Semantic Analysis for Narrative Texts

SUMMER RESEARCH FELLOWSHIP PROGRAMME

June 2017- July 2017

A Report

Submitted by

UPASANA GHOSH

Department of Computer Science and Engineering

U.I.E.T, Panjab University

Chandigarh

Under the guidance of

Dr Hariharan Ramasangu

Department of Electronic and Communication Engineering

MS Ramaiah University of Applied Sciences

Bangalore

CERTIFICATE

This is to certify that this summer research work presented in this project entitled ***Semantic Analysis for Narrative Texts*** is the certified and bonafide work of **Upasana Ghosh**, a student of U.I.E.T, Panjab University, Chandigarh under the guidance of **Dr Hariharan Ramasangu**, Department of Electronic and Communication Engineering, MS Ramaiah University of Applied Sciences, Bengaluru in fulfillment of her Summer Research Fellowship, Indian Academy Of Sciences.

Project Guide
Dr Hariharan Ramasangu,
Electronic and Communication Engineering,
MS Ramaiah University of Applied Sciences,
Bengaluru

DECLARATION

I do hereby declare that this project titled ***Semantic Analysis for Narrative Texts*** has been originally done under the guidance of **Dr Ramasangu Hariharan**, Department of Electronic and Communication Engineering, MS Ramaiah University of Applied Sciences, Bangalore in fulfillment of my Summer Research Fellowship.

Project Student
Upasana Ghosh
ENGS6741
U.I.E.T
Panjab University

ACKNOWLEDGMENTS

As we express our gratitude, we must never forget that the highest appreciation is not to utter words, but to live by them.

- *J.F. Kennedy*

I would like to admit that it would have been difficult if not impossible for me to complete this study without support and encouragement of the people to whom I am about to express my gratitude.

I wish to place on record, my sincere gratitude to Dr Ramasangu Hariharan for accepting me to work under his guidance and introducing me to the intriguing field of Natural Language Processing. His perpetual energy and enthusiasm in research motivated me in my studies. It gives me great pleasure to acknowledge his valuable suggestions, constructive criticism and incredible patience.

I am indebted to Indian Academy of Sciences for giving me an opportunity to explore the scientific field of research.

My deepest gratitude goes to my parents for their unflagging love, unconditional trust, timely encouragement, endless patience and support throughout my life. It was their love that raised me up again when I got weary. I am indebted to them for inculcating in me the dedication and discipline to do whatever I undertake.

Contents

Chapter 1

Introduction

A language is a communication tool using which people can share their thoughts, ideas or knowledge. *Natural languages* are the languages that humans used for communications. *Natural Language Processing (NLP)* is a field of computer science which gives computer system the ability to understand, manipulate, generate and process the natural language text. It has evolved from the study of machine learning, linguistic mathematics, robotics, psychology, etc. Some of the applications of NLP are machine translation, information extraction, summarization, speech recognition, automated online assistance etc.

NLP based techniques are the key component in the semantic analysis of the narrative text.

1.1 Stages of Analysis of NLP

NLP is a complex task. The task of analysing natural language is decomposed into multiple stages;

1.1.1 Text Preprocessing

Text preprocessing require filtering irrelevant data from the text that does not aid in the development of NLP system. It can be done by following ways:

Filtering stopwords

Stopwords are the words which appear frequently but do not contribute much to the content of the text. For example, the sentence, *'The book is on the table'* can be expressed as a phrase *'book on table'*. This phrase is enough to represent the same fact that the book is on the table. Hence, we can drop the

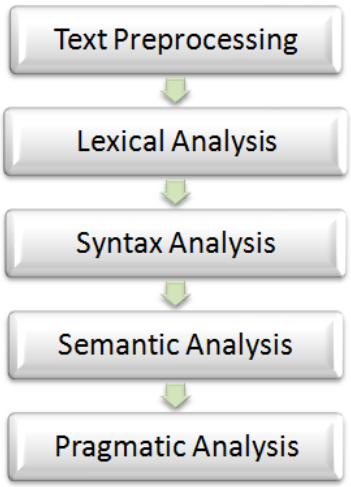

Figure 1.1: Stages of analysis of NLP

words *the* and *is* from the sentence. *'the'* and *'is'* is a stopword. Standford core NLP has a list of 258 stopwords.[1]

Stemming

Stemming is a process of removing the affixes from a word. An affix is a word element that creates a new word if attached to a base or root of some particular words. For example, the word *'running'* will become *'run '* after removing the affix *'ing'* from it. The word *'reactivate '* will become *'activate'* or *'reactive'* after removing the affix from it.

Lemmatization

Lemmatization is a process of finding the root word. For example, the root word of *'running'* is *'run'* and *'reactivate'* is *'active '*.

Word replacement

Word replacement is a technique of replacing a word with some other word. Mostly it is done to make a word more meaningful. For example, The word *'can't'* can be replaced with *'cannot'* and *'we're'* with *'we are'*. The phrase *'not simple'* with *'complex'* or *'difficult'*.

1.1.2 Lexical Analysis

Lexical analysis is the process of analysing the lexeme or tokens in a text. Lexeme or tokens are the collections of words or phrases. It can be done using Tokenization. Tokenization is the process of splitting sentence or phrases into a list of tokens. The process of converting a paragraph into a list of sentences is called Tokenizing paragraph into sentences. For example, the paragraph *'I reached here. The hall in empty. It started raining'* can be tokenized into list of sentences as *'I reached here.', 'The hall in empty.'* and *'It started raining'*. These sentences can be further tokenized into a list of words. This process is known as Tokenizing sentences into words. The sentence *'I reached here.'* Can be tokenized as *'I', 'reached', 'here'*.
Lexical analysis can be performed prior to text preprocessing.

1.1.3 Syntax Analysis

Syntax analysis (or parsing) is the process of analysing the tokens that were generated by the lexical analyser. Syntax analyser requires a grammar and a parser. The grammar will specify the rules of the language and the parser will parse the text and associate tags with the tokens. For example, the sentence *'These flowers are beautiful'*, when parsed through NLTK (Natural Language Toolkit) syntax analyzer will yield *'[[('These', 'DT'), ('flowers', 'NNS'), ('are', 'VBP'), ('beautiful', 'JJ')]]'* where *DT* symbolizes Determiner, *NNS* symbolizes plural noun, *VBP* symbolizes verb and *JJ* symbolizes adjective.

1.1.4 Semantic Analysis

Semantic analysis is the process of analysing the semantic of a text, i.e. determining the meaning of the text and analysing it. For example, the sentences *'Tom is reading a book.'* and *'A book is being read by Tom.'* is syntactically different but are semantically same as the words used are different but both sentences are conveying the same meaning that a person named, *Tom* , is reading a book. Semantic analysis is a complex and tedious task.

A semantic analyser will find it difficult to analyse and understand the meaning of the following utterance:

- "In the midst of winter, I found there was, within me, an invincible summer "– Albert Camus

- "If a man does not keep pace with his companions, perhaps it is because he hears a different drummer."– Walden, Henry David Thoreau
- "The path to my fixed purpose is laid on iron rails, on which my soul is grooved to run."– Moby Dick, Herman Melville

1.1.5 Pragmatic Analysis

Pragmatic analysis is the analysis of a text from the point of view of usage.[2] For example, the last stanza of the popular narrative poem, 'The Road Not Taken 'by Robert Frost,

> "I shall be telling this with a sigh
> Somewhere ages and ages hence:
> Two roads diverged in a wood, and I
> I took the one less travelled by,
> And that has made all the difference "

Here, the poet is talking about travelling inroads in writing. People generally misinterpret it as Frosts choice of not following the crowd, while Frost's intention was to comment about indecision and people finding meaning in inconsequential decisions.[3]

1.2 Components of NLP

1.2.1 Natural Language Understanding

Natural language understanding means understanding the text and drawing inferences from the text. According to Liddy and Feldman[4], in order to understand natural languages, the system should be capable of identifying and distinguishing the following seven interdependent levels:

- the phonetic or phonological level that deals with pronunciation
- the morphological level that deals with the smallest parts of words, that carry a meaning, and suffixes and prefixes
- the lexical level that deals with the lexical meaning of words and parts of speech analyses
- the syntactic level that deals with grammar and structure of sentences
- the semantic level that deals with the meaning of words and sentences

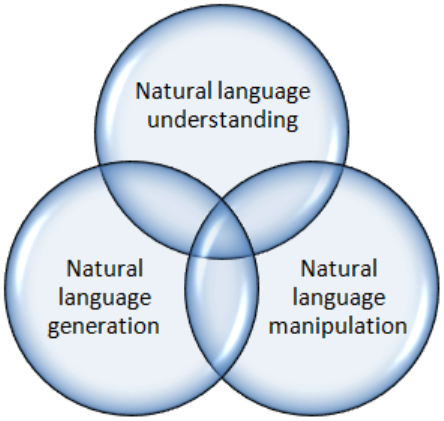

Figure 1.2: Components of NLP

- discourse level that deals with the structure of different kinds of text using document structures and

- the pragmatic level that deals with the knowledge that comes from the outside world, i.e., from outside the contents of the document.

1.2.2 Natural Language Manipulation

Natural language manipulation deals with the manipulation of the text in order to retrieve useful information from it. It also helps in the removal of unnecessary words like stopword from the text. Natural language manipulation has successfully aided in the development of text summarization and question-answering systems.

1.2.3 Natural Language Generation

A natural language processing system should be capable of generating natural language text that could be understood by the humans. For example, Chatbots and intelligent personal assistance generate natural languages to communicate with humans. ELIZA, an NLP system written by Joseph Weizenbaum written between 1964-66, responded to 'My head hurts ' with 'Why do you say your head hurts?'[5]

1.3 Applications of NLP

NLP has a wide range of applications. Some of the applications are mentioned below:

1.3.1 Information Retrieval (IR)

According to Baeza-Yates and Ribeiro-Neto ,"IR deals with the representation, storage, organisation of, and access to information items. These information items could be references to real documents, documents themselves, or even single paragraphs, as well as Web pages, spoken documents, images, pictures, music, video, etc." It is one of the successfully applied domains of NLP. SMART (System for the Mechanical Analysis and Retrieval of Text) is one of the prominent examples of Information Retrieval System developed at Cornell University in the 1960s. TREC (Text REtrieval Conference) is an ongoing series of workshops focusing on a list of different information retrieval (IR) research areas.

1.3.2 Question answering (QA)

A Question answering system is capable of answering natural language questions. It attempts to answer a wide range of questions including temporal and geospatial questions, biographical questions, multilingual questions, and questions about the content of audio, images, and video. Watson is a question-answering computer system built by IBM which is capable of answering open-domain questions. QA system is different from document searching system. Document searching system processes a keyword query and returns a list of documents related to that keyword. QA system processes a natural language question and returns a precise answer. Watson, developed in IBM's DeepQA project, is one of the popular QA systems which was specifically developed to answer questions on the quiz show Jeopardy! and in 2011, it competed against the former winner and won the first place prize of 1 million dollar.[6]

1.3.3 Automatic summarization

Automatic summarization is the process of creating a summary or abstract with the major points from the original document. For example, the Reddit bot 'autotldr'created in 2011 summarises news articles in the comment section of Reddit posts. It was found to be very useful by the Reddit community which garners a lot of upvotes every day.[8]

1.3.4 Virtual assistant

virtual assistant is software which exploits the power of natural language processing to process user text or voice input into executable commands. It is usually available as an application for mobiles and computers. It is capable of providing a wide range of services like providing information from the internet, streaming music, setting an alarm, remembering user preferences, etc. Some examples of virtual assistants are Apple‘s Siri, Google Assistant, Amazon Alexa, Microsoft Cortana, etc.

Chapter 2

ToolKits for NLP

2.1 Popular ToolKits

NLP is a complex task but has a wide range of applications. Due to its complex nature and popularity, many researchers have contributed to the development of NLP toolkits. Some of the popular open source Toolkits available for text processings are:

- **Natural Language Toolkit:** NLTK is a natural language toolkit is a suite of libraries written in the Python programming language for natural language processing.

- **Stanford CoreNLP:** It is a framework licensed under the GNU General Public License. It is capable of processing seven different languages including Arabic, Chinese, English English(KBP), French, German and Spanish. Some of its components are: POS tagging, NER, constituency parsing, dependency parsing, coreference esolution, sentiment. [9]

- **Apache UIMA project:** Software systems that analyse large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIM application might ingest plain text and identify entities, such as persons, places, organisations; or relations, such as works-for or located-at. [10]

- **Apache OpenNLP:** OpenNLP supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, language detection and coreference resolution.

2.2 ToolKits used in the program

- **Python 3.6** Python is a interpreted, interactive, object-oriented, extensible programming language. Python is open source software and has a community-based development model. It is extensively used in:
  - web and Internet development using Django and Pyramid framework
  - scientific and numeric computing using SciPy and Pandas libraries
  - Desktop GUIs using Tk GUI library
- **Natural Language Toolkit 3.0** Natural Language ToolKit (NLTK) is a Python library for natural language processing. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. It has over 50 corpora and lexical resources. Its text processing libraries include:
  - classification
  - tokenization
  - stemming
  - tagging
  - parsing
  - semantic reasoning

Figure 2.1: Python logo

- **Tkinter** Tkinter is a standard Python GUI toolkit. It was written by Fredrik Lundh. It provides an easy way to create GUI applications. It provides:
  - message widget
  - canvas widget
  - text widget
  - buttons
  - event handling
  - layout handling

Chapter 3

Text Processing with NLTK

This chapter will demostrate text processing with NLTK 3.0 and Python 3.6. The following paragraph from a Panchantra story 'The Monkey and the Wedge'is taken for text processing

> There was once a merchant who employed many carpenters and masons to build a temple in his garden. Regularly, they would start work in the morning and take a break for the mid-day meals, and return to resume work till evening.

3.1 Tokenization

Following is the result of tokenizing sentences into words.

['There', 'was', 'once', 'a', 'merchant', 'who', 'employed', 'many', 'carpenters', 'and', 'masons', 'to', 'build', 'a', 'temple', 'in', 'his', 'garden.', 'Regularly,', 'they', 'would', 'start', 'work', 'in', 'the', 'morning', 'and', 'take', 'a', 'break', 'for', 'the', 'mid-day', 'meals,', 'and', 'return', 'to', 'resume', 'work', 'till', 'evening.']

3.2 List of Stopwords

Following is the list of stopwords from the paragraph.

['was','a','who','and','to','in','this','would','the','for','till']

The list of stopwords can vary.

Following is the new tokenized list after removing the stopwords.

['There', 'once', 'merchant', 'employed', 'many', 'carpenters', 'masons', 'build', 'temple', 'his', 'garden.', 'Regularly,', 'they', 'start', 'work', 'morning', 'take', 'break', 'mid-day', 'meals,', 'return', 'resume', 'work', 'evening.']

3.3 Part-of-speech (POS) Tagging

Part-of-speech tagging is the process of adding a tag with each word which signifies whether it is a noun, verb, adjective, adverb, etc.

Following is the list of words along with their POS tags. Table 3.1 presents the list of POS tags along with their descriptions.

[[('There', 'EX'), ('once', 'RB'), ('merchant', 'JJ'), ('employed', 'VBD'), ('many', 'JJ'), ('carpenters', 'NNS'), ('masons', 'NNS'), ('build', 'VBP'), ('temple', 'VB'), ('his', 'PRP$'), ('garden', 'NN'), ('Regularly', 'RB'), ('they', 'PRP'), ('start', 'VBP'), ('work', 'VB'), ('morning', 'NN'), ('take', 'VB'), ('break', 'NN'), ('mid', 'JJ'), ('day', 'NN'), ('meals', 'NNS'), ('return', 'VBP'), ('resume', 'JJ'), ('work', 'NN'), ('evening', 'NN')]]

3.4 Chunks extraction

Chunk extraction is the process of extracting phrases from a POS tagged sentence. The chunk pattern can be modified according to the requirments.

Following is the input to chunk extractor:

[('merchant', 'NN'), ('employed', 'VBD'), ('many', 'JJ'), ('carpenters', 'NNS')]

Following is the output of chunk extractor:

Tree('S', [Tree('NP', [('merchant', 'NN'), ('employed', 'VBD')]), ('many', 'JJ'), Tree('NP', [('carpenters', 'NNS')])])

Figure 3.1 represents the output graphically.

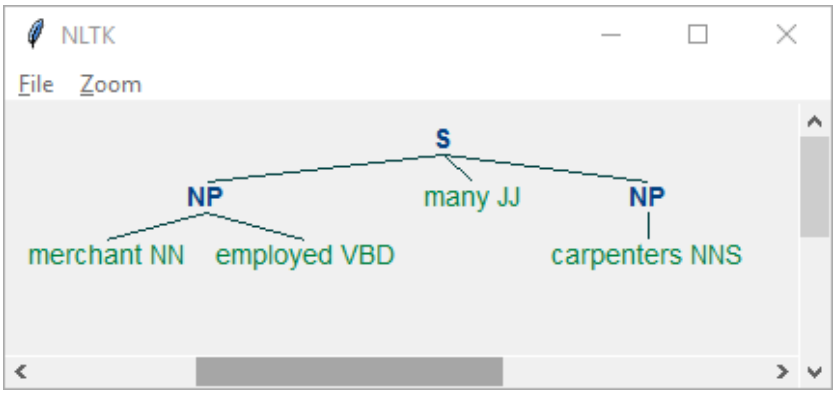

Figure 3.1: Chunks extraction

3.5 Using WordNet

In nltk, WordNet is a lexical database for the English language.

3.5.1 Synsets of a word

Synsets of a word is the list of synonymous words.
Following is the example of how WordNet defines apple

syn = wordnet.synsets('apple')[0]
syn.name()
'apple.n.01'
syn.definition()
'fruit with red or yellow or green skin and sweet to tart crisp whitish flesh'

3.5.2 Lemma of a word

The root word of a word is called as lemma.

lemma.lemmatize('playing',pos ='v')
playing
lemma.lemmatize('caught',pos ='v')
catch

3.5.3 Synonyms of a word

WordNet has the list of synonyms of words in its database. Following is the list of synonym extracted from WordNet of a word *book*.

['book', 'book', 'volume', 'record', 'record book', 'book', 'script', 'book', 'playscript', 'ledger', 'leger', 'account book', 'book of account', 'book', 'book', 'book', 'rule book', 'Koran', 'Quran', "al-Qur'an", 'Book', 'Bible', 'Christian Bible', 'Book', 'Good Book', 'Holy Scripture', 'Holy Writ', 'Scripture', 'Word of God', 'Word', 'book', 'book', 'book', 'reserve', 'hold', 'book', 'book', 'book']

3.5.4 Antonyms of a word

Following is the list of antonyms associated with the word *good* in nltk package.

word = wordnet.synset('good.n.02')
word.lemmas()[0].antonyms()
Lemma('evil.n.03.evil')

word = wordnet.synset('good.n.03')
word.lemmas()[0].antonyms()
Lemma('bad.n.01.bad')

3.5.5 Semantic Similarities between words

NLTK package uses Wu-Palmer Similarity method to find semantic similarity between words. Following example shows the WordNet synset similarity between coffee - sugar and coffee - cat.

- Coffee and Sugar

coffee = wordnet.synset('coffee.n.01')
sugar = wordnet.synset('sugar.n.01')
coffe.wup_similarity(sugar)
0.5882352941176471

- Coffee and Cat

coffee = wordnet.synset('coffee.n.01')
cat = wordnet.synset('cat.n.01')
coffee.wup_similarity(cat)
0.19047619047619047

The Wu-Palmer similarity method shows that sugar and coffee are more realted (59% approx.) to each other than cat and coffee (19% approx). Coffee and sugar are semantically more related than coffee and cat.

Table 3.1: part-of-speech tags

Number	**Tag**	**Description**
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Chapter 4

Semantic Analaysis of Text

Semantic analysis is the process of analysing the semantic of a text, i.e. determining the meaning of the text and analysing it. To measure the semantics of a text, certain *semantic measures* have been developed. According to the literature of natural language processing [13], semantic measure can be any theoretical tool, mathematical function, algorithm or approach which enables the comparison of semantic entities according to semantic evidence. Following are the semantic proxies used for semantic measures:

- **Corpora of texts** - It consists of unstructured or semi-structured texts. These texts contain informal evidence of semantic relationships. For example text, dictionaries, etc.
- **Ontologies** - It consists of a large range of knowledge models i.e. structured vocabularies, highly formal ontologies, etc.

For semantic analysis of narrative texts, we are extracting a list of keywords from a text that is semantically similar to that text.

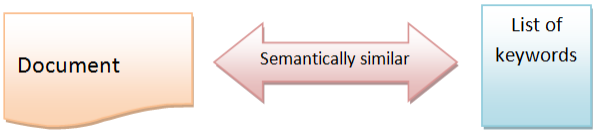

Figure 4.1: Extraction of keywords which are semantically similar to the document text

We are proposing the following five feature for keywords extraction:

- Word frequency

- Sectional weight
- PoS tags
- Co-occurrence
- Sequencing

Following is a story, 'The Monkey and the Wedge' from Panchatantra which is an ancient Indian collection of animal fables.

> There was once a merchant who employed many carpenters and masons to build a temple in his garden. Regularly, they would start work in the morning and take a break for the mid-day meals, and return to resume work till evening. One day, a group of monkey arrived at the site of the building and watched the workers leaving for their mid-day meals. One of the carpenters was sawing a huge log of wood. Since, it was only half-done; he placed a wedge in between to prevent the log from closing up. He then went off along with the other workers for his meal. When all the workers were gone, the monkeys came down from the trees and started jumping around the site, and playing with the instruments. There was one monkey, who got curious about the wedge placed between the log. He sat down on the log, and having placed himself in between the half-split log, caught hold of the wedge and started pulling at it. All of a sudden, the wedge came out. As a result, the half-split log closed in and the monkey got caught in the gap of the log. As was his destiny, he was severely wounded.The wise indeed say: One, who interferes in other's work, surely comes to grief.

4.1 Word Frequency

Word frequency is one of the standard approaches used for keyword extractions. In narrative text, important characters and their actions are repeatedly referred. Using word frequency as one of the features helps the algorithm to identify key characters and their actions. But frequency alone can't be used for summarization as in short narrative text relevant words are rarely repeated. Following is the list of keywords when only word frequency is taken into consideration from the above mentioned Panchatantra Story:

['log', 'wedge', 'one', 'he', 'workers', 'work', 'placed', 'monkey', 'his', 'started', 'site', 'mid-day', 'meals', 'it', 'half-split', 'caught', 'carpenters', 'came', 'wounded', 'wood', 'wise', 'went', 'watched', 'trees', 'they', 'temple', 'surely', 'sudden', 'start', 'severely', 'say', 'sawing', 'sat', 'return', 'resume', 'result', 'pulling']

It is not easy to formulate the story using the above list of keywords. Also, there are dangling pronouns i.e 'he', 'his', 'it'which are not useful for story formulation. Hence, frequency alone can't be used for semantic similarity.

4.2 Linguistic knowledge

Linguistic knowledge helps the algorithm to filter out non-relevant words according to their PoS tags. We have used only nouns and verbs as potential keywords. Nouns in the passage tell us the important characters and hence are helpful in framing the story. Verbs are also important in telling the task the nouns are performing in the text. [14], [15] has shown the importance of linguistic knowledge for identification of important keywords. Following is the list of keywords using word frequency and Pos tags as features :

['log', 'wedge', 'monkey', 'worker', 'work', 'start', 'place', 'meal', 'come', 'site', 'mid-day', 'half-split', 'caught', 'carpenter', 'wound', 'wood', 'wise', 'watch', 'tree', 'temple', 'say', 'saw', 'sat', 'return', 'resume', 'result', 'pull', 'prevent', 'play', "other's", 'morning', 'merchant', 'mason', 'leave', 'jumping', 'interferes', 'instrument']

Adding linguistic knowledge helps in bringing nouns and verbs from the story but still the list of keywords is not similar to the story.

4.3 Word co-occurrence

Word co-occurrence brings those words which are not frequently occurring but are associated with prominent keywords. Hence, word co-occurrence features recommend relevant word with lower word count as candidate keywords. Following is the list of the keywords when word frequency, PoS tags and word co-occurrence is taken into consideration

'work', 'monkey', 'log', 'start', 'worker', 'meal', 'wedge', 'site', 'place', 'mid-day', 'day', 'come', 'build', 'half-split', 'caught', 'carpenter', 'watch', 'leave', 'group', 'building', 'arrive', 'tree', 'play', 'jumping', 'instrument', 'sat', 'return', 'resume', 'pull', 'morning', 'hold', 'break', 'temple', 'result', 'merchant', 'mason', 'garden']

Adding PoS tags, word co-occurrence and word frequency helps in bringing important keywords around which the story is revolving but still the information presented in the above list is scatterd. Hence, sequencing of keywords is important. Also, in narrative stories, we have found that certain sections are more informational than other sections.

4.4 Proposed Features

We have used word frequency, sectional weight, PoS tags, co-occurrence and sequencing as features to extract keywords that are semantically similar to the text. Following is the list of the keywords using our proposed features.

['merchant', 'employ', 'carpenter', 'mason', 'build', 'temple', 'garden', 'start', 'work', 'morning', 'break', 'meal', 'return', 'resume', 'day', 'group', 'monkey', 'arrive', 'site', 'building', 'watch', 'worker', 'leave', 'log', 'wood', 'half-done', 'place', 'wedge', 'close', 'come', 'catch', 'result', 'destiny', 'wise', 'say', 'interfere', 'grief']

The above list is semantically similar to the Panchatantra story mentioned above. Weights are associated with the words according to their PoS tags, frequency, sectional position and co-occurrence.

4.4.1 Algorithm used

The algorithm proceeds as follows. First, the text is divided into sections. Anaphors (pronouns) are resolved with recent antecedent available. Words are then tokenized after stopwords are filtered. Words are annotated with parts of speech (PoS) tags. In order to reduce the size of keywords, a syntactical filter is used which filter only nouns and verbs from the lexical unit list. Weights are assigned to candidate keywords according to their position in the text and the PoS tags associated with the word. The noun which appears first in a sentence (dominating noun) is given more weight than the nouns appearing later in the sentence. Nouns and verbs which are present in the first and last sections are given more strength then present in other sections.

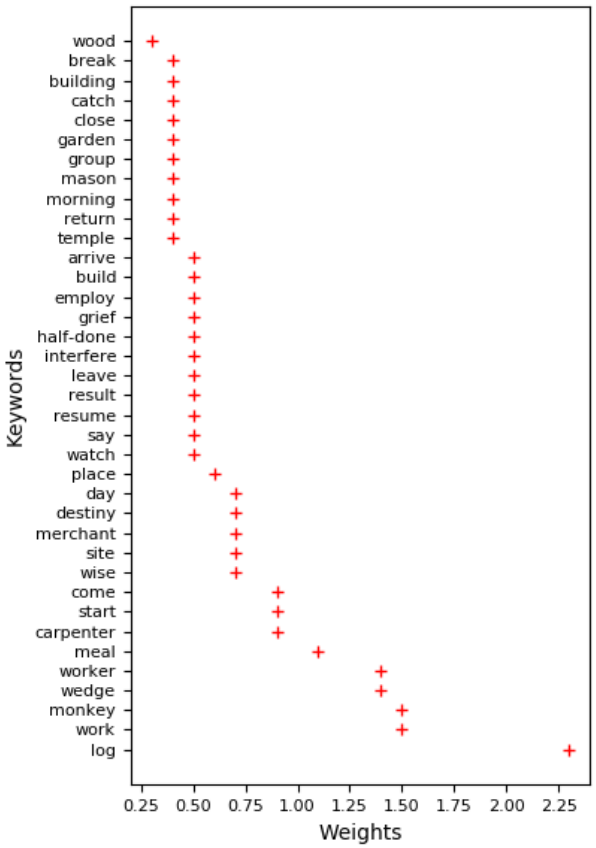

Figure 4.2: Weights vs Keywords

At the end, weights of the words are added and the word co-occurrence is further evaluated and added in the weights. After final evaluation of weights, top 17% of the words are preserved as keywords. The keywords are then presented in a sequential order which will help the reader to formulate the information.

Graph 4.2 represents the weights associated with each keyword. 'log', 'work', 'monkey', 'wedge'are appearing as the top keywords. The above mentioned Panchatantra story is also revolving around the words *monkey, log and wedge.*

Graph 4.3 represents the co-occurrence of keywords in a sentence. Keywords are listed sequentially. Different colour symbols are used to depict keyword's weights in different sections. Keywords like 'log', 'work', 'monkey', 'wedge'are appearing in multiple sections showing their importance in the story. The graph represents that most of the words are occurring only in one section. There are only a few words which are occurring in multiple sections showing their importance in the story.

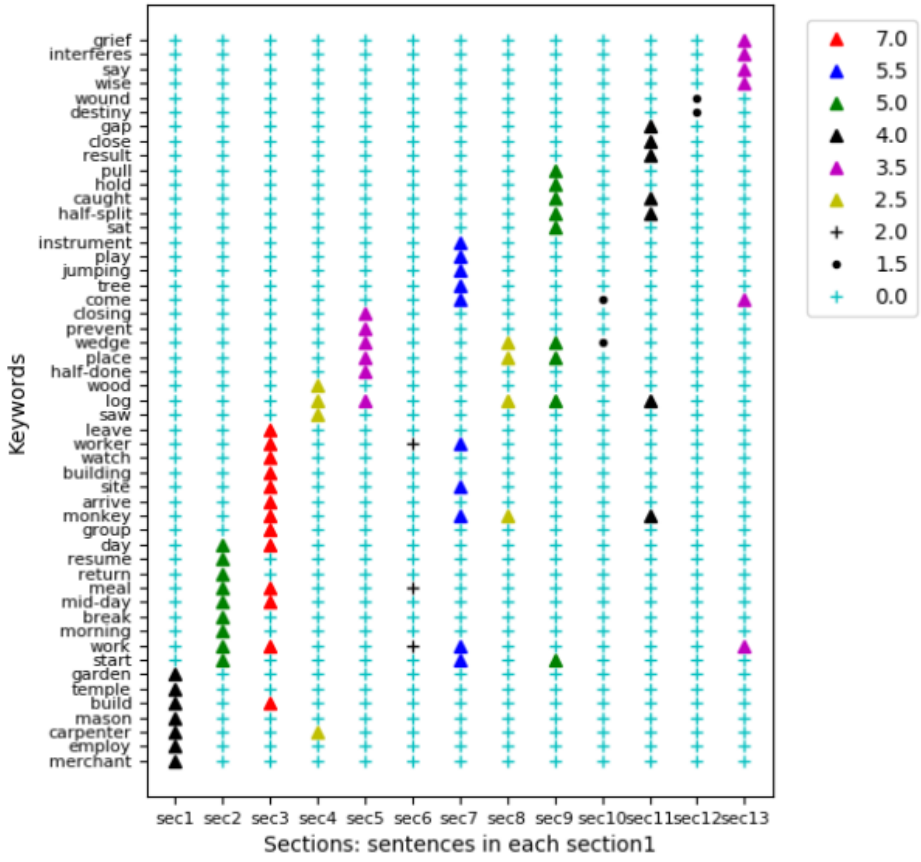

Figure 4.3: Sections vs Keywords

Bibliography

[1] https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt

[2] https://www.uni-due.de/SHE/REV_SemanticsPragmatics.htm

[3] Sterbenz, Christina. "Everyone Totally Misinterprets Robert Frost's Most Famous Poem". Business Insider. Business Insider. Retrieved 13 June 2015.

[4] Liddy, E. (1998). Enhanced text retrieval using natural language processing. Bulletin of the American Society for Information Science, 24, 14-16.

[5] https://en.wikipedia.org/wiki/ELIZA

[6] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading, MA, 1999.

[7] "Dave Ferrucci at Computer History Museum - How It All Began and What's Next". IBM Research. December 1, 2011. Retrieved February 11, 2012.

[8] https://www.reddit.com/user/autotldr

[9] https://stanfordnlp.github.io/CoreNLP

[10] https://uima.apache.org/

[11] Python 3 Text Processing with NLTK 3 Cookbook by Jacob Perkins

[12] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. OReilly Media Inc.

[13] Harispe, Sbastien, et al. "Semantic similarity from natural language and ontology analysis." *Synthesis Lectures on Human Language Technologies* 8.1 (2015): 1-254.

[14] A. Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Japan, August.

[15] Mihalcea R and Tarau P 2004 Textrank: Bringing order into texts. In Proceedings of EMNLP 2004 (ed. Lin D and Wu D), pp. 404411. Association for Computational Linguistics, Barcelona, Spain.

[16] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html