on Bag of Words(BOW) Approach.
- For each sentence of review file, sentence is broken down into tokens.
- Preprocessing Includes:
  - Stop words removal
  - punctuation and special character removal
  - POS Tagging is done for each token:
    - Tokens with POS Tags as {NN, JJ, RB, VB} are only retained. Other Tags are discarded
- Negative and Positive Polarity for each tagged token is calculated based on the score present in SentiWordNet dictionary.
- The Overall score for each line/sentence is calculated
- Based on Average Positive and Negative score, the sentence polarity is calculated as positive sentence or negative sentence.
- Precision, Recall, and F-Measure of whole document is calculated.
- Accuracy: 79%

# Assignment\_2: Text Sentiment based on BOC Approach

- Rather than working with parts of text in which sentiment is expressed by polarity of words in it, Bag of Concepts is a Sentic computing approach in which affective and common sense knowledge is gathered from the text.
- The score is calculated based on common sense knowledge obtained from text
- It preserves the overall meaning and emotion of the sentence.
- Difference Between BOW and BOC Approach:
  - Given a Sentence: “ I study in IIIT Delhi College”
  - BOW Approach = {I, study, in, IIIT, Delhi, College}
  - BOC Approach = {study\_in\_Delhi\_College, iiit\_Delhi, delhi\_college}
- The difference clearly states that BOC Approach preserves the meaning of the sentence while the BOW approach just divides the text into various pieces and evaluates the polarity of a sentence based on affect terms and polarity terms.
- BOC Approach forms a common sense knowledge graph based on parts of speech tags.
- The approach helps in various online product reviews where people use to express their sentiments for various products or product features. Approach extracts the text affection and sentiment and provide the user review of the product.

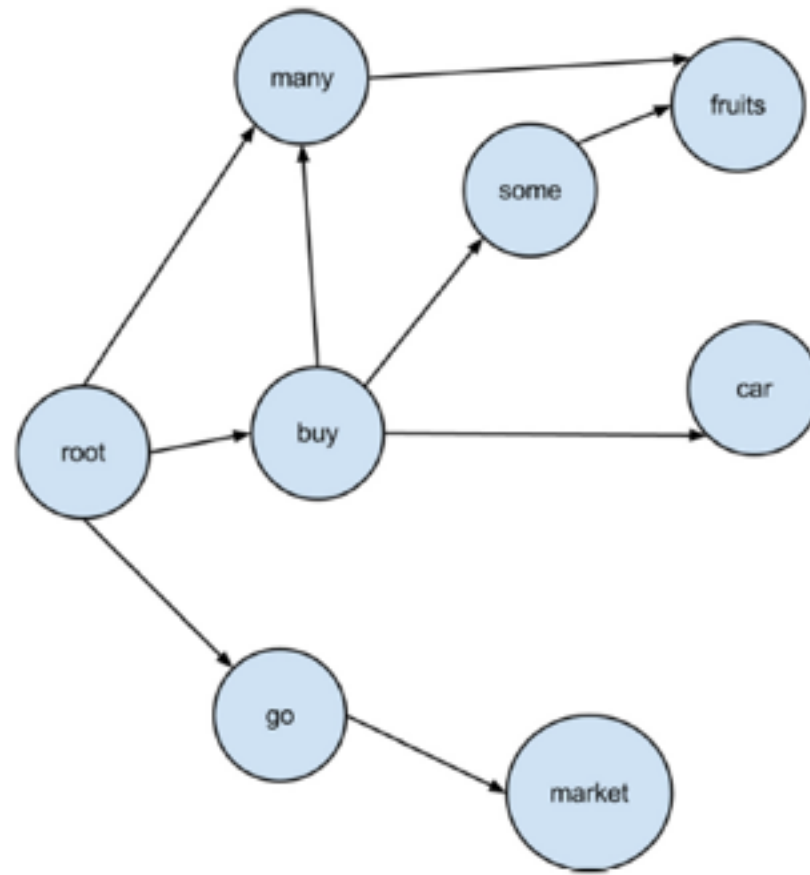
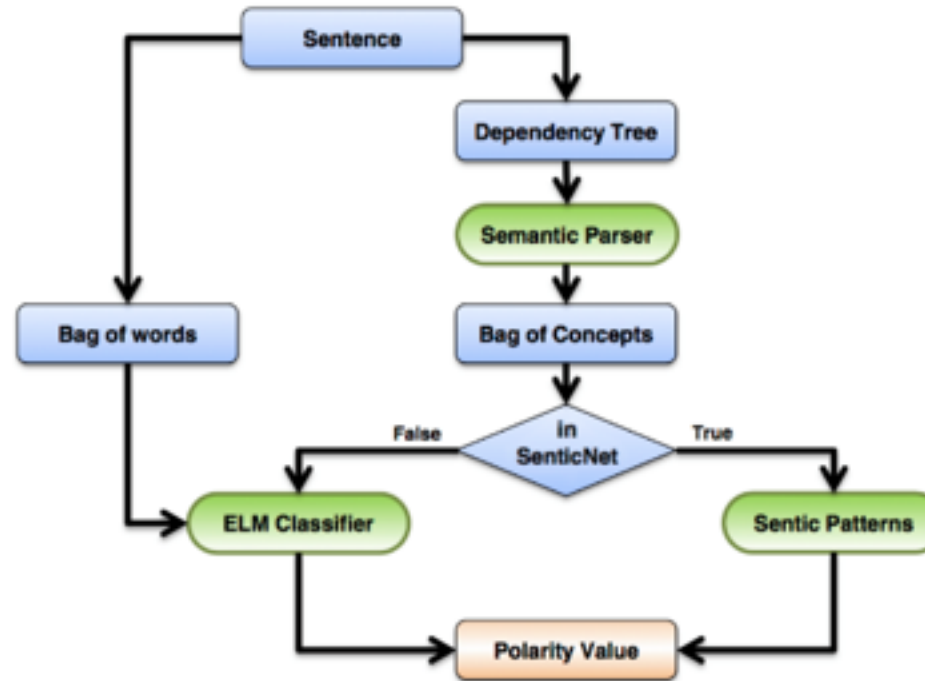


Fig. 1. Example parse graph for multi-word expressions.

Formation of Parse graph based on Concepts model to form Multi Word Expression

# Assignment\_2: Algorithm used

- As stated in the paper and with review of few papers, We are working on following Algorithm:



The Above diagram is taken from SenticNet site.

- We divide the text into various sentences.
- For each sentence, a dependency parse tree is formed using Stanford Parser.
- The parsed data is preprocessed to remove:

# Assignment\_2: Algorithm Used (Contd.)

- Punctuations
- Stopwords
- For Every Noun Phrase, Algorithm1 from Stanford Dependency Manual algorithm is applied.
- Noun Concepts are extracted.
- Similarly , Verb Phrase Concepts are also extracted as described in Algorithm 2.
- The concepts are fed up into SenticNet common sense database which returns the 4 factors on which a sentence's affective valence is calculated:
  - Pleasentness
  - Attention
  - Sensitivity
  - Aptitude
- Total Polarity Value is calculated.

### Algorithm 1. POS-based bigram algorithm

---

**Data:** NounPhrase  
**Result:** Valid object concepts  
Split the NounPhrase into bigrams;  
Initialize concepts to Null;  
**for** *each* NounPhrase **do**  
    **while** *For every* bigram *in the* NounPhrase **do**  
        POS Tag the Bigram;  
        **if** adj noun **then**  
            add to Concepts: noun, adj + noun  
        **else if** noun noun **then**  
            add to Concepts: noun + noun  
        **else if** stopword noun **then**  
            add to Concepts: noun  
        **else if** adj stopword **then**  
            continue  
        **else if** stopword adj **then**  
            continue  
        **else**  
            Add to Concepts: entire bigram  
        **end**  
    repeat until no more bigrams left;  
**end**  
**end**

---

## Algorithm 2. Event concept extraction algorithm

---

**Data:** Natural language sentence

**Result:** List of concepts

Find the number of verbs in the sentence;

**for every clause do**

    extract VerbPhrases and NounPhrases;

    stem VERB;

**for every NounPhrase with the associated verb do**

        find possible forms of *objects*;

        link all *objects* to stemmed verb to get *events*;

**end**

    repeat until no more clauses are left;

**end**

---



# Algorithms Used

## Algorithm 1

- **Semantic Parsing** -The aim of the semantic parser is to break text into clauses and, then deconstruct such clauses into concepts, to be later fed to a vector space of common-sense knowledge.

## Algorithm 2

- The POS-based bigram algorithm extracts concepts from text “I am going to market to buy some fruits and vegetables” as { market, some fruits, fruits, and vegetables}.
- In order to capture **event concepts**, matches between the object concepts(noun) and the normalised(lemmatised) verb chunks are searched.
- This is done by exploiting a parse graph that **maps** all the multi-word expressions contained in the knowledge bases.

- Such an **unweighted directed graph** helps to quickly detect complex concepts, without performing an exhaustive search through all the possible word combinations that can form a common-sense concept.
- Single-word concepts, e.g., house, that already appear in the clause as a multi-word concept, e.g., beautiful house, in fact, are pleonastic (providing redundant information) and are discarded.
- In this way, the Algorithm 2 is able to extract event concepts such as go market, buy some fruits, buy fruits, and buy vegetables, representing the concepts to be fed to a common-sense reasoning algorithm for further processing.

# Progress

## ***Work Done So Far***

- Sentiment calculated using sentiWordNet resource with an accuracy of 79%
- All Preprocessing part is completed
- Noun Phrase Concepts are extracted

## ***Work in Progress***

- Verb Phrase/Event Concepts are need to be extracted
- SenticNet features calculations to find the polarity of sentence

*Thank you*