All Purpose Textual Data Information Extraction, Visualization and Querying

by

Syed Usama Hashmi

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2018 by the
Graduate Supervisory Committee:

Ajay Bansal, Chair
Srividya Bansal
Javier Gonzalez Sanchez

ARIZONA STATE UNIVERSITY

August 2018

ABSTRACT

Since the advent of the internet and even more after social media platforms, the explosive growth of textual data and its availability has made analysis a tedious task. Information extraction systems are available but are generally too specific and often only extract certain kinds of information they deem necessary and extraction worthy. Using data visualization theory and fast, interactive querying methods, leaving out information might not really be necessary. This thesis explores textual data visualization techniques, intuitive querying, and a novel approach to all-purpose textual information extraction to encode large text corpus to improve human understanding of the information present in textual data.

This thesis presents a modified traversal algorithm on dependency parse output of text to extract all subject predicate object pairs from text while ensuring that no information is missed out. To support full scale, all-purpose information extraction from large text corpuses, a data preprocessing pipeline is recommended to be used before the extraction is run. The output format is designed specifically to fit on a node-edge-node model and form the building blocks of a network which makes understanding of the text and querying of information from corpus quick and intuitive. It attempts to reduce reading time and enhancing understanding of the text using interactive graph and timeline.

DEDICATION

I dedicate this work to my parents whose prayers and constant support, and faith in my abilities has led to the completion of my master's and this thesis.

ACKNOWLEDGMENTS

I am grateful to Dr. Ajay Bansal for valuable guidance throughout my master's program and through my thesis. I would like to show gratitude for the support shown by Dr. Javier Gonzalez-Sanchez and Dr. Srividya Bansal during my research and for serving on my thesis committee.

I would also like to thank Stanford Natural Language Processing team for providing free to use parsers for developers to work and enhance upon. Without the presence of their open-source and free to use toolkit, this work would take a lot of time.

I would also like to thank developers who contribute to the open-source vis.js library for development of visualization software. The front end used for the tool would not have been possible to develop without the presence of vis.js library which is free to use, in given time, and with the resources available at my disposal.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

APPENDIX Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Overview

Since the advent of internet, the amount of textual data has exploded, and even more since the advent of social media, textual information has grown without bounds. The problem with so much data is that, it is useless if it cannot be analyzed and converted to meaningful, query able information. Not all kinds of text are useful, and not all sources of information are considered reliable. Besides that, different kinds of textual information are important for some, and other kinds are important for others. Visualization techniques have always helped understand large amounts of data and text data is no different. While visualization techniques have existed in a long while, the real trouble with text data has always been the unstructured nature of the data. Other kinds of data are mostly easily encoded into a structured format that can easily be poured into visualization molds, it has been hard to convert textual data into structured formats. These structured formats have also existed for a long time, but the conversion techniques are not exactly very simple and do not work for most kinds of texts. The idea of representing text as a graph has been around for quite some time. The power of graph theory has long been applied to data and textual data and well-known entities have their own versions of knowledge graphs for quick information retrieval.

This thesis explores possible extraction of information into a format consistent with graph theory and one that would make visualization of textual input easy and intuitive to understand

1.2 Statement of the Problem

The problem with large amount of unstructured data such as text is the lack of ability to query the data for information.

This thesis proposes an end to end implementation of "all-purpose" information extraction from text and an approach to visualizing that information with the ability to querying the graph "database" without having to writing any queries but visually interacting with the graph. This thesis also proposes the usage of an interactive timeline along with the interactive graph for querying of the textual information.

Let's define the problem a little more. This is a two-part problem. Explaining the latter part of the problem will help phrase it out better. The visualization of textual data which can be intuitively visualized as well as queried on demand. Having very specific visualization and querying requirements, the extracted information can only be in a specific format and general purpose, the extraction has to be general purpose and fits the desired format.

Requirements:

1. All purpose
2. Intuitively visualize-able/understandable
3. Query-able

Text and natural language is a two-dimensional representation of very high dimensional data that require a lot of human imagination to be converted into the actual

representation of information. Humans can only visualize three physical dimensions and the screens of our computers are mostly two dimensional. Graph theory can be used to understand high dimensional data such as text visualized into two dimensions. We explore ways to convert any given text into a format that can be visualized as a graph or a network and queried for information that the user wants to find out about.

The problem statement also emphasized on large amounts of text. When dealing with large amounts of text, visualization of the entirety of the text can be overwhelming for the user to absorb at once. We explore methods to understand chunks of text quickly while internalizing the information and moving onto the next chunk of information contained in the text.

CHAPTER 2

BACKGROUND LITERATURE

2.1 Text mining

Data mining is the term for extracting meaningful information from data. Text mining is a subset of data mining that deals with extraction of meaningful information from textual data. General text mining process involves text gathering, preprocessing, application of mining technique and analysis of output of the technique resulting in knowledge discovery [1]. Text is unstructured data available as natural language. Natural language is complex owing to its non-regular, non-context free nature (details in [2]). A very simple understanding of complexity of natural language can be gauged with the possibility of ambiguity where same words may mean different things owing to context and different words would mean the same thing at times.

Methods used in text mining are term based, phrase based, concept-based and pattern taxonomy based. Term based focuses on semantic meanings of words based on context identifying polysems and synonyms. Phrase based methods focus on semantic richness in phrases. Concept based deals with building concepts around words in word vector spaces using techniques like TF-IDF [3]. Pattern taxonomy methods rely on pattern mining techniques like association rule mining, frequent itemset mining and closed pattern mining. Text mining techniques involve information extraction techniques like tokenization, named entity extraction and parts of speech tagging, clustering techniques like K-means, and visualization methods like word cloud, graphs networks and trees are

also used [1]. Finally, summarization is a text mining process to retain most related and informational content from text while excluding the rest. With these techniques, text mining is advanced and currently provides powerful insights from texts.

2.2 How does NLP play a part?

Natural language processing deals with the study of grammatical structure of text or speech and not just the statistical information contained in the text. NLP has played a very important part in preprocessing of text for applying data mining techniques, and it has also been used completely independently for text mining tasks. However, a lot of NLP techniques are built using machine learning techniques that are employed directly in data mining as well. For instance, StanfordNLP has used statistical techniques as well as neural networks for its dependency parsers. Natural language processing concerns with how computers can make sense of human languages that are neither regular, nor context free. While statistical text mining techniques work on document similarity and retrieval, NLP plays a key role in information extraction. Almost all text-based information extraction systems are based off natural language processing techniques.

Information extraction has been discussed a lot. The applications and usage of information extraction from textual data will give more insight on the importance of natural language processing. Information extraction is the process of converting unstructured data into a structured format, ideally, filling a database with extracted information. Information extraction starts with entity recognition, which is important in

understanding the real-world entities about whom the information in the text belongs to [4].

Real world entities could be people, organizations, dates, places etc. Coreference resolution helps connect the pronouns, other mentions to the actual entity. Next step in information extraction is relation extraction. Relation extraction task classifies semantic relations between entities. Relations can be used to populate “relational” databases or forming RDF triples or forming edges between nodes usually entities in graphs (graph theory) [5]. Event extraction is the process of extracting events in which the entities got involved. Event extraction also sometimes require extraction of spatio-temporal properties attached to the event. Spatio-temporal information is generally extracted via the named entity recognition process where the information gets tagged as location and date where appropriate. Text parsing, named entity recognition, coreference resolution and dependency parsing are techniques used for certain kinds of information extraction from text hence natural language processing is at the core of information extraction.

2.3 Text Visualization

Data visualization is a technique used to understand data by making its visual representations. The visualization of text or numeric data makes it easier to understand and analyze it clearly and efficiently. Data could be visually represented in the form of tables, graphs and charts to make it user friendly. When there is large volume of data collected it is important to make their visual representations to understand it, this could be done in the form of maps, graphs, tables, charts etc.

People belonging to fields varying from research, health, marketing, education, banking, business have extensive use of data. It is difficult for humans to retain comprehensive text or numeric data or to interpret countless numbers unless they are represented in a way which is meaningful and easier to understand. Therefore, it is very important for individuals to understand what data visualization is, its importance and techniques used to visualize data.

Converting large volume of data into visual forms is not an easy task that could be performed by humans directly. Softwares and tools are required to convert data into visual forms. These softwares create visual representation in the form of maps, graphs, charts by extracting data from the database. The tools which convert data into visual forms are usually connected with database systems. The data visualization tools are flexible that is we can represent data in different forms also we can make modification in the data for particular representations [6].

Daily large amount of data is produced but it isn't known how much amount of data is present in the world as it keeps growing fast. Some studies say that a large proportion of data has been created in the past few years and every day in 2012 about 2.5 billion gigabytes (GB) of data was generated [7].

Data science was not a field known to many a few years back but now due to technological advancement and amount of data that is processed every day, data scientists play an important role in keeping businesses integrated by processing their large amounts of data and representing in simple and meaningful forms for them to understand and make decisions on.

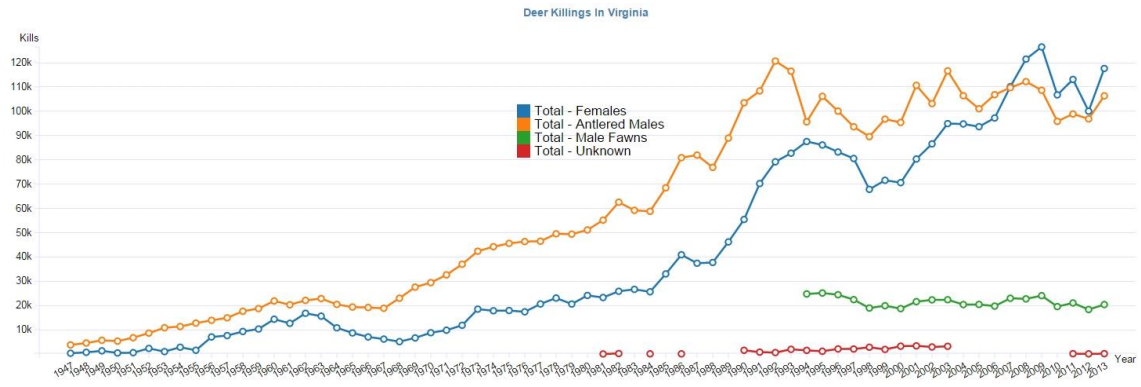
2.3.1 Data Visualization techniques:

Data can be visualized in different ways depending on how big the data is and how it needs to be compared or read. Whether we have to compare categories or make comparison between different sectors, the use of data will decide on how it will be represented. Businesses get added advantage when they have access to data, but they get the real power when they understand the data accurately. Tools used for visualization of data helps in using data in the most productive and efficient manner which leads to increased productivity, revenue and profits. Also, it has led to cutting of cost, saving labor hours and the decision-making process to be faster than before. The following mentioned below are the few visualization techniques:

2.3.1.1 Line Charts:

Line chart shows the relationship between two variables where one variable is dependent on the other. The dependent data are on the vertical axes whereas, the independent data are on the horizontal axes. It is important to see that when we compare data using line charts, the axes should be drawn to same scale. Line graphs are frequently used to analyze change in data over time and show trends. They help in predicting future value of which the data is not yet available, this is done do by following the past trend.

Handling and making sense of text with a lot of data directly and finding relations between variables becomes tedious or an arduous task. That is where line charts come in handy when we want to compare between two variables over the same time frame. An illustration of deer killings in Virginia is visualized in the form of line charts below:

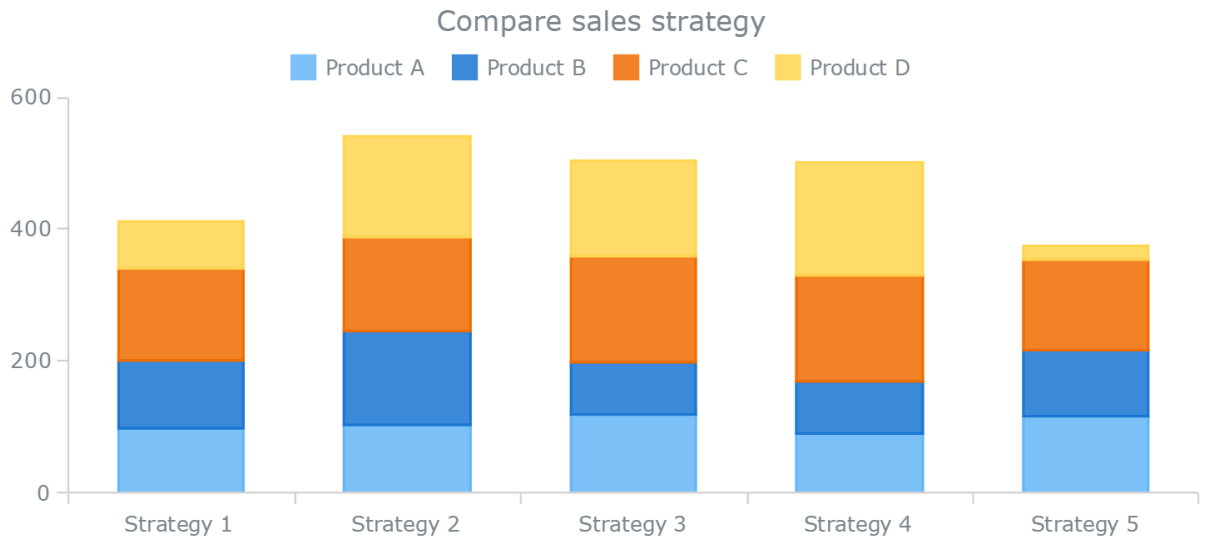


1. Visualization of numeric data to enhance understanding. Source: http://www.appsbi.com/wp-content/uploads/2014/11/deerkillingsfemalevsmale.jpg

2.3.1.2 Bar Charts:

Bar charts are usually used to represent categorical data. The height of the bar is proportional to the value it represents. It can be drawn horizontally or vertically, both, where one axes represents the categories and the other represents the values which is mostly in the form of percentages.

Bar graphs can further be divided into grouped bar graphs, stacked bar graphs and segmented bar graphs, these are used to represent subgroups of the categories.



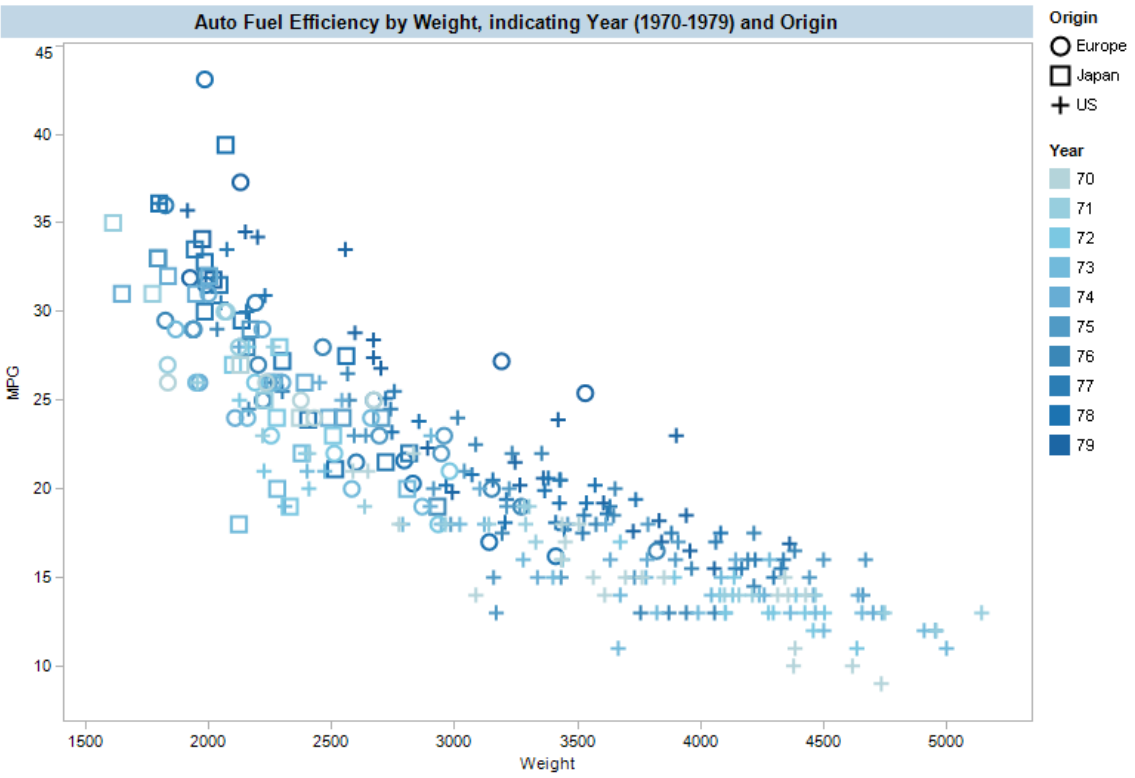
2. Using bar chart to show advantage of strategies on each product. Source: https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/68573aed-e59f-418e-98f7-36f2ae503a1f/5-stacked-bar-chart-large-opt.png

When there is a huge amount of text where certain words are frequently appearing, it can be visualized clearly as a bar chart. An example below illustrates the following:

In the above bar chart, we can clearly which company strategy is more successful and which product in the strategy contributes the same. Understanding the same through text would not have been intuitively and naturally conclusive.

2.3.1.3 Scatter Plot:

Scatter plot is a two-dimensional plot that uses Cartesian coordinates to display values for two variables to show joint variation of the two data items. Scatter plot shows how two variables are related, that is, is the relationship between the variables positive or negative and if positive or negative association then whether strong, moderate or weak.



3. Using scatter plot to visualize trends in dataset and patterns that are otherwise hard to recognize from numbers. Source: http://searchuserinterfaces.com/book/images/auto-scatterplot-years.png

The above graph tries to show the correlation between income of people living in a metro and their health. The relation between the variables seem to have an upward trend. This may not have been found out from just analyzing the text or looking at number

2.3.2 Visualization techniques used with text analytics

2.3.2.1 Word Cloud

It is a picture made of words used to represent a subject where the word's size depends upon the number of times it occurs, or in other words the frequency. Word cloud helps understand the concepts contained in a textual data and the most important words (in terms of frequency) used within the text.

4. Word cloud displaying important words from a corpus. Source: http://www.uxforthemasses.com/wp-content/uploads/2011/06/Word-cloud-min.jpg

The word cloud gives data analyst an idea of what words are most important and what terminologies does the idea really revolve around.

2.3.2.2 Graph

Graphs have played a big role in visualization and exploration of connected data. This thesis identifies graphs or networks as the most viable form of textual information and is discussed in detail in the next section.

This section has shown the importance of visualization when understanding information that is in forms of numbers and alphabets and is not intuitive to the human vision. Next section discusses how text has been used in conjunction with graphs for better understanding.

CHAPTER 3

RELATED WORK

This section discusses work done by others to display text using graphs or networks. When designing a network to contain textual information, certain things must be decided. The two most important questions are: what the nodes/vertices; what information the edge represents. Friendship networks, authorship networks, citation networks etc. are simple kinds of networks created from a more structured text format rather than natural language where, as their name suggests, edges represents friendship, authorship and citation.

Some very interesting methods have been used in the past to place text on graphs and hence information extraction algorithms have come up. Co-occurrence networks map two words as nodes with an edge between them if they occur together. Sometimes, co-occurrence refers to if two words occur adjacent to each other, and other times co-occurrence can also refer to two words appearing in the same sentence. Others have plotted poly-singularity between words to a network. In this kind of graph, nodes are people, concepts, objects and situations and co-occurrence between them will generate a connection between them. Dependency networks are another kind of text represented as graph where word connections are syntactical dependencies between words in a sentence plotted in a graph. Word context networks are defined and generated given a context where entities are pre-defined, and their interactions are discovered.

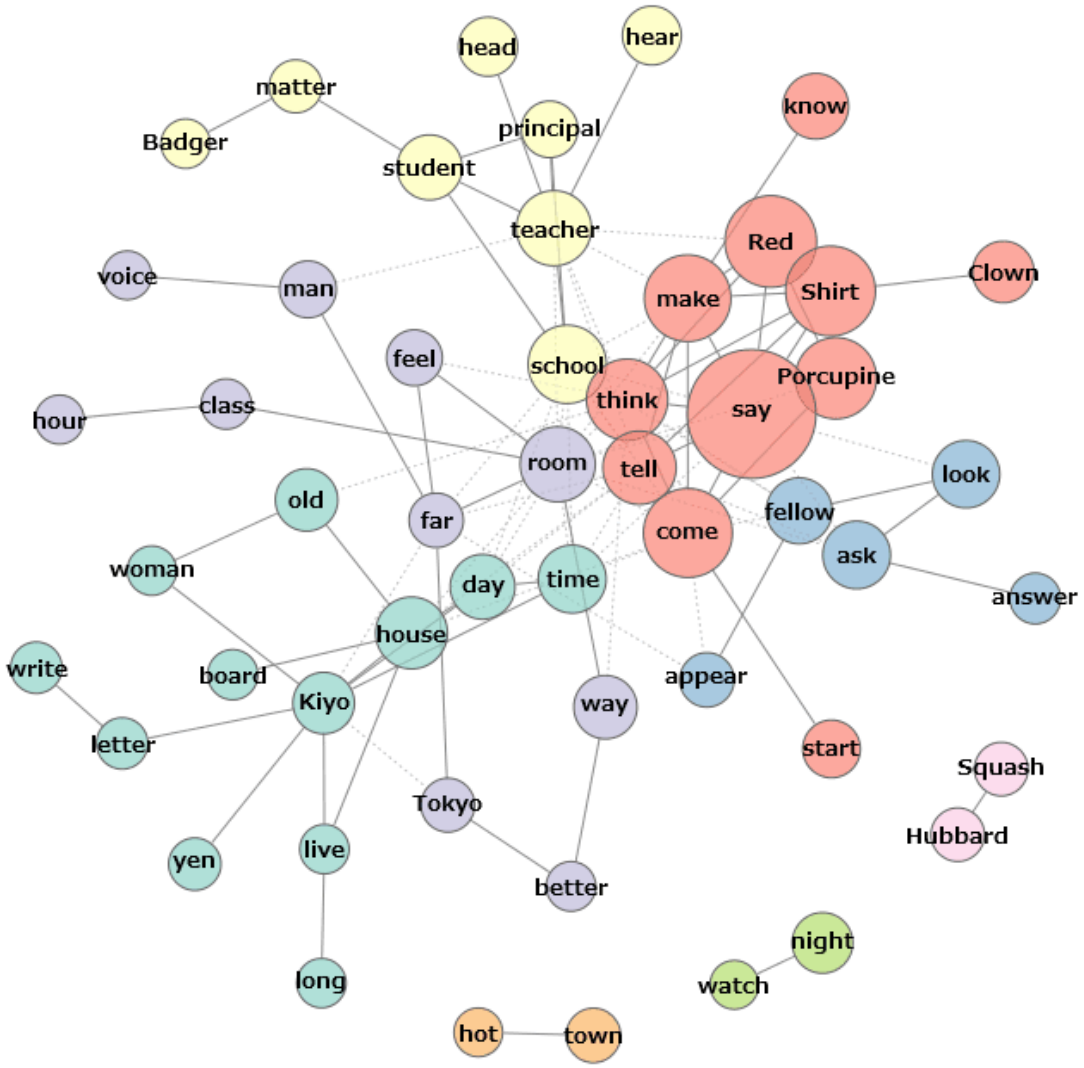
3.1 Co-occurrence Networks

Co-occurrence networks are networks in which two words/nodes will be connected by an edge if they “occur together” [8]. The term occur together is rather unclear opening up a lot of possibilities. Here are some possibilities of networks. After stop words removal, if two words occur adjacent to each other, only then will they have a connection between them. Such a kind of network can be used for implementing techniques of text suggestions or next word suggestions [9].

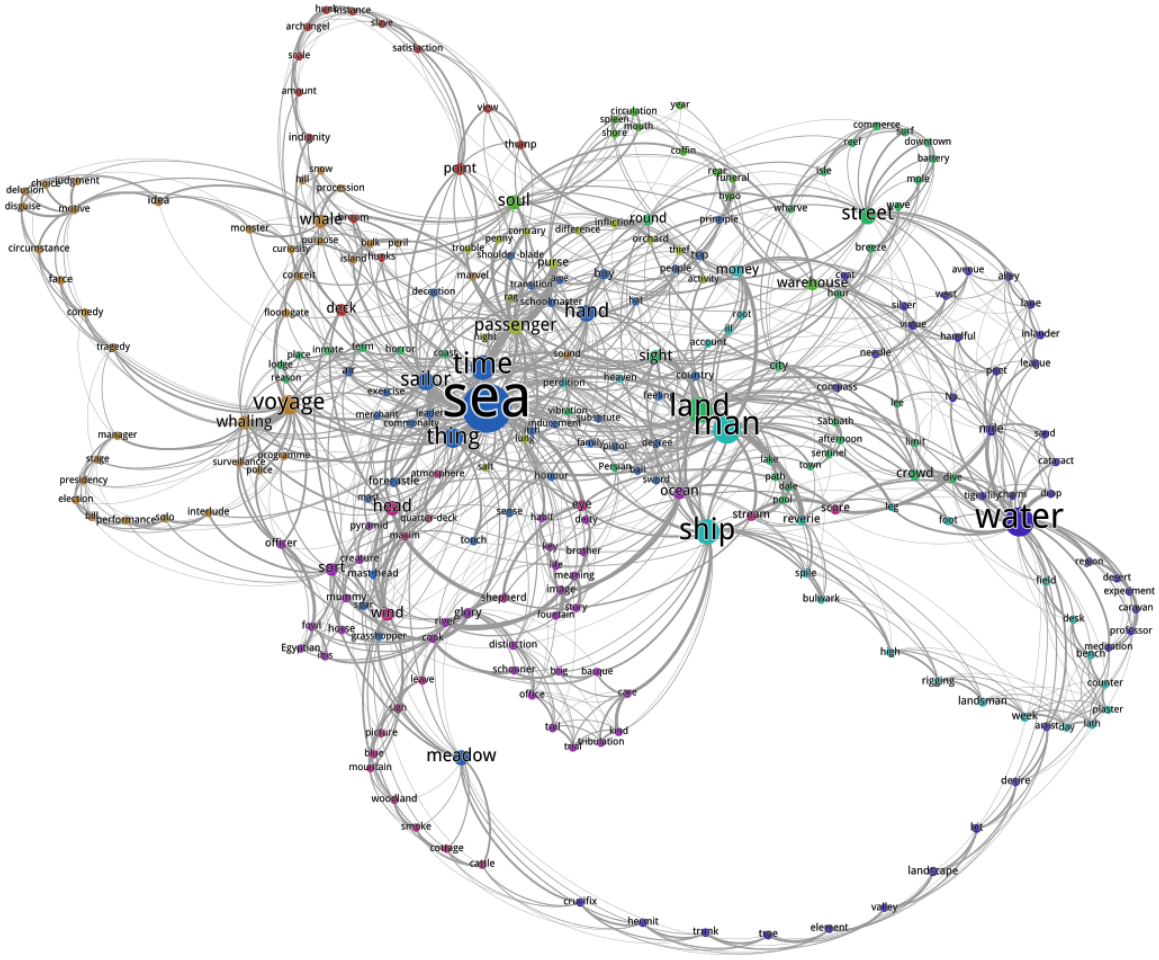
5. Connections between words at a distance of one for co-occurrence networks. Source: https://senereko.hypotheses.org/files/2015/05/cooccurrence04.png

Another possibility is to connect all nodes that have occurred in the same sentence. It would form a rather messy graph with a single node having a lot of connections. One approach as implemented by Nodus labs [10] in their text analysis toolkit is to use a sliding window in any given sentence to set connections. In simpler words, if a word occurs within n words of a given word, it will have a connection with that word.

6. Connections between words at a distance of 4 for co-occurrence networks. Source: https://senereko.hypotheses.org/files/2015/05/cooccurrence07.png



7. Simple co-occurrence network example with a distance of one. Source: https://en.wikipedia.org/wiki/Co-occurrence_networks#/media/File:Khcoder_net_e.png



8. Final network of the first chapter of "Moby Dick". Source https://senereko.hypotheses.org/files/2015/05/moby_dick_chap1.png

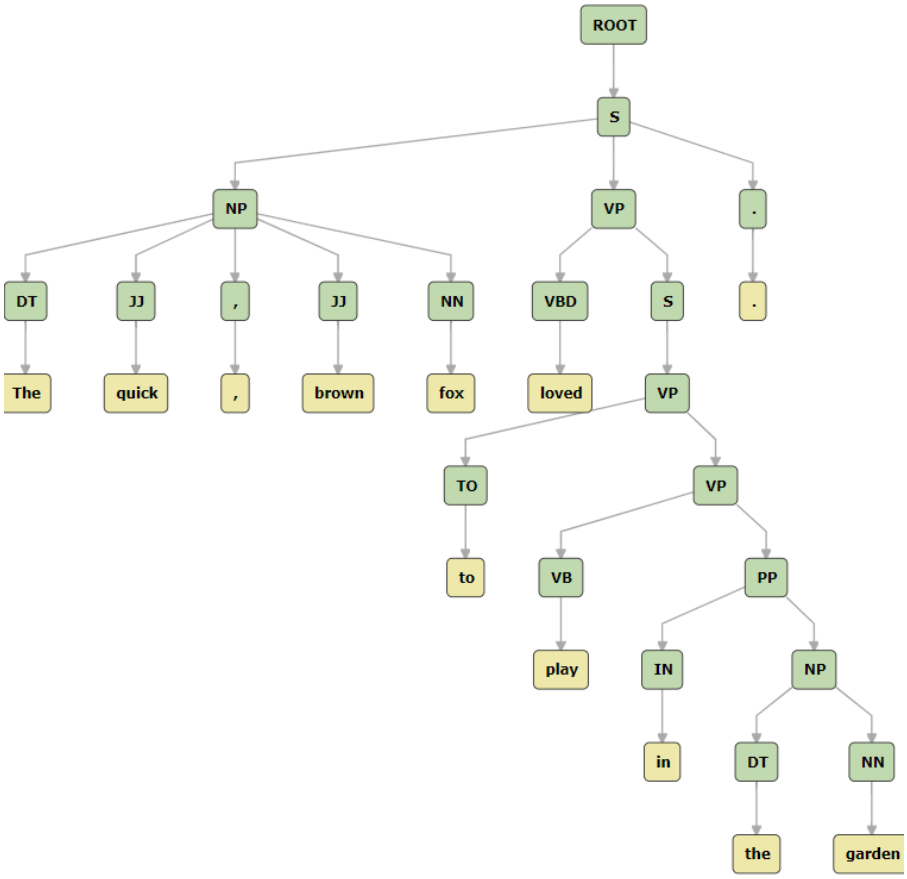
Some important insights from this kind of a network will tell a user most important words in the text and a general idea of what the text is about.

3.2 Dependency Networks

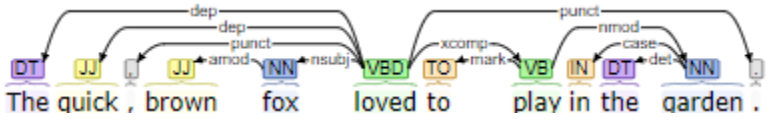
As discussed earlier, dependency parsed trees provide dependency information between words in a sentence. These dependencies can be plotted on a network as well [11]. This provides a more sensible representation of word connections onto a network than just placing co-occurrences. It also solves the problem of having to decide the best value of the moving window while choosing the co-occurrence threshold. By using dependency information, one can ensure that two words will only appear connected in a network if and only if they have some kind of syntactic dependency between them.

The other kind of syntactic structure of sentences can be found in constituency parsing. Constituency parsed information can also be used to generate word networks by connecting only nodes if they have some connection in the constituency tree. However, since dependencies have a more inherent appearance as a network and better connections in terms of a more circular structure rather than a more linear structure, dependencies are preferred over constituencies when dealing with text networks.

The idea of a constituency parse starting from one end of the sentence leads to a structure that is lopsided as we have seen in examples in the NLP section. Constituency parsed trees will generally have more prestige/ centrality on the first verb of the sentence as other verb phrases are sub trees of the first verb.



9. Constituency parse output for the sentence "The quick, brown fox loved to play in the garden."



10. Dependency parse output for the sentence "The quick, brown fox loved to play in the garden."

These images essentially define the nature of these parses. The focus on the word loved and its prestige and centrality in the sentence is well highlighted in dependency parsing whereas in constituency parsing, such information is not supported explicitly. Hence dependency parsed output becomes the choice for more sensible text networks.

Representing dependencies of multiple sentences as a network can be done in a lot of different ways other than just plotting dependencies between words. A simple kind of network based on dependencies can have verbs as edges and everything else in the nodes. A filtered dependency network can be defined such that it only shows specific dependencies and ignores everything else. Depending on what the use-case is, many different kind of filtered dependency networks can be constructed.

3.3 Context Networks

Context networks, as name suggests, are representation of the text in a given context and nodes and edges are defined depending on what the network has to represent. The network can be unimodal or multimodal, in the sense that it can represent one or many kinds of information at the same time [12].

In context networks, ontologies and taxonomies are important. A classification schema needs to be defined before the networks are developed. Different kinds of nodes can be represented in different ways. Different kinds of edges can also be represented in different ways, but ontology must predefine the type of information presented in the node [12].

Certain networks only represent certain relationships for example owner-property relationship, “relative” relationship etc. Graph theory has been used in a lot of other applications rather than just text. A lot of context graphs are used to represent data that are important to the user, and they give a good intuition of how relationships are

changing over time enabling the users to make timely decision based on the information. Some uses are in network intrusion detection, operations management, supply chain management, network load management etc.

CHAPTER 4

NATURAL LANGUAGE PROCESSING

This section starts to discuss the necessary pre-requisite knowledge for understanding of information extraction. Natural language processing almost singlehandedly plays the most important role in information extraction from text. Techniques employed by natural language processing and the tasks they achieved are discussed at length.

Another important reason for the existence of this section being as detailed as it is, is to impress upon the reader, complexity of natural language processing and related tasks. With examples of each of the natural language processing techniques, and complexity and lack of proper grammar of spoken/written human languages or at least relaxation in rules has rendered, what is currently possible with natural language processing techniques quite a feat.

This section aims to show that coming up with a general-purpose information extraction algorithm even after employing current state-of-the-art natural language processing tools and techniques is not a trivial exercise. The complexity and nature of the outputs from natural language processing systems makes them as complex as the paradigm of natural language itself. Hence information extraction algorithm even though built on top of existing tools, is a remarkable feat.

4.1 Techniques

4.1.1 Tokenization

Given a text, tokenization is the technique of splitting the text into tokens where each occurrence of a work becomes a token. The order in which the tokens appear remains the same as the order of the words in the text. One the face of it, tokenization seems like a very simple problem which should not be considered a core in natural language processing. The problem description is simple, using a string of text as input, output an array of string where each string is just a token or a word. The solution becomes tricky because natural language is involved. Ideally, splitting a string on " "(space(s)) should do, then remove all punctuations. But D'Lilah is a name, should it be converted to DLilah? If we ignore ', then from text "Darryl said, 'What are you up to?'" will contain "'What' as a token which is not desired. Complicating it further, should "isn't" be "isnt" or "isn't" or "is" and "n't"? These are only problems faced in English language, other languages make such problems even more complicated. [13] contains more complications and problems. IBM blog on tokenization, "Art of Tokenization" has compared different methods of tokenization and their advantages and disadvantages. Stanford Tokenizer implements a heuristics-based tokenizer under a finite automation algorithm which is deterministic which provides a lot of options for different kinds of tokenizations. Almost all other tasks in natural language processing require tokenization [14].

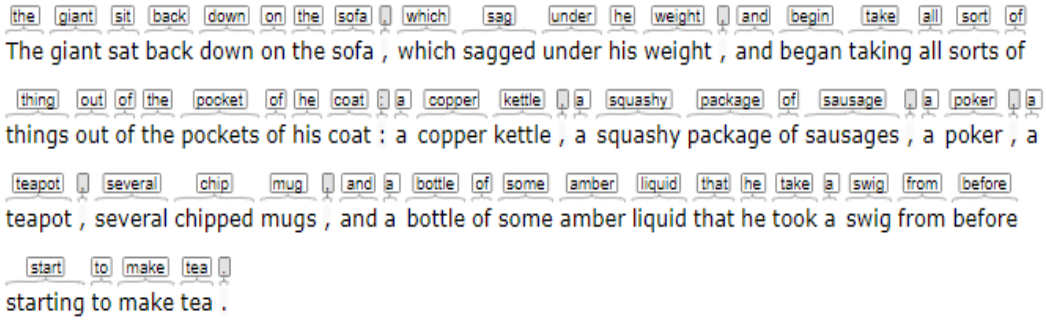
4.1.2 Lemmatization

Lemmatization or stemming is the process of converting a word into its lemma or stem. A stem or lemma can be defined as a root form, or base word from which other words of similar meaning/reference are made. “Be” is a lemma for “is” and “are” and “differ” is a lemma for “difference” and “differential” and so on.

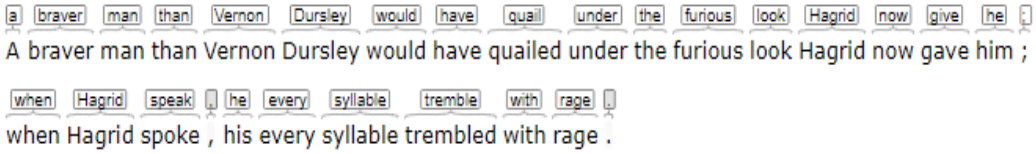
[15] provides a few examples of words and their lemmas. Stemming and lemmatization are essentially not the same things. They are performed differently and produce different results, but the aim of lemmatization is the same. The aim is to be able to discover documents with different derivatives of the given word. For example, in case you need to get all documents containing “differences”, you will miss documents that will have “difference” but not “differences”. With lemmatization/stemming, the search becomes more effective. [16] contains a detailed difference between lemmatization and stemming. Stemmer focuses on removing the suffix or prefix of a word to reach the root word, while lemmatization also deals with changing characters to get to the lemma of the word while taking into consideration morphological analysis of the words. They both have their advantages and disadvantages. Most tasks in natural language processing require lemmatization. Stemming, however is barely used.

The examples below show some examples of lemmatization from Stanford Lemmatizer. Lemmatization/stemming is a relatively simple problem of natural language

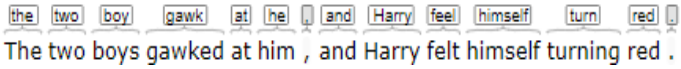
processing and these examples are not complex.



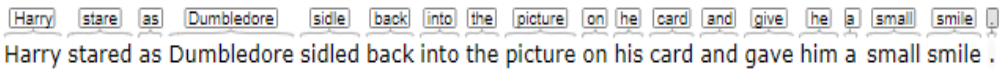
11. Output of Stanford Lemmatization Algorithm for "the giant sat back down on the sofa, which sagged under his weight, and began taking all sorts of things out of the pockets of his coat: a copper kettle, a squashy. package of sausages, a poker, a teapot, several chipped mugs, and a bottle of some amber liquid that he took a swig from before starting to make tea.



12. Output of Stanford Lemmatization Algorithm for "A braver man than Vernon Dursley would have quailed under the furious look Hagrid now gave him; when Hagrid spoke, his every syllable trembled with rage".



13. Output of Stanford Lemmatization Algorithm for "The two boys gawked at him, and Harry felt himself turning red".



14. Output of Stanford Lemmatization Algorithm for "Harry stared as Dumbledore sidled back into the picture on his card and gave him a small smile".

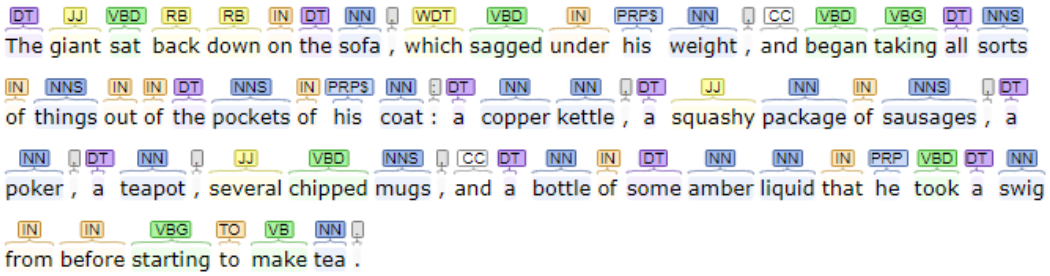
For more examples of lemmatization, see appendix.

4.1.3 Parts of Speech Tagging

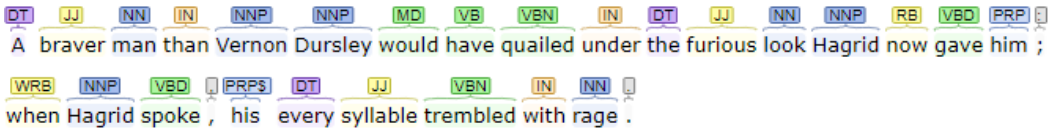
Parts of speech tagging is the process of marking each word with the word class (part of speech) it belongs to [17]. In English, there are two types of word classes namely open class types and closed class types. Closed class types can only take a word from a given set of whereas Open types can take the types of words that are added to the vocabulary of the language. For example, prepositions have been coined and it is unlikely to coin a new preposition whereas nouns are being coined as we progress into new ages. The words selfie and social media are nouns of the 21st century and did not exist before. In English, noun, verb and adjective are some word classes. Penn Treebank POS tag set is the most commonly used set of word classes which has 45 different tags [18]. Knowing the part of speech of each word gives a very good understanding of the importance of the word in the sentence and the structure of words around the given word. It becomes the key to syntactic parsing. They are also very useful in named entity recognition and information extraction. POS tagging can be done using Hidden Markov Models and also by Maximum Entropy Markov Model [19].

POS tagging is a disambiguation task since same words do mean differently in different contexts. For instance, in “Let’s go to that pizza place”, place is a noun, whereas in “Please place it on that table” it is a verb, and book can refer to a novel, and could also refer to the act of reserving/booking a flight/hotel etc. Novel can be an adjective as well as a noun. On page 7 of [20] you can find 6 different tags for the word “back”. Tagging

techniques use n-grams (2-grams and 3-grams generally) to calculate probabilities of the word being of a certain type if it is followed by a certain type and is led by another. While HMM and MEMM techniques are only from left to right, there are techniques that run from right to left, a model named Conditional Random Field. POS Tagging forms the bases of all kinds of parsing done in NLP. Almost all kinds of parsing and even some advanced tagging methods require the text to be POS tagged.



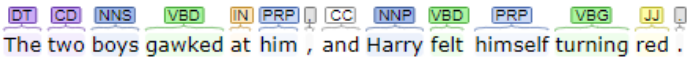
15 Parts of Speech tagged output for " The giant sat back down on the sofa, which sagged under his weight, and began taking all sorts of things out of the pockets of his coat: a copper kettle, a squashy package of sausages, a poker, a teapot, several chipped mugs, and a bottle of some amber liquid that he took a swig from before starting to make tea".



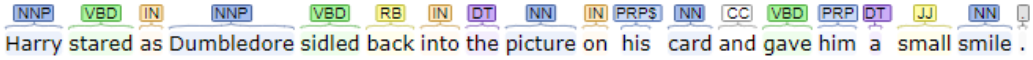
16. Parts of Speech tagged output for "A braver man than Vernon Dursley would have quailed under the furious look Hagrid now gave him; when Hagrid spoke, his every syllable trembled with rage".

In the examples above and below, even though the sentence is big, there are no ambiguities. Some important things to understand here are that all words tagged "IN" provide positional information about the words and might be helpful in determining the position of a certain thing with respect to another. NNS is a plural noun and might find its place on the nodes while VBD are verbs and will find their place in the edges. This is critical information in determining where a piece of information is determined on the

network. Words marked “JJ” are adjectives and will essentially be considered when gathering properties for entities.



17. Parts of Speech tagged output for “The two boys gawked at him, and Harry felt himself turning red”.



18. Parts of Speech tagged output for “Harry stared as Dumbledore sidled back into the picture on his card and gave him a small smile”.

For more examples of Parts of Speech Tagging, see appendix.

4.1.4 Named Entity Recognizer Tagging

Named Entity Recognition (NER) is an important problem in natural language processing. Named entities in a text essentially represent the entities about which the text might be, or some information about these entities are present in the text. Common classification is done into classes called PERSON, LOCATION (COUNTRY, CITY or STATE) and ORGANIZATION. These can be for each word separately or for multi word entities. NER is important for information extraction because extracted NE entities essentially become the information extracted.

Just like POS tagging, NER tagging also faces the same disambiguation problem. New York University or New York University could have been the new “York University” or “New York University”. Or is White House a LOCATION or an ORGANIZATION? The problem is because metonymy is a real construct in natural

language. It is when a certain entity is used to define another because of their proximity to the other, for instance dish would refer to the food instead of the container of the food or Pentagon being used as an organization instead of the building Pentagon. A whole list of very interesting metonyms can be found in [21].

Methods of NER tagging are gazetteering, semi supervised machine learning (bootstrapping) or supervised machine learning techniques. Gazetteering is a method which involves a list of all entities and a substring search algorithm. If substring matches any in the list, you have the NER. It is the simplest algorithm and it is efficient, but not effective. It leads to two major problems. Problems of disambiguation and fixedness. Problem in disambiguation occurs when one word is in two lists representing two different tags. Fixedness refers to having a finite list and nothing beyond the list will ever be recognized. The bootstrapping method requires a small number of seeds and a semi supervised learning algorithm. It fixes the problem of fixedness but does not help much with disambiguation.

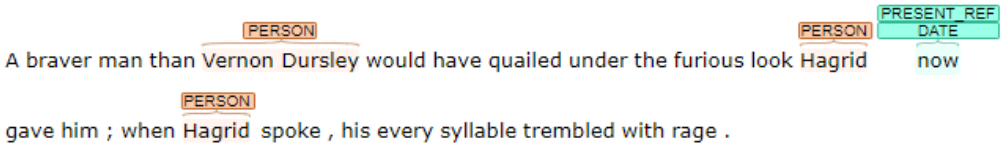
Supervised methods require a manually annotated training set, a manually annotated test set and sometimes a gazetteer. There are discriminative vs generative learning methods that are used. Methods used are Naive Bayes, Hidden Markov Models, Generative Directed Models, Logistic Regression, Linear Chain CRFs and General CRFs.

The giant sat back down on the sofa , which sagged under his weight , and began taking all sorts of things out of the pockets of his coat : a copper kettle , a squashy package of sausages , a poker , a teapot , several chipped mugs , and a bottle of some amber liquid that he took a swig from before starting to make tea .

19. NER Tagged output for "The giant sat back down on the sofa, which sagged under his weight, and began taking all sorts of things out of the pockets of his coat: a copper kettle, a squashy package of sausages, a poker, a teapot, several chipped mugs, and a bottle of some amber liquid that he took a swig from before starting to make tea".

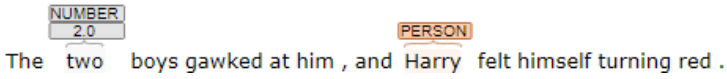
The sentence parsed above has no named entities. Such a sentence is a representative of most sentences that do not contain any proper nouns but have pronouns. Such sentences need to be taken care by using coreference resolution for it to correctly mention the reference of the pronoun.

This sentence can make sense in dependency network or in a context network where all kinds of nouns are nodes and not just proper nouns, but it is one of the complex sentences when keeping in mind the objective of representing it on a graph for intuitive understanding without having to read through the whole of it.



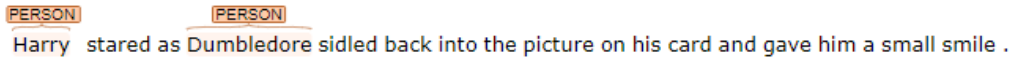
20. NER tagged output for "A braver man than Vernon Dursley would have quailed under the furious look Hagrid now gave him; when Hagrid spoke, his every syllable trembled with rage".

An example of how “now” is recognized as the current date which can be used for timeline generation. The problem here is that not all sentences have dates in them and placing them on a timeline becomes a disambiguation problem.



21. NER tagged output for "The two boys gawked at him, and Harry felt himself turning red".

An example of tagging of numbers present in a text. This is imperative when reading word problem solutions from math books in an attempt for automatic question answering.



22. NER tagged output for "Harry stared as Dumbledore sidled back into the picture on his card and gave him a small smile"

More examples of NER Tagged output are present in the appendix.

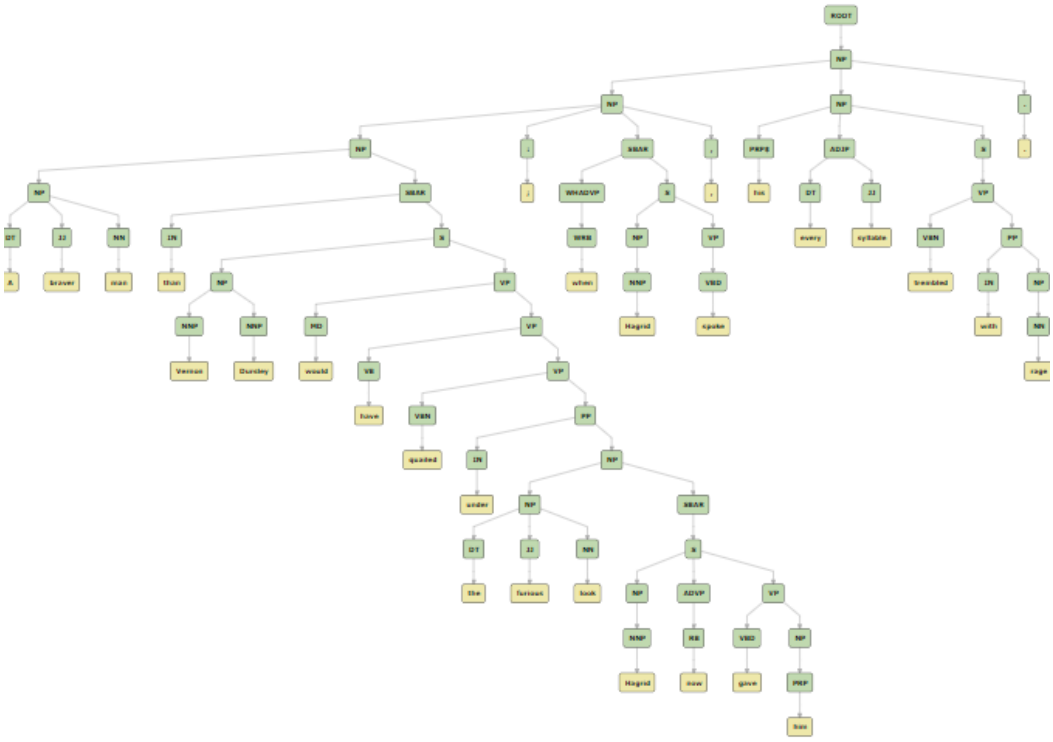
4.1.5 Text Parsing and Parsed Trees

Text parsing refers to the idea of converting text into a parsed tree. Parsing is a simple problem for context free languages and context free grammar. Since we have already established that natural language is not context free or regular language. Parsing performs an important function of breaking the text into its constituent elements that makes up the whole text. Parsing is done sentence wise. Most types of parsing require the text to be tokenized and then POS tagged. Two techniques used for parsing are top down and bottom up techniques [22]. This is a syntactic parsing approach.

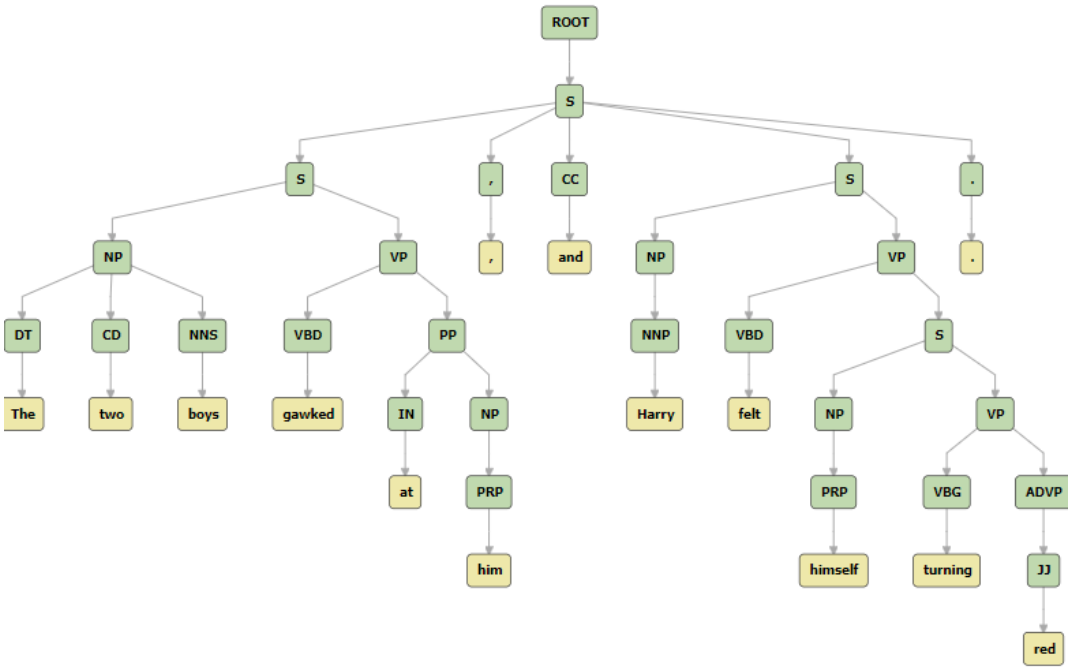
Top down technique starts at the root S and generates all possible combinations of parsed trees by analyzing the rules based on the root node and keeps matching until it finds the parsed tree that completely matches the text. It starts at the root node and works

its ways to the leaves and backtracks if it reaches a node where input string's part of speech does not match the tree's, so essentially pruning a tree without building the entire tree at the first wrong match found [23]. This backtracking is the result of starting to parse without looking at the input. This can be optimized by reading the next most k tokens and apply predictive parsing.

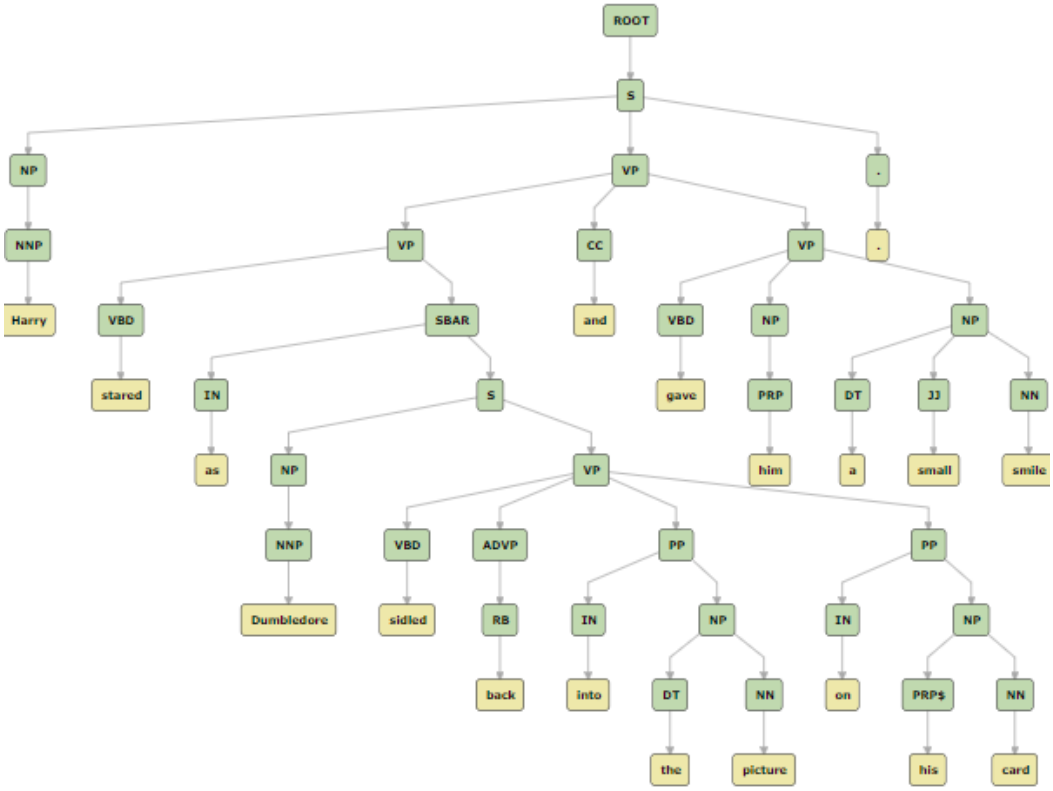
Bottom up parsing starts at the word and works its way up to the node. The parse is successful if the tree up to the node contains all the tokens. It is a data directed search that tries to undo the sentence production process and reduce the sentence to S so essentially it is a reduction process that reduces the tokes to S [23]. As the right side of some rule matches the substring the input token(s), token(s) is(are) replaced with the left-hand side of the rule, when S is reached, parsing stops. It employs the well-known shift reduce paradigm, where shift pushes the next token to the top of the stack and reduce matches the rule and replaces. Once out of tokens and S is not reached, tree is incomplete, and error is raised. If S is reached but all tokens are not used, tree is not correct, error is raised. If shifting reducing to last token reaches root S, parsing is complete. An operational shift reduce parser implements LR parsing with LR(K) grammar with left-right parsing, rightmost derivation done in reverse and k forward lookups. Even with parsers like these, disambiguation issue is still not resolved. This brings us to another form of parsing, called dependency parsing.



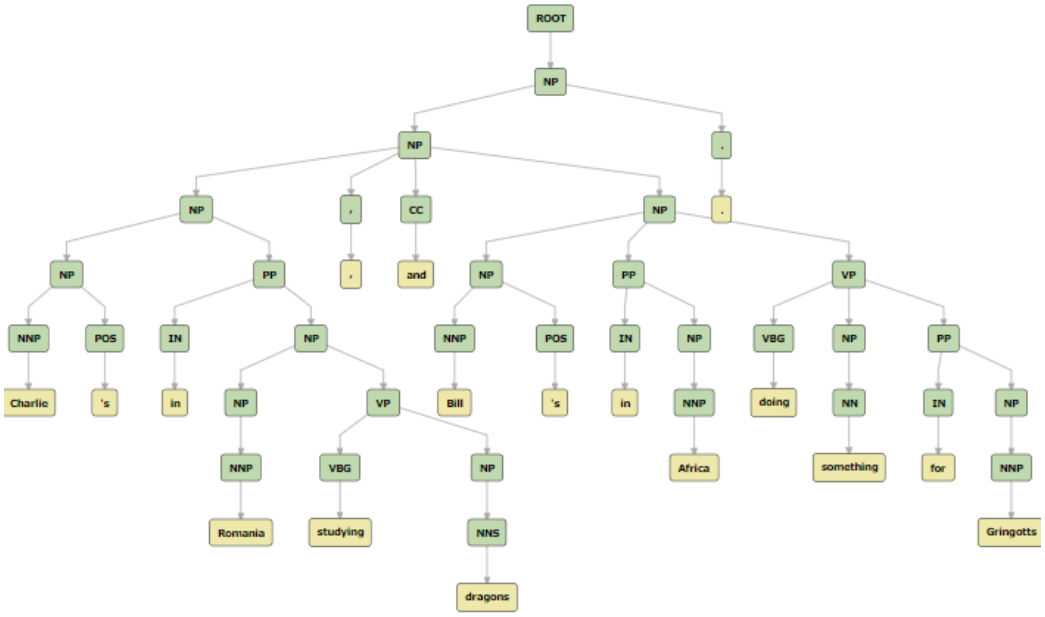
23. Parse tree for "A braver man than Vernon Dursley would have quailed under the furious look Hagrid now gave him; when Hagrid spoke, his every syllable trembled with rage".



24. Parse tree for "The two boys gawked at him, and Harry felt himself turning red".



25. Parse tree for "Harry stared as Dumbledore sidled back into the picture on his card and gave him a small smile".



26. Parse tree for "Charlie's in Romania studying dragons, and Bill's in Africa doing something for Gringotts".

Parse tree can and have been used for information extraction but here are two main reasons why for our purpose, parse trees did not provide enough information.

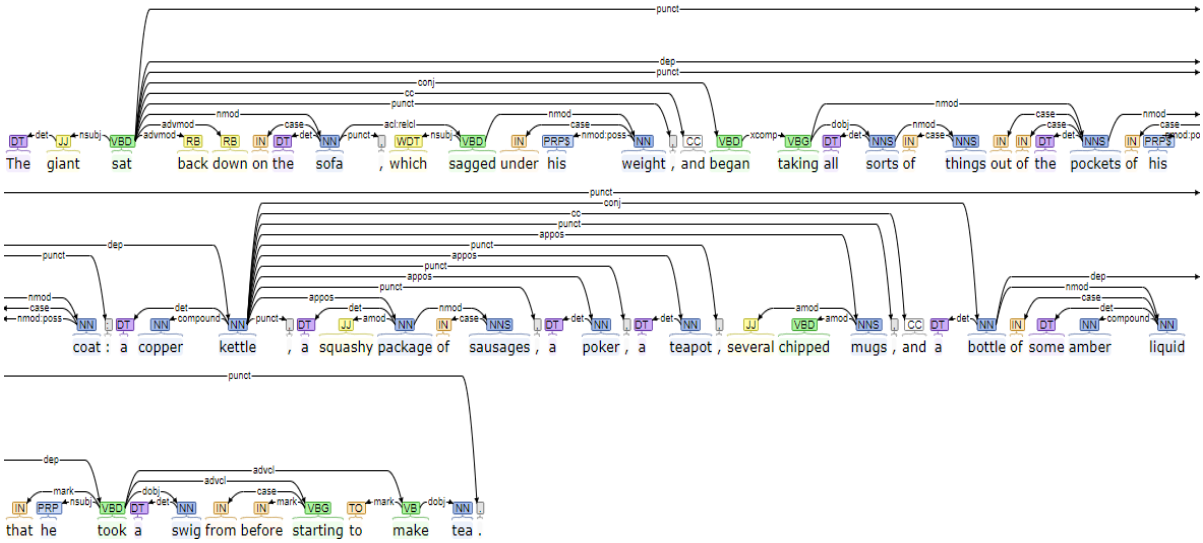
1. Parse trees do not provide information on whether something is a subject or an object
2. Parse trees are trees and their general representation is not of a graph. This makes parse trees not an intuitive format from which information should be extracted to place on a tree.

Parse trees could still have been used to extract information for visual representation, but dependency parsed outputs are more intuitive and as fast as parsing itself, providing more information about the syntactic structure of text. They are discussed in the next section. More examples of constituency parsed output can be seen in the appendix.

4.1.6 Dependency Parsing and Dependency Trees

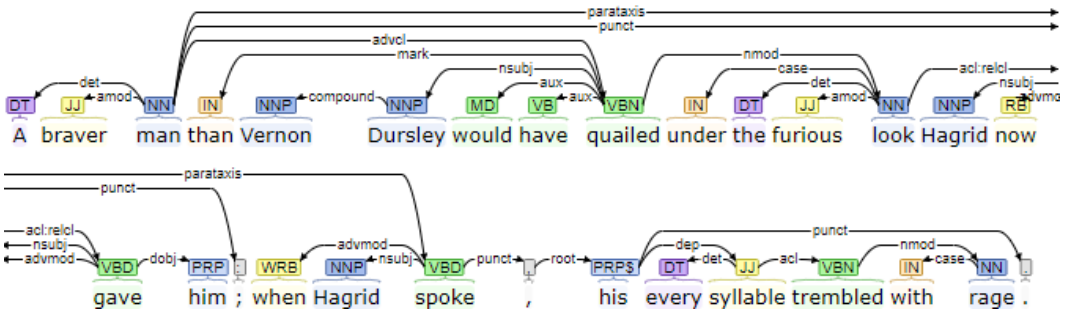
Dependency parsing is a form of parsing that provides a syntactic representation that encodes functional relationship between words. Constituency does not play a major part in dependency parsing. The relations in the dependency structure is head-dependent, NSUBJ means that the head is the nominal subject for the target of the relationship. This information is most information in information extraction, semantic parsing and question answering. Two kinds of approaches are used for dependency parsing, transition based and graph based [24]. Transition based approach uses greedy stack to get dependency structures. Graph based approach makes use of the maximum spanning tree algorithm from graph theory. Both techniques make use of supervised machine learning techniques where Treebank data is used to train these learning systems. Dependency treebanks can use human annotators or automatic transformation from phrase-structure treebanks.

We will go into more detail of dependency parsing and its output because this solution extensively makes use of the basic Stanford dependencies [25]. Basic dependencies for a tree structure. Each thing in the sentence is dependent on exactly one thing. There are no cycles in basic dependencies. List of dependency output provided by Stanford can be found in the Appendix.



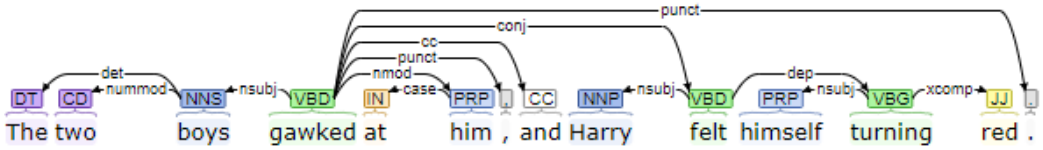
27. Dependency parsed output for "The giant sat back down on the sofa, which sagged under his weight, and began taking all sorts of things out of the pockets of his coat: a copper kettle, a squashy package of sausages, a poker, a teapot, several chipped".

Dependency parsed output provides information about of dependency between words. In this case, the dependency between “giant” and “sat” is “nsubj” representing that giant is a nominal subject with respect to sat. This information is imperative when deciding subjects for the graph. The relation between “sat” and “sofa” is nmod representing sofa as the nominal modifier for sat. This ends up becoming our object and the triple becomes giant - sat - sofa. Since we are looking forward to a more detailed extraction format, essentially not leaving any information behind, dependency information like a advmod (adverbial modifier) or simply adverb provides additional information about the action. This leaves our triple to be giant - sat (back, down) - sofa and so on [26].

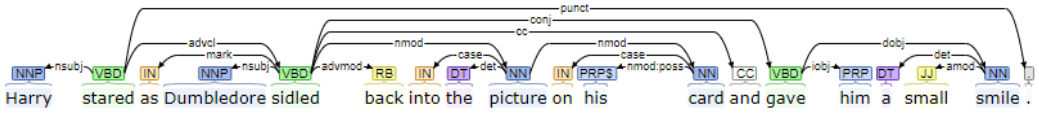


28. Dependency parsed output for "A braver man than Vernon Dursley would have quailed under the furious look Hagrid now gave him; when Hagrid spoke, his every syllable trembled with rage".

In the above sentence, a triple extraction is very hard. And even so, thinking of putting such a sentence on a graph while not losing meaning becomes not just a non-trivial exercise, it becomes a hard one. Such complex sentences are not outliers, but present in lots of different kinds of text. The information contained here, if ignored will not make a difference to the actual plot of the story as the network represents it.



29. Dependency parsed output for "The two boys gawked at him, and Harry felt himself turning red".



30. Dependency parsed output for "Harry stared as Dumbledore sidled back into the picture on his card and gave him a small smile"

The above two are much simpler sentences and outputs are “Harry - stared(at) - Dumbledore” and “Harry-feel (turn, red)-Harry”. Here since the information is about the

same entity, it becomes the subject and the object itself. More examples are present in the appendix.

4.1.7 Coreference Resolution

Coreference resolution is the process of identifying all nouns/ phrases that refer to the same entity. Coreference resolution plays the most important role in information extraction. It helps in full text understanding, machine translation, text summarization and in question answering [27]. The simplest explanation of necessity of coreference resolution is to resolve what pronoun is linked to which noun. Since parsing is done sentence wise, making connections between mentions in two different sentences becomes a problem. Let's consider:

"John Doe went to New York to get his Master's degree. He liked the place so much that he ended up staying there."

From the first sentence, information about John Doe can be extracted and about New York. but from the second sentence, it is hard to tell about who liked what and did what about it? Let's consider the ideal sentence we need to capture all that information.

"John Doe went to New York to get his Master's degree. John Doe liked New York so much that John Doe ended up staying in New York."

Considering a parsed tree from the second sentence now, the information relating to John Doe and New York and him ending up staying in New York becomes easy to extract. So coreference resolution plays a very vital role in information extraction.

As vital as its role is in IE, it is not an easy problem to solve. There are three kinds of references.

- Referring expressions:

"John Doe", "Director Doe", "Dr. Doe" and "the director of XYZ" are referring expressions referring to the same entity. These kinds of references are more common but harder to extract in practise.

- Free and bound variables:

"The dancer hurt herself." and "Jane saw her pay increase." These are more grammatically constrained, more towards linguistic theory, easier to extract in practise.

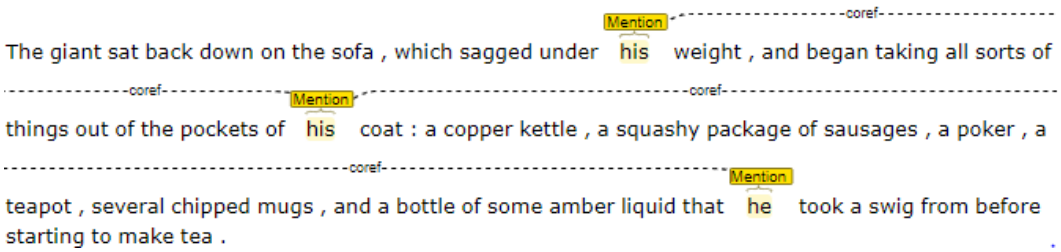
Noun phrases are not always referring. For instance, "every person for their own" or "no dancer twisted her knee".

Coreference is when two mentions refer to the same real world entity. Anaphora resolution is when a term refers to another term, the antecedent, which traditionally comes first, and the interpretation of the anaphor is determined by the interpretation of the antecedent. Anaphora mentions are not necessarily coreferential and anaphora resolution has been used for mention resolution but coreference resolution has caught up

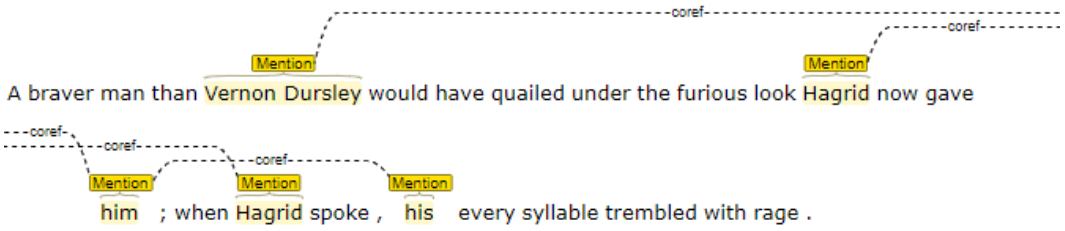
and is now the state of the art mention resolution concept, since coreference resolution models cover anaphora resolution as well. Hobb's Naive algorithm (1976) for anaphora resolution is being used for resolution but in 2013, Hector J came up with Knowledge based pronominal coreference resolution [28].

There are three kinds of coreference models. Mention pair models, mention ranking models and entity mention models. Mention repair models treat coreference chains as collection of pairwise links and reconciliation is done in a deterministic way. Mention ranking models rank, explicitly, all candidate antecedents for each mention. Entity mention models explicitly cluster mentions of the same discourse entity. Supervised mention pair models use classification techniques for binary classification tasks. For each antecedent, annotate whether a mention is a true mention or a false mention and then train the classifier. There are neural coreference models that are currently being used. Deep reinforcement learning models for mention ranking coreference by Clark and Manning, 2016, considers coreference resolution on the document level. It uses a standard feed forward neural network where the input layer is the word embeddings and a few categorical features. This reinforcement learning model makes use of the reinforce algorithm from (Williams, 1992) and a novel reward rescaling method.

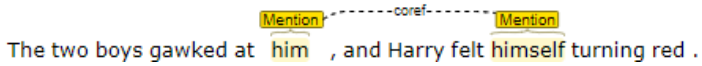
The problem with coreference resolution is that even the latest algorithms are only 66% accurate at best that becomes the leading cause of errors in information extraction.



31. Coreference resolution for "The giant sat back down on the sofa, which sagged under his weight, and began taking all sorts of things out of the pockets of his coat: a copper kettle, a squashy package of sausages, a poker, a teapot, several chipped mugs, and a bottle of some amber liquid that he took a swig from before starting to make tea".



32. Coreference resolution for "A braver man than Vernon Dursley would have quailed under the furious look Hagrid now gave him; when Hagrid spoke, his every syllable trembled with rage."



33. Coreference resolution for "The two boys gawked at him, and Harry felt himself turning red"

34. Coreference resolution for "Harry stared as Dumbledore sidled back into the picture on his card and gave him a small smile"

Coreference resolution is very important when dealing with proper nouns as nodes. When a representative pronoun is present in place of a proper noun, before dependency parsing, it is very important that it is replaced. This has not been implemented as part of this thesis, but it is a requirement for a full scale information extraction system that aims to place any kinds of nouns as nodes on networks.

CHAPTER 5

TEXT ANALYSIS TOOL

The aim of this thesis is to provide a solution to full scale text analysis and knowledge discovery in large text corpus. I have already discussed a lot of text analysis approaches that are currently being used. I propose context-free network of a specific kind along with a timeline and a heavy edge display section for understanding your text better. Such approaches are used in other kinds of data visualization and have yet to be tested full scale on text.

From the perspective of a reader, text is a time series data that has to be processed chunk by chunk by the reader and form a mental/cognitive map of the information inside the user's brain to make sense of the information contained in the text. As more text is added, the story line inside the human brain increases and the network of entity interaction also becomes bigger increasing the amount of information the text provides. I have attempted to create a tool that aims to enhance the understanding a user gets from the text, reduce reading time and improve information retention and recall.

The tool has three main components. A network, a timeline and an edge information list display. The most important feature of the tool is that it allows querying the graph but without having the user to write any queries. This requires the timeline and the graph to be interactive in the sense that the interactions represent queries. This tool is aimed at enhancing human understanding of the text.

The format in which text is extracted is done ensuring that it might be used for automated question answering in the future owing to its subject predicate object nature. The extract algorithm provides a novel approach towards general purpose information extraction. The extraction algorithm and the format are discussed in detail in their respective sections.

5.1 Extraction Algorithm

The extraction algorithm is designed to ensure that the data can be fit onto a node-edge-node model. Node-edge-node forms the basis of a graph structure. The idea here is to contain all actions and relationships between any two entities within the nodes between them and to provide tools to access required actions or relationships easily.

5.1.1 Extraction Format

The extraction format is very simple on the outside. It is a simple (to, from, relationship/action) triple but enhanced to allow the node and the edge to contain more information than just a word. The node is set to contain a list of adjectives and the ids of sentences from which each adjective comes from. The node looks like,

```
{     "entity":<entity name/identifier>,
      "nodeProperties":[
            {"property":<property/adjective/modifier>,"ids":[<sentence
      id>]}]}
```

For example, “The brown lazy fox jumped over the crazy cat.”


```
{
   "entity":"fox",
   "nodeProperties":[
      {
         "property":"brown",
         "ids":[
            1
         ]
      },
      {
         "property":"lazy",
         "ids":[
            1
         ]
      }
   ]
}
```


As properties are discovered, they are added to the node. If the same property appears again, the sentence id is added to ids. This can help track of how powerful an adjective is and understand how the sentiment regarding an entity has changed over time.

The next challenge is to contain all the information in the sentence between two entities or about one entity into an edge. The idea is to allow all actions and relationships be crudely stored in the edges. This can help in semantic role labeling later using wordnet or other such datasets (not part of the scope for this thesis). The format is presented below:

```
"connection":{
      "to": <nodeid>,
      "from": <nodeid>,
      "edgeContent":[{
                  "word":<word>,
                  "lemma":<word lemma>,
                  "edgeProperties":[
                        {"property":<modifier>,"enhancement":[<modifier
                  modifier>]},
                  "edgeContent": [<edge content>]
                  }, …]
      }
```

The extraction format consists of to, from like in all node-edge-node linkages, but the edge is recursive. Each connection has a list of edge contents, within each edge content there is a list of edge contents and so forth. This allows the format to fully grasp and represent the recursive nature of human language.

An edge can have multiple connections representing information extracted from different sentences. The extraction format is pretty complex and it requires adapters to extract information from the extraction format of the extraction algorithm to place it on network and timeline.

Let’s consider an example sentence: “Robert was going to the old town to play cricket with his friend, Ronald.”, (node id of Robert: 1, node id of Ronald: 2)


```
{
   "to":2,
   "from":1,
   "edgeContent":[
      {
         "word":"going",
         "lemma":"go",
         "edgeProperties":[
            {
               "property":"town",
               "enhancement":[
                  "old"
               ]
            }
         ],
         "edgeContent":[
            {
               "word":"play",
               "lemma":"play",
               "edgeProperties":[
                  {
                     "property":"cricket"
                  },
                  {
                     "property":"friend",
                     "enhancement":[
                        "his"
                     ]
                  }
               ]
            }
         ]
      }
   ]
}
```


This is extracted into a simpler format:

(Robert()) going(town(old) play(cricket friend(his))) (Ronald())

This format captures the nature of word dependencies without keeping the actual dependency information that might not be important for the visualization perspective, but the hierarchy of connections is very important in determining the importance of words in a narrative. It is in a format where the dependency hierarchy is maintained. This will be discussed in detail in the visualization section.

5.1.2 Prerequisites

The extraction algorithm extracts the information from basic Stanford dependencies hence making Stanford dependencies a prerequisite. I have used the Stanford CoreNLP toolkit provided by Stanford for the algorithm. This can however be implemented on any dependency parser. The extraction algorithm and the format are dependent on the hierarchy of dependencies rather than the actual dependency between words. It does not care much about what the dependencies are except when it has to ignore a certain dependency.

5.1.3 Algorithm

The extraction algorithm is a recursive modified DFS algorithm that requires Stanford dependencies as an input. The algorithm traverses through the graph and extracts the words placing them into the place in the format. The recursive nature of Extract-Edge makes it a perfect match for the extraction format, as you go deeper into the dependencies, the “EdgeContent” part of the extraction format becomes deeper.

```
Extract-Edge (graph, root, ignoredDependencies):
        for dependency in root.dependencies:

                if dependency.relation.contains(["subj", "appos"]):
                        addSubjectNode(graph, dependency)

                else if dependency.relation.contains(["obj", "mod"]):
                        addObjectNode(graph, dependency)

                else if dependency.relation.contains(["neg", "acl", "dep", "xcomp"]):
                        addEdgeContent(graph, dependency, ignoredDependencies)

                else if not dependency.relation.contains(ignoredDependencies):
                        Extract-Edge(graph, dependency, ignoredDependencies)
        }
```

35. Edge extraction algorithm (simplified)

Functions “addSubjectNode” and “addObjectNode” are very similar in terms of extraction, except that they place to information at a different part of the output. The function addEdgeContent is a recursive algorithm that goes deeper into the chosen dependency and extracts information till the leaf and calls on the Extract-Edge method to compare dependencies and extract accordingly.

```
addEdgeContent (graph, node, ignoredDependencies):

        graph.addContent(node.content)

        for dependency in node.dependencies:
                Extract-Edge(graph, dependency, ignoredDependencies)
        }
```

36. Edge content extraction (simplified)

This is a very simplified version if the algorithm used in the demo. This algorithm captures the gist of the algorithm, and its recursive nature and how it starts at the root of dependencies and explores each edge and node until it has extracted every piece of information contained in the text. Nodes are extracted in a very similar fashion, with further constraints to what constitutes as a node in the given scenario and what doesn’t.

5.2 Visualization Tool

The visualization is done using a network, a timeline and a list. These entities are different ways of showing different kinds of information that can be viewed in harmony to make almost complete sense of the text. Three key points were kept in mind when designing this text analytics tool.

1. All purpose
2. Intuitively visualize-able/understandable
3. Query-able

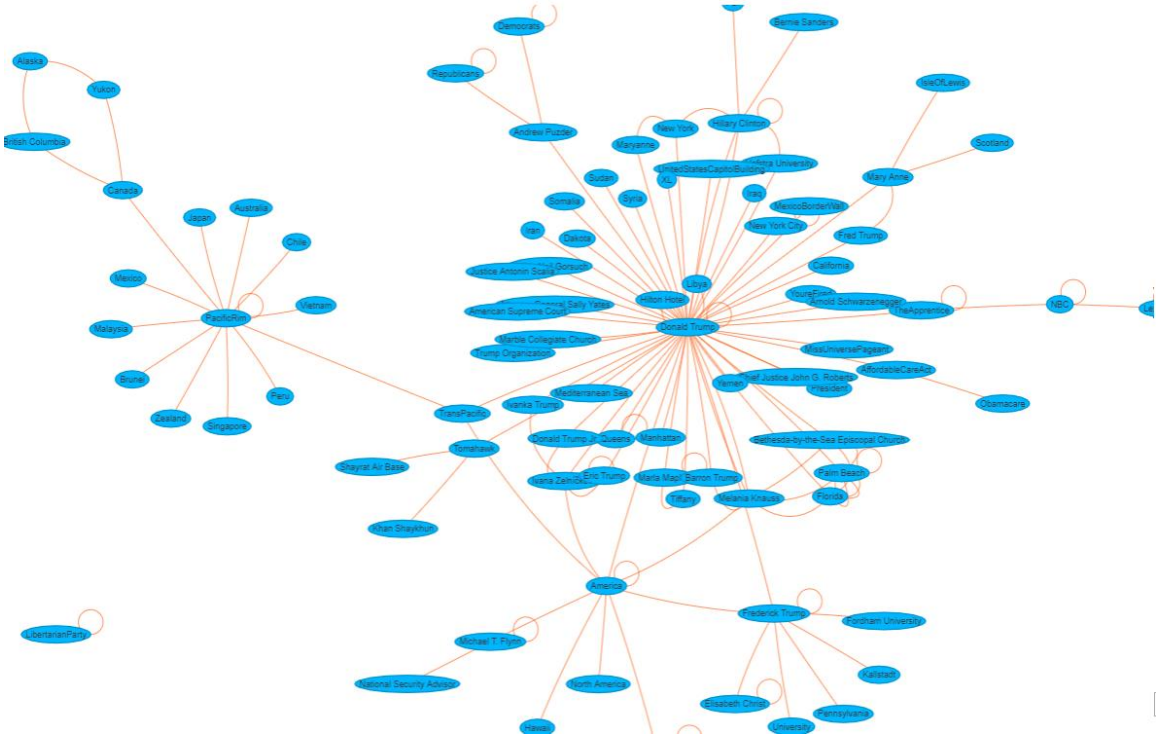
Let us keep these things in mind and discuss design decisions of the tool in detail.

5.2.1 Network

The network is built on top of the vis.js network library and uses its functions to define new functionalities on top of the network to make it more interactive and engaging. Almost all the text analysis tools that use network for text analysis are non-interactive static images that try to represent a basic idea of what the text contains.

Previously we have discussed three kinds of networks when dealing with text analysis networks. Cooccurrence networks, dependency/constituency networks and context networks. These networks are good in giving a general idea of the presence of ideas and key concepts in the text, but a more deeper understanding of the text is not possible with any of the techniques.

The network part of this tool consists of named entities as nodes and all other kinds of information in the edges. The adjectives in the text defining those nodes become part of the node's description and do not make it in the edges. All the adjectives defined for each node can be seen by hovering over the node. The edges just show that there is a connection between any two nodes, but the information contained by these nodes are displayed in a list form which can be seen when the edge is clicked on.



37. Graph generated from a 2000-word essay about United States and its current President.

The network itself is very simple and intuitive and shows how entities mentioned in the text are connected.

This can be further enhanced by color coding the information contained in the text and having distinct colors for nodes representing locations, person, organization and

miscellaneous nodes to get a more detailed information about the text more intuitively. The implementation of such small enhancements is trivial and not part of the thesis. The focus is on the format of extraction and display, and how the data can be manipulated via visual queries.

Since the network is interactive, there are certain queries that can be done just by interacting with the network alone. Special attention has been paid to make the interactions intuitive and simple and as fast as possible. The use of keyboard and extra clicks far away from the network itself has been minimized. When an edge is clicked on, only that edge remains on the map and every other edge disappears and the network focuses on the nodes this edge connects. The information is displayed separately for the edge in the order it has appeared in the text.

A node can be clicked on selecting all the edges connected to that node. To query those nodes, the user clicks on the network background. This will remove all other edges from the graph and leave only the edges for the chosen nodes to give the user more clarity over the content of the network regarding those nodes. More information about querying the network is provided in the querying section.

5.2.2 Timeline

We have already discussed that text can be perceived as a time series data where each word is a data point as it appears to the reader of the text. Since this perception of

text as a time series data is intuitive to the human brain, we can use time series data visualization techniques to help humans understand text in a better way.

There are two approaches to representing text in a timeline. We will discuss both in detail but only show results of implementation of one.

1. Information as text reveals it:

   This is the order in which a piece of information appears in the text. This is the order in which a user perceives information. Showing a timeline in the order in which it appears in the text, gives a visual representation of the text in the same way as text does, but more enriched and appealing to the reader than just reading plain text. This also helps the user put the information in the perspective of the whole text more easily and intuitively and perform analysis from multiple angles as new information comes into the view. This will be discussed in further detail when discussing querying methods for timelines.
2. Information with respect to a specific actual time defined in the text itself [29]:

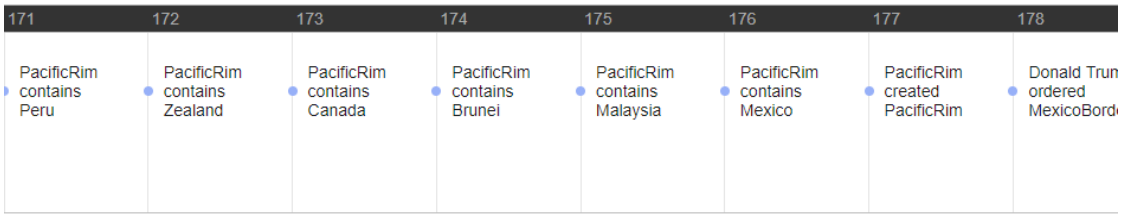
   This approach revolves around putting events on a time scale. This is ideally the best form of timeline that can be generated from text giving a better understanding of the text to the user, but due to the complexity of timeline generation for all kinds of text renders it too costly for our purpose. Phrases like “In the earlier days” or “once upon a time” are hard, or even impossible for even humans to correctly place on a timeline, it is an untrivial problem for computers

to perform by the information given in the text which can lead to incomplete information being placed on the timeline.

This approach can be useful in history books where time is very objective and unambiguous (found in [30]) or for stories where accuracy of the timeline generated is not of paramount importance to the learner.

For this thesis we have explored the use and advantages of display of data on a timeline as it appears in the text.

The purpose of the timeline in the entire system is not to provide a subsystem that can explain to the user, meaning of the text, but in a way, help to query the network that acts as the main knowledge disbursement unit. It is an interactive way of querying the parts of network the use might want to see and focus on at any given time.



38. An actual timeline generated by the software.

The unit of this scale is the number at which the connection/edge information generated. The information displayed is minimal and only contains three pieces of information per unit of timeline. It displays the (entity, “a word”, entity) triple. The word discussed here is syntactically the most important word in the entire sentence. This will not reveal all the information that is contained in the text, but it does give an idea of what part of the text is being discussed and the entities being discussed in that connection.

Syntactic importance here is measured by the number of out-nodes in the dependency tree. Since there is only one purpose this timeline exists, querying of the network, the design of the querying using the timeline is as simple as clicking on two places of the timeline and it will leave the network with only the information that is between the two chosen points.

5.2.3 Edge Information

Edge information is all the interaction that has occurred between two entities. The step words are removed, and the information is displayed in plain text. This part has not been worked upon in detail. The information extracted is contained in a recursive format and right now it is displayed using brackets surrounding words. The deeper into the sentence a word is, the more brackets surround that word. To understand this better, let us look at a few examples.

(America()) country (second (largest)) (North America())
(America()) made (states (50) district (federal) territories (five)) (America())

39. Edge information for "America is the second largest country in North America. America is made up of 50 states, a federal district, and five territories".

(America()) has (influence (great finance (world) trade culture military politics technology)) (America())
(America()) republic (federal) (America())
(America()) consists (states (50) territories (5) district (1 called)) (WashingtonDC())

40. Edge information for "America has great influence over world finance, trade, culture, military, politics, and technology. America is a federal republic. America consists of 50 states, 5 territories and 1 district called WashingtonDC".

(Alaska()) reached passing (British Columbia())
(Alaska()) reached passing (Yukon())
(British Columbia()) part (Canada())
(Yukon()) part (Canada())
(Hawaii()) located (middle) (Pacific Ocean())
(Hawaii()) located (middle) (Pacific Ocean())
(Hawaii()) rest (far (so)) (America())
(Hawaii()) rest (far (so)) (America())
(WashingtonDC(capital capital)) district (federal split (states 1791)) (Maryland())
(WashingtonDC(capital capital)) district (federal split (states 1791)) (Maryland())
(WashingtonDC(capital capital)) district (federal split (states 1791)) (Virginia())
(WashingtonDC(capital capital)) district (federal split (states 1791)) (Virginia())
(WashingtonDC(capital)) capital (WashingtonDC(capital))

41. Edge information for " Alaska can be reached by passing through British Columbia and the Yukon. British Columbia and the Yukon are part of Canada. Hawaii is located in the middle of the Pacific Ocean and is so far from the rest of America that Hawaii can only be reached by airplane. WashingtonDC, the national capital, is a federal district that was split from the states of Maryland and Virginia in 1791".

(TransPacific()) agreement (PacificRim())
(Donald Trump()) signed (order (executive) withdrawing) (America())
(PacificRim()) contains (also) (Chile())
(PacificRim()) contains (also) (Vietnam())
(PacificRim()) contains (also) (Japan())
(PacificRim()) contains (also) (Singapore())
(PacificRim()) contains (also) (Peru())
(PacificRim()) contains (also) (Zealand())
(PacificRim()) contains (also) (Australia())
(PacificRim()) contains (also) (Brunei())
(PacificRim()) contains (also) (Malaysia())
(PacificRim()) contains (also) (Mexico())
(PacificRim()) created (zone (free-trade) percent (40 (about) economy (world))) (PacificRim())
(Donald Trump()) signed (order (executive) withdrawing) (TransPacific())
(TransPacific()) agreement (America())
(PacificRim()) contains (also) (Canada())

42. Edge information for "On January 23, 2017 Donald Trump signed the executive order withdrawing America from TransPacific. TransPacific is a trade agreement between PacificRim and America. PacificRim also contains Australia. PacificRim also contains Chile, Japan, Peru, Singapore and Vietnam. PacificRim also contains Brunei, Canada and Zealand. PacificRim also contains Malaysia and Mexico. PacificRim would have created a free-trade zone for about 40 percent of the world's economy"

Edge information is in the format

(Subject(SubjectProperty(PropertyEnhancement)))

(ConnectionEntity(ConnectionEntity)) (Object(ObjectProperty(PropertyEnhancement))).

This is not a reading friendly format, but it gives an idea about the recursive nature of the extraction algorithm. A future work can be done on the linear display of the edge information bar of the proposed software.

5.2.4 Covering up for undecipherable information

The edge information display section has not been designed in a user-friendly manner and hence it is sometimes hard for the user to understand the information provided. To cover up for that, a feature has been added to make sure that the user does not get misguided. This feature allows the user to click on any of the edge information connections and the actual sentence it was parsed from is also displayed. This will help in resolving any ambiguous information in the edge information section.

Sometimes there are sentences that can't be parsed into a visualize-able format. This approach also helps in information that is not easily visualize-able. Let's consider a few examples. In the sentence "Adam likes Eve", the visualization is simple:

(Adam)-likes-(Eve)

In more complex sentences like "Adam likes to go to the park to play with Eve", visualization can be

(Adam)- likes - go - park - play -(Eve)

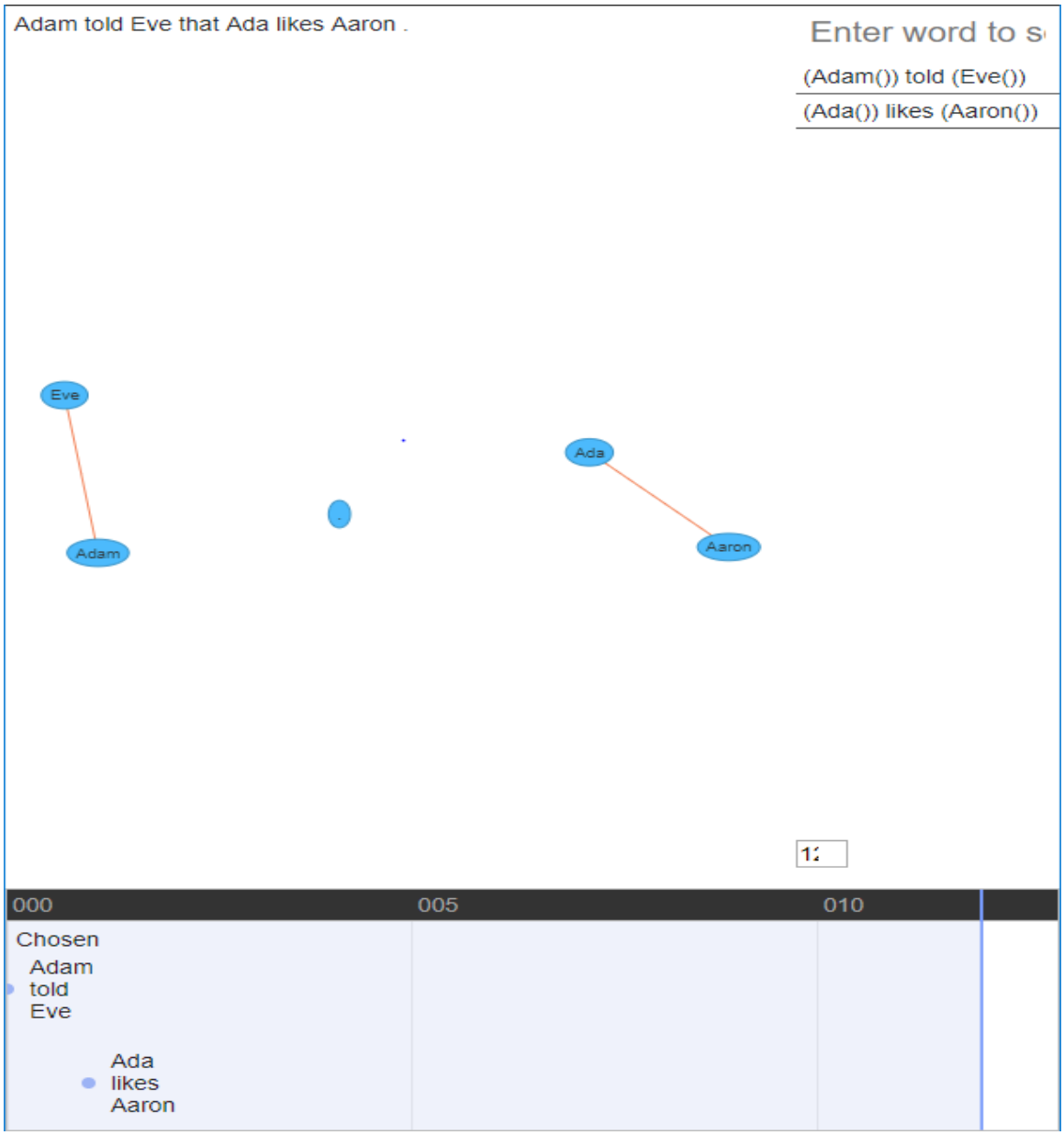
An even more complex sentence, "Adam told Eve to play in the garden with Aaron" can be visualized as

(Adam)- told -(Eve)- play - garden -(Aaron)

But consider this configuration, "Adam told Eve that Ada likes Aaron"

(Adam)- told -(Eve) (Ada)- likes -(Aaron)

Here there are two node-edge-node connections and there is a disconnect between the two and one piece of information, even though dependent on each other, do not seem to be connected at all and will create confusions for the person visualizing the text using a graph. Currently we are presenting such information as shown below



43. Network, Timeline and Edge Information for "Adam told Eve that Ada likes Aaron".

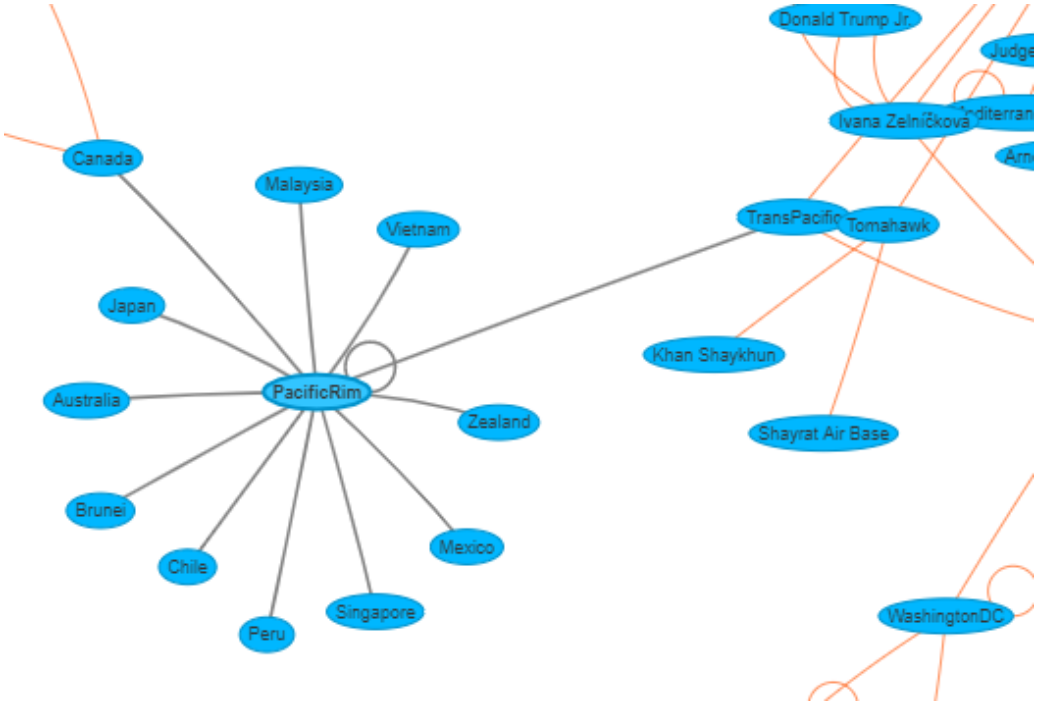
5.3 Querying

We have discussed visual querying of information. Since this tool is meant for fast information extraction and helps support quick learning and revision exercises of text,

writing queries would be inconvenient. All modules of the prescribed software are interactive and actions with them are essentially queries.

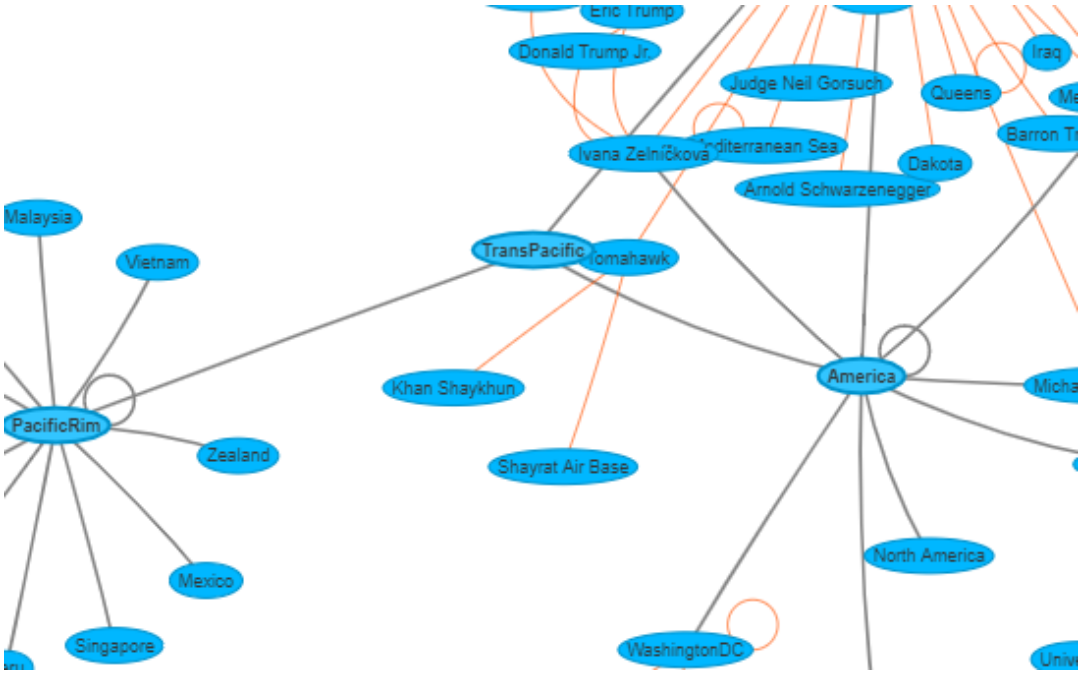
5.3.1 Network Queries

Network can be queried by clicking on the nodes and the edges. When a node is clicked on, all associated edges are chosen.



44. Interactive Querying for node "PacificRim".

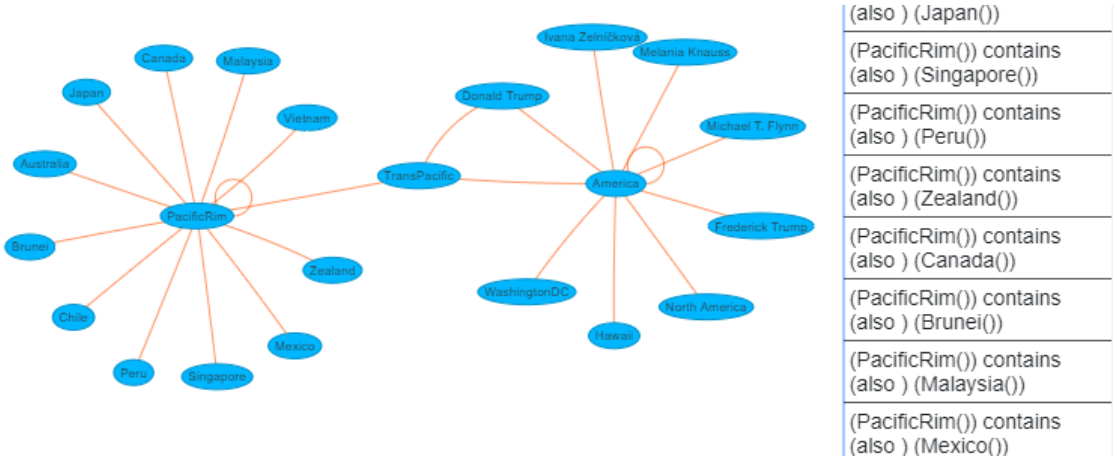
Multiple nodes can be chosen and all associated edges to the chosen nodes become part of the query. In the image below, TransPacific, America and PacificRim are chosen.



45. Interactive query for multiple nodes.

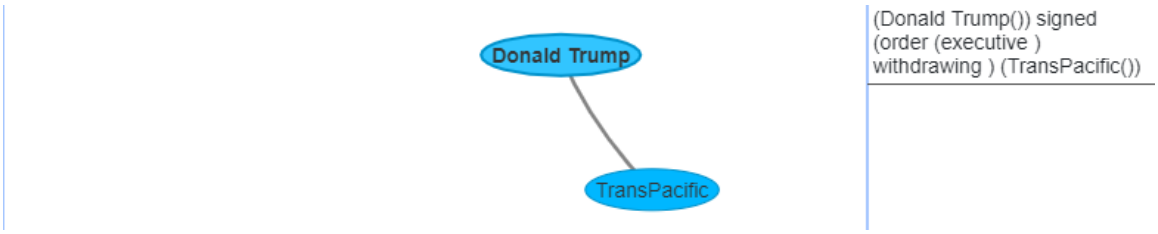
A node query can help with understanding the interactions of one entity with every other entity in the given text. When a node query is done, only the edges connecting the node with all other nodes they have interacted with remain. This helps in determining the influence of a given node.

Once the background is clicked, the query is activated. Query activation means that the information contained within the chosen edges are displayed in the edge information bar that is on the right side of the network area.



46. After the query is run.

If an edge is clicked on, the edge is chosen, and query is run right away, with the side bar displaying all the information present in that edge.

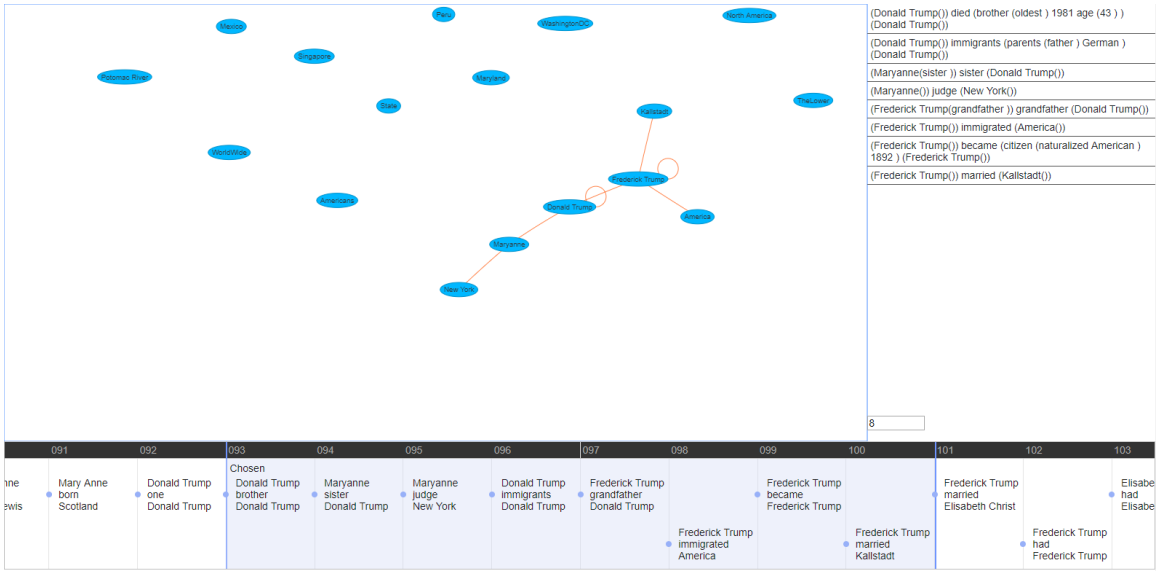


47. Query on an edge.

A use-case of such a query are of immense importance because it allows the user to visualize and understand the interactions between two nodes. Interactions between two nodes are all the actions that have happened between the nodes in the entire timeline of the text. This is important when analyzing the variance of sentiment between two characters as the story progressed. It can help with relationship analysis etc.

5.3.2 Timeline Queries

Timeline queries are simple. Timeline helps in choosing a part of text that needs to be displayed. The timeline has the granularity to the level of information extracted from any given sentence. Hence, the timeline can be used to query the network between any two given connections. An example query is shown below



48. A simple timeline query.

Two parts of the timeline were clicked on and the area between the chosen parts are queries and displayed on the edge information bar as well as the network. The area chosen also changes its background color to light blue to show the chosen area. It is helpful in focusing on a part of the text and comparing relationship between entities as they change in two different parts of the text.

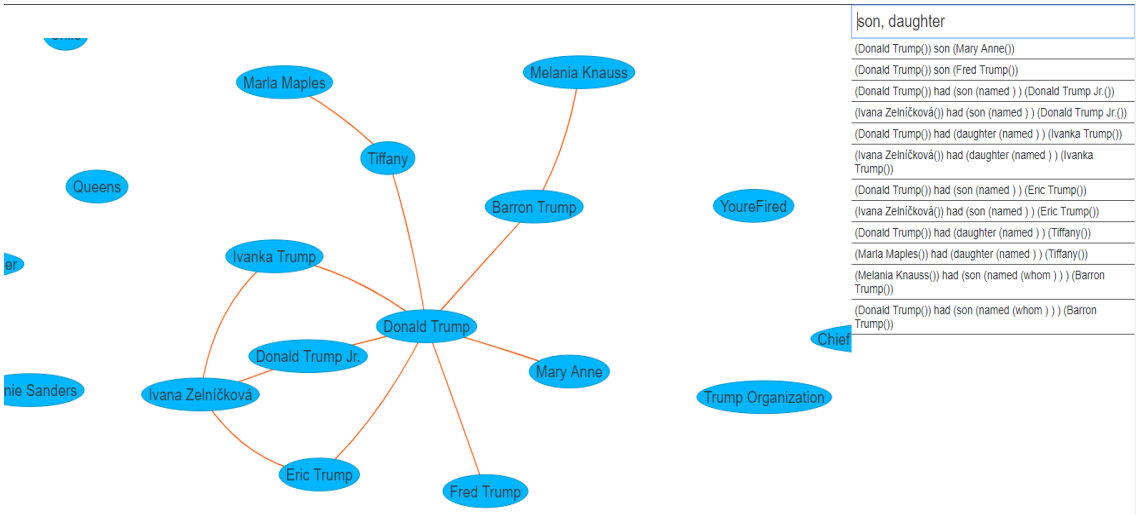
5.3.3 Story Mode Querying

The user can use the story mode by pressing the arrow keys. Forward arrow key moves the story forward and the back-arrow key moves the story back by n connections. It helps the user to iterate through the text in a linear fashion as they would do if they were reading the text. It can help with quickly moving through the story, visualizing the connections as they form.

Since this type of querying requires all modules of the software user interface, showing examples with images will make the content unreadable. This is an interesting feature and it has yet to be tested whether it will help increase reading and understanding speed.

5.3.4 Search Query

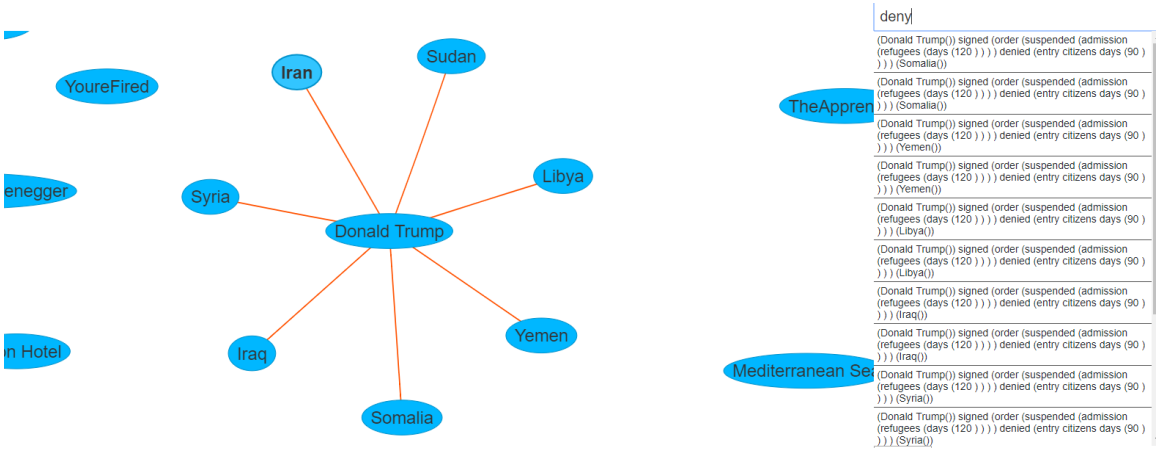
A simple search is implemented to help with determining relationships and actions, whether they exist between two entities or not. This is a very basic feature albeit a very important one. Let’s look at a few examples.



49. A simple search query on "son, daughter".

Searching for “son” and “daughter” has displayed all connections with the words present. This shows promising knowledge discovery for students, researchers and analysts when looking for specific relationships in the text they need to understand and analyze.

Another interesting result from the standard chosen text is the word “deny”.



50. A simple search query on "deny" showing countries whose residents were banned entry as an executive order signed by Donald Trump.

This shows all the countries that were denied entry into the US by the first few executive orders given by Donald Trump.

CHAPTER 6

ANALYSIS

Analysis of a software which does not have a prior is rather difficult. This analysis is qualitative rather than quantitative. This work is a proof-of-concept for a system that facilitates textual learning through visual representation. This section discusses what this system achieves, its short-comings, comparison with a similar system for another purpose and its use-cases.

6.1 Achievements of the system

The system boasts a few basic achievements that are novel. It provides a general-purpose information extraction solution which aims to put all information in a prescribed recursive all-purpose format while relating the information to the entities who were subject and object of the extracted information. It helps associate a piece of information with who it is about or who it is for. In a visualization system, such information can be placed associated with the entities they are attached to. This approach is helpful in certain real-life scenarios that are very common to people who go through a lot of text daily.

This system supports the use of cognitive load theory (Sweller, J., Ayres, P., Kalyuga, S.: Cognitive load theory. Springer, New York (2011)) in hiding information at the front end unless the user wants to see that information while providing the user with a system of queries for drilling down deeper into the information pool. Cognitive load theory argues that due to limited short term memory of human beings, in order to increase

learning and understanding and help “connect the dots” more easily, instructional design should be such that it does not overflow short term memory. If a certain idea has a connection to another idea in the text, it should be ensured that the information between these ideas do not overload the short-term memory of the learner, thus making it hard for the learner to grasp the connection.

This system does exactly that. The user interface prescribed and created as a proof of concept lets the user understand the key connections present in the text just by a glance and then helps them “dig” deeper for more knowledge and understanding of how things have happened over time and how relationships have changed over time.

6.2 Problems with the current system

The current system is developed on top of pre-existing language parsing systems. This makes the system as accurate, at best, as the system underneath it. The current implementation totally depends on the parsed outputs of a tagger and a parser. This system currently uses the best and the fastest syntactic parser, which is Stanford’s dependency parser. POS Tagger provided by Stanford is also used to determine what entities should go to the nodes and for marking the nodes with different colors if they represent a different kind of entity. Even though the system uses the best parser, it is only 92% accurate according to the developers of the Stanford dependency parser [31].

According to the Stanford POS tagger web page [32], the tagged output is 97% accurate. Since our algorithm is directly dependent on both the systems, the overall

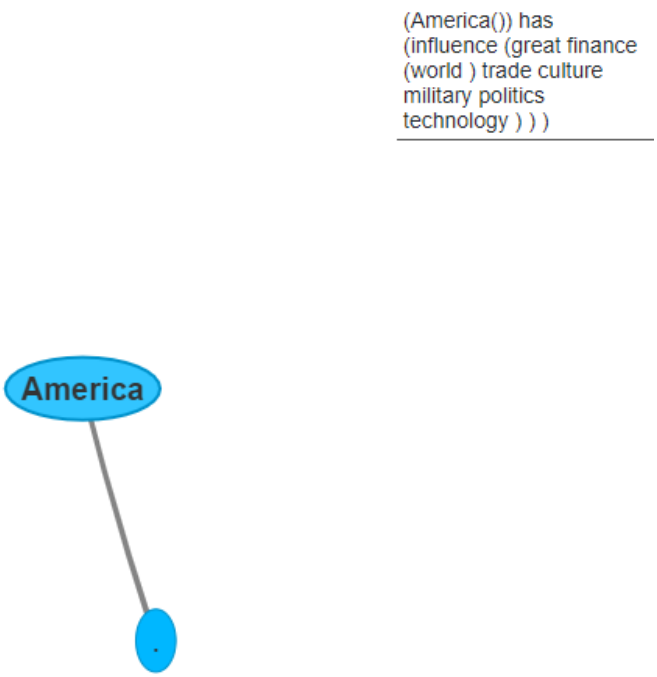
accuracy of the current system is reduced to 89%. This accuracy does not include the missing coreference resolution module. The best coreference resolution module present currently is only 60% accurate [33]. This greatly affects the purpose of the software as coreference resolution is a very important future module of the system for it to become a widely used information visualization and understanding system. This leaves the system only about 54% accurate where the accuracy is only mostly affected by the coreference resolution system.

The use of coreference resolution in the current system is only for one task: to replace the pronouns with its representative proper nouns. The low accuracy of coreference systems is owed to it being the most frequently occurring disambiguation problem. Since our purpose is much simpler and we do not care about non-pronoun mentions of proper nouns, we are looking at much higher accuracies even with current coreference resolution systems. This is discussed further in the future works section.

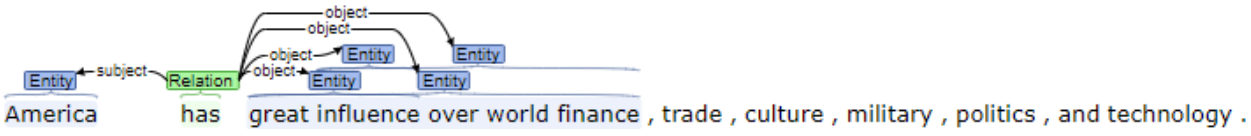
6.3 Comparison with Stanford Open Information Extraction

This section deals compares the output of the proposed system against Stanford OpenIE [34], that is not designed for the purpose but can arguably be used for presenting information on a network. Since OpenIE does not provide the format to match a node-edge-node model in the way we do, additional processing has to be performed on OpenIE output to get it to desired format. OpenIE also uses dependencies to extract the information that it does, hence our system is presumably faster than the OpenIE extraction plus the adaptation algorithm to adapt to the prescribed format. Let us look at a

few examples of how the OpenIE system has parsed information and how our system has. The examples chosen are from a Wikipedia article about America [35] and are changed to adapt to the coreference resolution problem to get best results.

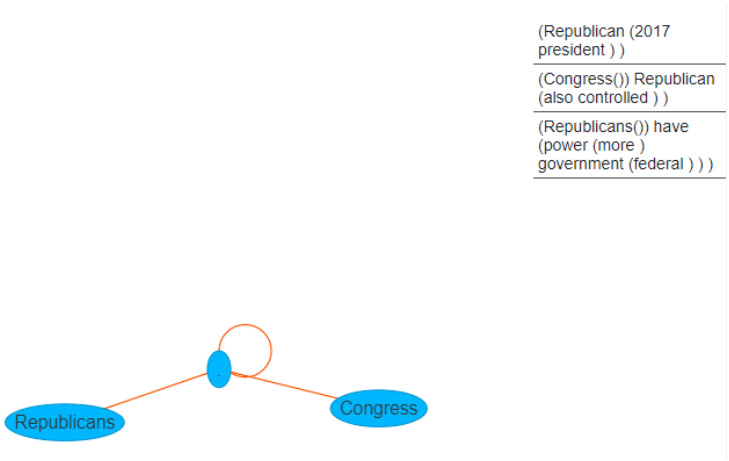


51. Our output for "America has great influence over world finance, trade, culture, military, politics, and technology."

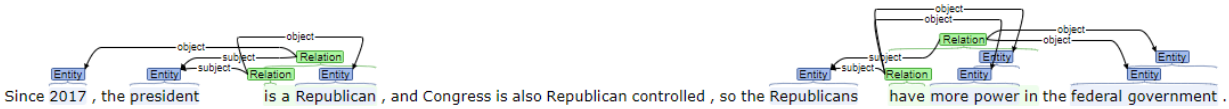


52. Stanford OpenIE output for "America has great influence over world finance, trade, culture, military, politics, and technology."

Our system, in this example, has successfully extracted the information about America's influence on world finance, trade, culture and so on while the OpenIE system could not correctly identify trade, culture, military, politics and technology as seen above.



53. Our output for "Since 2017, the president is a Republican, and Congress is also Republican controlled, so the Republicans have more power in the federal government".

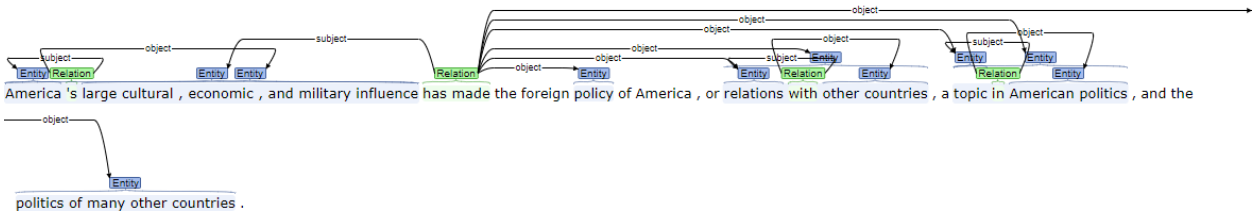


54. Stanford OpenIE output for "Since 2017, the president is a Republican, and Congress is also Republican controlled, so the Republicans have more power in the federal government".

In this example, a key piece of information that the OpenIE system failed to extract was that Congress is Republican controlled, but the proposed system does it well.

(made (influence (large cultural economic military) policy (foreign) relations (countries (other)) topic (politics (American)) politics (countries (many other))) (America())

55. Our output for "America's large cultural, economic, and military influence has made the foreign policy of America, or relations with other countries, a topic in American politics, and the politics of many other countries".



56. Stanford OpenIE output for "America's large cultural, economic, and military influence has made the foreign policy of America, or relations with other countries, a topic in American politics, and the politics of many other countries".

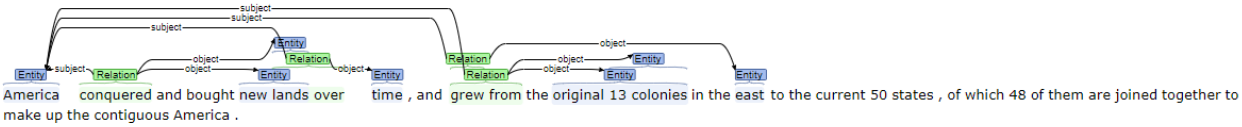
For the sentence above, the OpenIE system and the proposed system have both extracted all the information provided in the text.

(America(contiguous contiguous)) conquered (lands (new) time) grew (colonies (original 13) east (states (current 50) joined (which together make (up)))) (America(contiguous contiguous))

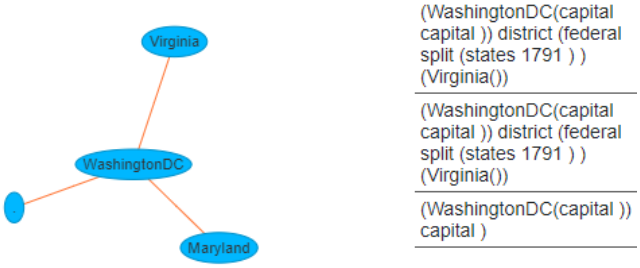
(America(contiguous contiguous)) conquered (lands (new) time) grew (colonies (original 13) east (states (current 50) joined (which together make (up)))) (America(contiguous contiguous))

(grew (colonies (original 13) east (states (current 50) joined (which together make (up)))) 48 (them) (America(contiguous))

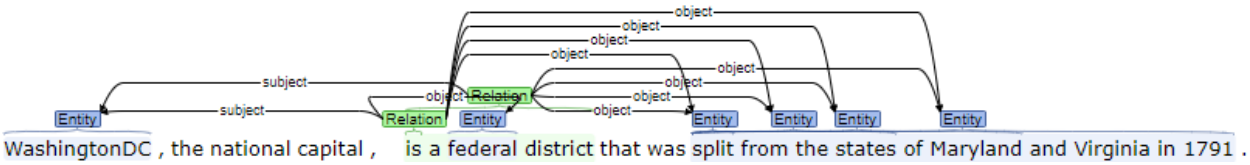
57. Our output for "America conquered and bought new lands over time, and grew from the original 13 colonies in the east to the current 50 states, of which 48 of them are joined together to make up the contiguous America."



58. Stanford OpenIE output for "America conquered and bought new lands over time, and grew from the original 13 colonies in the east to the current 50 states, of which 48 of them are joined together to make up the contiguous America."



59. Our output for "WashingtonDC, the national capital, is a federal district that was split from the states of Maryland and Virginia in 1791"



60. Stanford OpenIE output for "WashingtonDC, the national capital, is a federal district that was split from the states of Maryland and Virginia in 1791"

In the above example, the OpenIE system failed to extract the information that WashingtonDC is a capital whereas our system does it and stores it as an information in the node WashingtonDC. It is important to note here that the system is trying to place all the information provided by the text in their appropriate places. OpenIE does not currently explicitly deal with adverbs and adjectives which are often very important in defining an entity and in defining an action. There are a lot of such examples and a few more are placed in the appendix for further comparison.

We have analyzed three different kinds of text from various sources to understand the nature of information extraction and its efficiency in these major areas. We have tried

to take a large enough text and run the algorithm on it to analyze the effect on it. Each text is large enough to contain as varied combinations of sentences as possible. Three categories are history text, fiction novel text and biography text. History text is from Wikipedia's World War II [36] page. Fiction novel is the first chapter of "A walk to remember". Biography text is Wikipedia's Donald Trump page [37]. Since the software is currently not resolving coreferences, the text has to be curated. The curated text can be found in Appendix X for all three types that was used in both the systems.

The OpenIE system was designed to solve the KBP slot filling problem. It extracts the same information in various formations as extracted above. All relations extracted that contains the main verb/action is counted as one for the purpose of this analysis. The purpose of our algorithm is to extract one information only once to put in a graph to help the user visualize the information correctly. The format explained works perfectly to achieve that end, whereas the OpenIE algorithm has to make sure it is not losing any slots it has to fill. In Table 1, correct actions extracted by our algorithm is considered total number of relationships/actions extracted minus the repeated number of relations/actions.

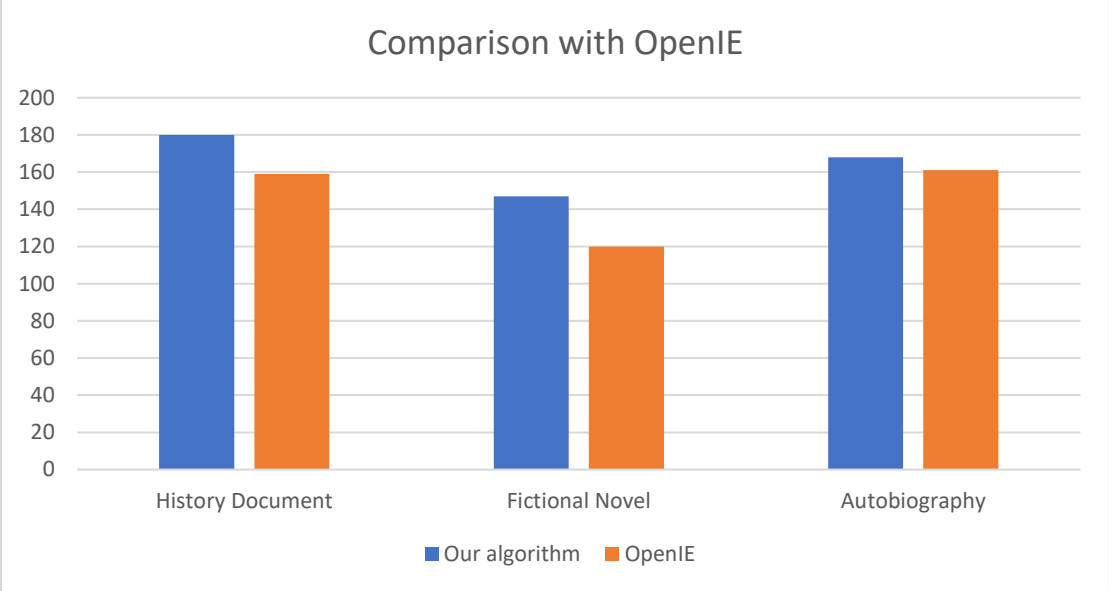
To be able to verify these results in Table 1, the exact text used for this analysis can be found in Appendix X.

For our purpose, this analysis is important to show that we were able to extract more relationships, from large enough text (2000+ words) in different categories, than the OpenIE system. These examples have enough variety and length to be a representative of much larger samples as used by OpenIE to perform the KBP slot filling task. As shown in

the earlier analysis with very specific examples, relationships/actions that were completely missed by the OpenIE system are also captured by our system.

Document Type	Our algorithm	OpenIE
History Document	180	159
Fictional Novel	147	120
Autobiography	168	161

Table 1. Number of correct relationships extracted



61. Comparison with OpenIE

The analysis presented in Table 1 and Figure 61 can be extrapolated to a much larger texts and can be shown to have extracted all the information that is marked correctly by the dependency parser. It is important to note that the time taken is dependent on the time the

Stanford dependency parser takes to parse text. The algorithm itself is a simple graph traversal algorithm on the Stanford output. The correct nodes are determined as nodes that are actual entities present in the test. Entities here can be human individuals, organizations and places. Actions are marked as correct if an action/relationship is found by the extraction algorithm that exists in the text and can be understood by reading the text.

It is also important to note that the entities found must be marked by the named entity recognizer by Stanford. The algorithm's efficiency directly depends on the efficiency of the parser. It is also important to note that the most important purpose of the software is to convert text into a format which is most feasible for visualization in a network as opposed to question answering or other kinds of text analysis. However, importance analysis and centrality analysis can be performed on the graph to find out other kinds of metrics generally associated with text analysis.

CHAPTER 7

CONCLUSION AND FUTURE WORKS

This system is the first of its kind. An end to end basic set of recommended features is implemented and argued upon their effectiveness. This tool can be used in teaching curriculums that require a lot of reading for instance economics, business studies, history, literature etc. The features in the tool have use-cases in the analysis industry where analysts have to read a lot of news articles and understand the effect news have on stock prices etc.

This system can also be used to parse legal documents and present to the signatories so that they understand the clauses presented in the future in a much better way. This is a very interesting future work that can be worked upon and is highly required. This tool is the first of its kind and can help as a facilitator for teachers teaching students about a certain text. They can refer to the interactive network and timeline in order to help explain the content of the text they mean to deliver to an audience. For students studying, this tool can be a revision assistant. Once they have read through the entire text, it can help them search for information.

For certain types of text understanding where analysis on entities and analysis of relationships is important like in business analysts analyzing news articles or literature students analyzing actions of a certain character throughout the story, this tool provides all the necessary information in one place.

Future works, with respect to the system, is working on a pre-processing pipeline that does coreference resolution and pronoun replacement with their representative proper nouns. Even though the current taggers and parsers have inaccuracies, there are certain ways in which they can be cheated on to improve accuracy. If a person who already understands the text provides a list of entities they would like to see on the network, they can be replaced by proper nouns that the parser always recognizes correctly and then converted back to their original names once the parsing is complete, in the parsed output. This almost certainly ensures that the parsed information is accurately parsed. This helps because, if a book is parsed correctly once, it can be used by anyone who needs a reference point for that book and does not need to parse that text again and again. An online library of books parsed through this system, or a visual Wikipedia which as all the text curated and parsed through the current system, will help users reach their desired information fast and accurately which is intuitively understandable quickly. With the amount of text future humans will be reading, a system that helps them visualize and understand the text faster.

This system can also be merged with semantic role labeling systems to mark nodes with their semantic labels. Standardized color coding schemes can be developed to ensure that colors on networks unanimously represent the same information across the globe. A professor at ASU suggested to use the system to teach students how to write more coherently and then use the system to test for coherence. Another professor requested the system to be developed to parse news articles to see how relationship

between bio-fuel companies have changed over time to see whether it can predict which ones are more sustainable than others.

There is a lot of improvement that can be done on the Edge Information section. It can be reorganized and made more presentable. This research can make use of linguists and psychologists to determine the best way of presenting textual information for very quick and accurate understanding without having to read the entire text. This extraction algorithm can also be enhanced, the extraction format can be made more detailed and the visualization can be made much better. Other query methods can be added and animations on network and timeline. This can be the future of human understanding of information instead of the 2000 years old way, reading one-dimensional text.

REFERENCES

[1] S. C. A. &. P. P. Vijaygaikwad, "Text Mining Methods and Techniques," *International Journal of Computer Applications*, pp. 85(17), 42-45, 2014.

[2] "CS.HAIFA.AC.IL," [Online]. Available: http://cs.haifa.ac.il/~shuly/teaching/08/nlp/complexity.pdf.

[3] P. Savyanavar and B. Mehta, "Multi-Document Summarization Using TF-IDF Algorithm," *International Journal Of Engineering And Computer Science.*

[4] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead and O. Etzioni, "Open information extraction for the web," *International Joint Conference on Artificial Intelligence*, vol. 7, pp. 2670-2676, 2007.

[5] G. Zhou, J. Su, J. Zhang and M. Zhang, "Exploring various knowledge in relation extraction," *Association for Computational Linguistics*, pp. 427-434, 2005.

[6] "https://planningtank.com/computer-applications/data-visualization-importance-techniques-tools," [Online].

[7] "BBC," [Online]. Available: http://www.bbc.com/news/business-26383058.

[8] C. D and H. Schutze, Foundations of Statistical Natural Language Processing, Cambridge (Mass.), London: MIT press, 1999.

[9] F. Moretti, "Network Theory, Plot Analysis," *New Left Review*, vol. II, no. 68, pp. 80-102, 2011.

[10] D. Paranvushkin, "Identifying the Pathways for Meaning Circulation Using Text Network Analysis," *Nodus Labs*, 2011.

[11] "JMLR," [Online]. Available: http://www.jmlr.org/papers/volume1/heckerman00a/html/node15.html.

[12] "Senereko," [Online]. Available: https://static.ceres.rub.de/legacy/uploads/senereko/hnrws15/

diesner.context_hnrws2015.pdf.

[13] "Stanford NLP," [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html.

[14] J. Webster and C. Kit, *Tokenization as the initial phase in NLP,* City Polytechnic of Hong Kong, 1992.

[15] "NLP Stanford," [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

[16] "Bitext," [Online]. Available: https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/.

[17] P. Schachter, "Parts-of-speech Systems," *Language Typology and Syntactic Description,* vol. 1, pp. 3-61, 1985.

[18] M. P. Marcus, B. Santorini and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn treebank," *Computational Linguistics,* vol. 19, no. 2, pp. 313-330, 1993.

[19] J. Kupiec, "Robust part-of-speech tagging using a hidden Markov Model," *Computer Speech and Language,* vol. 6, pp. 225-242, 1992.

[20] "Stanford," [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/10.pdf.

[21] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/List_of_metonyms.

[22] C. Samuelsson and M. Wiren, "Parsing Techniques," pp. 59-91.

[23] G. Dick and J. Ceriel, "Parsing Techniques A Practical Guide," *Ellis Horwood Limited,* pp. 64-66, 1990.

[24] "Stanford," [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/14.pdf.

[25] M.-C. d. Marneffe and C. D. Manning, "The Stanford typed dependencies representation," *Workshop on Cross-framework and Cross-domain Parser Evaluation,* 2008.

[26] A. Zouaq, M. Gagno and O. Benoit, "Semantic Analysis using dependency-based grammars and upper-level ontologies," *International Journal of Computational Linguistics and Applications,* vol. 1, pp. 1-2, 2010.

[27] "Stanford," [Online]. Available: https://web.stanford.edu/class/cs224n/archive/WWW_1617/lectures/cs224n-2017-lecture15.pdf.

[28] CS.Toronto.edu. [Online]. Available: http://www.cs.toronto.edu/~hector/Papers/ijcai-13-paper.pdf.

[29] "TimeLineCurator," [Online]. Available: http://www.cs.ubc.ca/group/infovis/software/TimeLineCurator/.

[30] "ACLWeb," [Online]. Available: http://www.aclweb.org/anthology/W17-5912.

[31] [Online]. Available: https://cs.stanford.edu/~danqi/papers/emnlp2014.pdf.

[32] [Online]. Available: https://nlp.stanford.edu/software/pos-tagger-faq.html.

[33] [Online]. Available: https://nlp.stanford.edu/software/dcoref.shtml#Questions.

[34] G. Angeli, M. J. Premkumar and C. D. Manning, "Leveraging Linguistic Structure For Open Domain Information".

[35] "United States," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/United_States.

[36] "World War II," [Online]. Available: https://en.wikipedia.org/wiki/World_War_II.

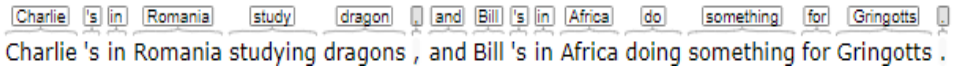
[37] "Donald Trump," [Online]. Available: https://en.wikipedia.org/wiki/Donald_Trump.

APPENDIX A

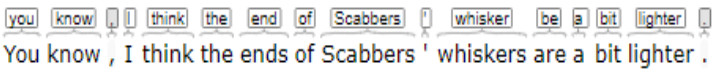
ADDITIONAL EXAMPLES OF SOFTWARES USED

I. ADDITIONAL EXAMPLES OF TOKENIZATION

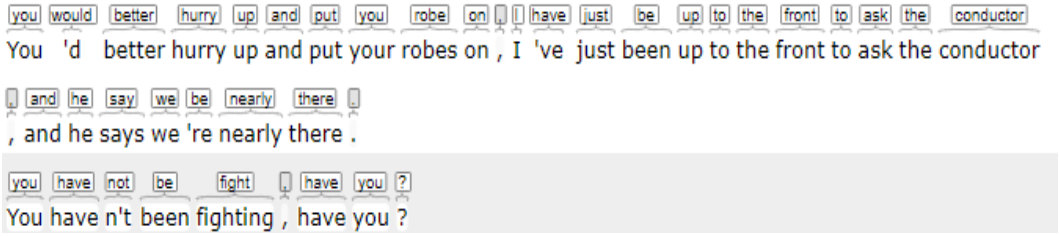
1. Charlie's in Romania studying dragons, and Bill's in Africa doing something for Gringotts.



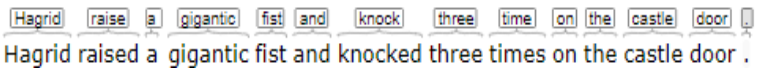
2. You know, I think the ends of Scabbers' whiskers are a bit lighter.



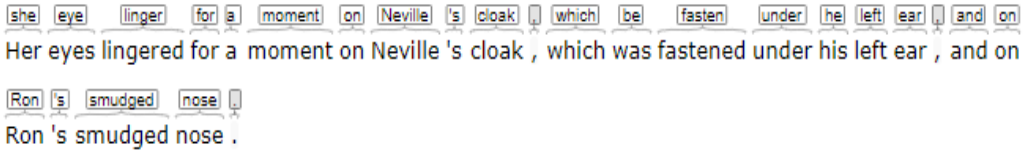
3. You'd better hurry up and put your robes on, I've just been up to the front to ask the conductor, and he says we're nearly there. You haven't been fighting, have you?



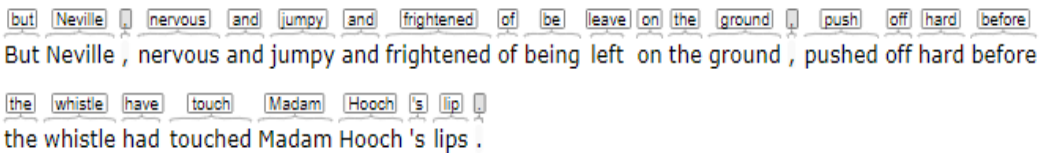
4. Hagrid raised a gigantic fist and knocked three times on the castle door.



5. Her eyes lingered for a moment on Neville's cloak, which was fastened under his left ear, and on Ron's smudged nose.

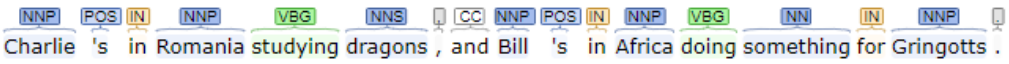


6. But Neville, nervous and jumpy and frightened of being left on the ground, pushed off hard before the whistle had touched Madam Hooch's lips.

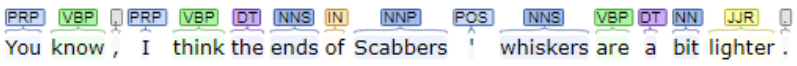


II. ADDITIONAL EXAMPLES OF PARTS OF SPEECH TAGGING

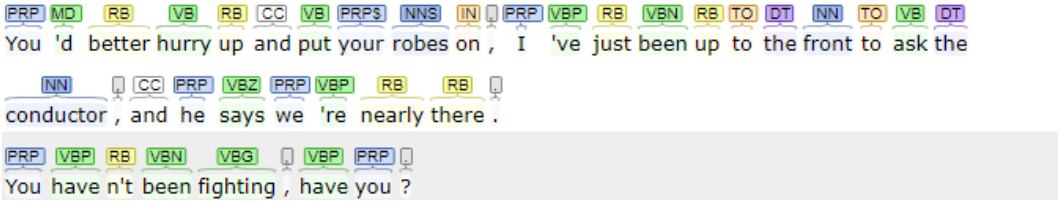
1. Charlie's in Romania studying dragons, and Bill's in Africa doing something for Gringotts



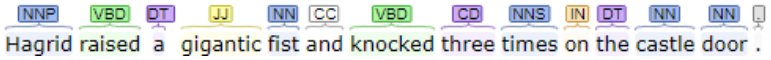
2. You know, I think the ends of Scabbers' whiskers are a bit lighter.



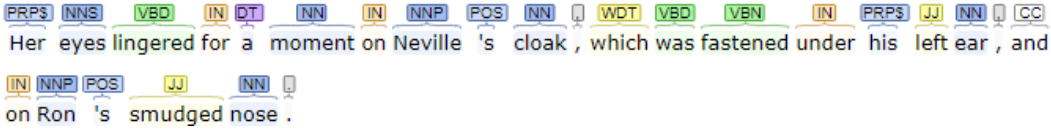
3. You'd better hurry up and put your robes on, I've just been up to the front to ask the conductor, and he says we're nearly there. You haven't been fighting, have you?



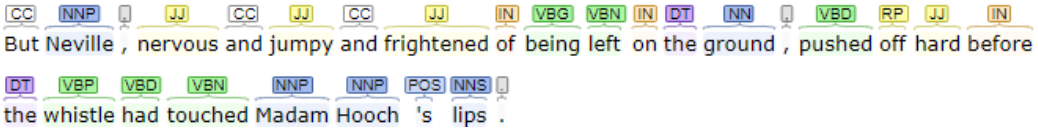
4. Hagrid raised a gigantic fist and knocked three times on the castle door.



5. Her eyes lingered for a moment on Neville's cloak, which was fastened under his left ear, and on Ron's smudged nose.

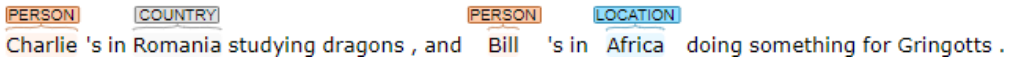


6. But Neville, nervous and jumpy and frightened of being left on the ground, pushed off hard before the whistle had touched Madam Hooch's lips.

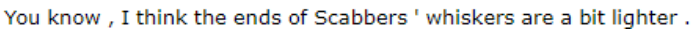


III. ADDITIONAL EXAMPLES OF NAMED ENTITY RECOGNITION

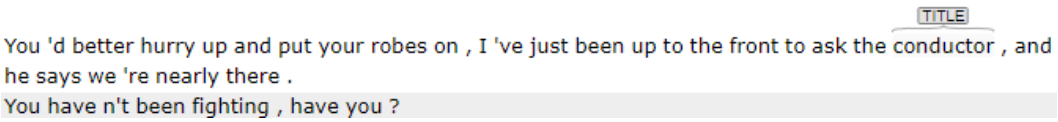
1. Charlie's in Romania studying dragons, and Bill's in Africa doing something for Gringotts



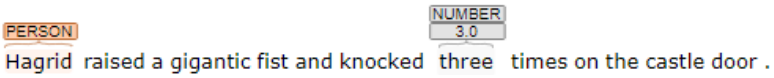
2. You know, I think the ends of Scabbers' whiskers are a bit lighter.



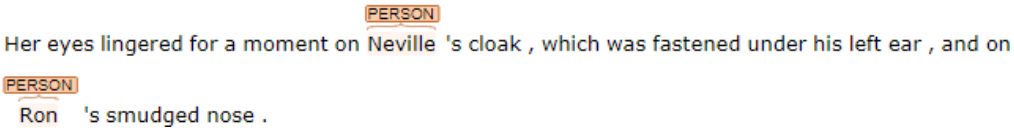
3. You'd better hurry up and put your robes on, I've just been up to the front to ask the conductor, and he says we're nearly there. You haven't been fighting, have you?



4. Hagrid raised a gigantic fist and knocked three times on the castle door.



5. Her eyes lingered for a moment on Neville's cloak, which was fastened under his left ear, and on Ron's smudged nose.



6. But Neville, nervous and jumpy and frightened of being left on the ground, pushed off hard before the whistle had touched Madam Hooch's lips

PERSON

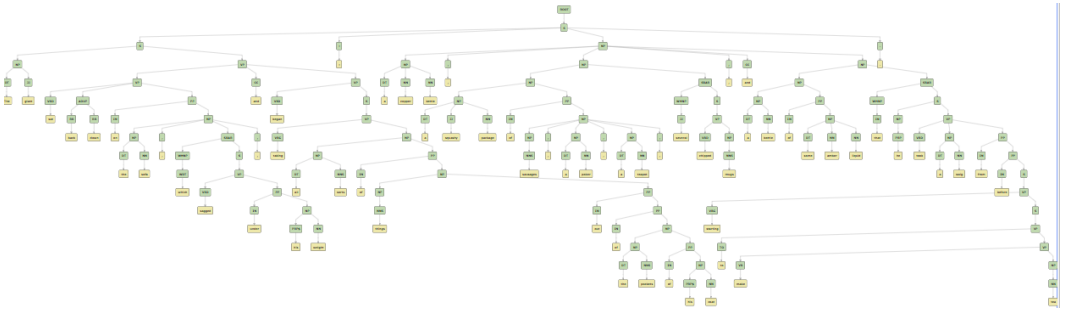
But Neville , nervous and jumpy and frightened of being left on the ground , pushed off hard before

PERSON

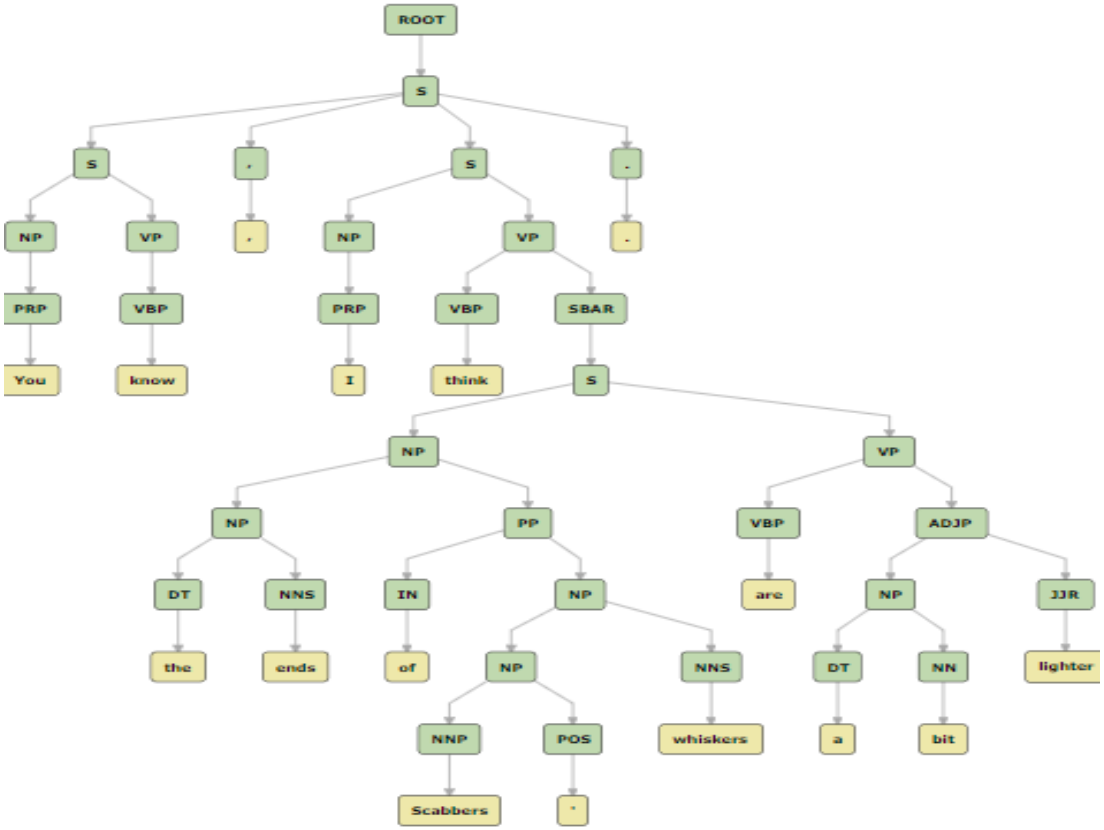
the whistle had touched Madam Hooch 's lips .

IV. ADDITIONAL EXAMPLES OF CONSTITUENCY PARSING

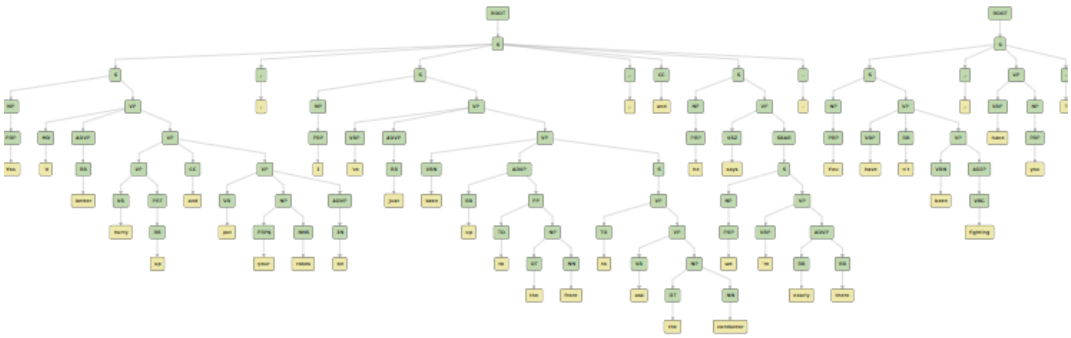
1. The giant sat back down on the sofa, which sagged under his weight, and began taking all sorts of things out of the pockets of his coat: a copper kettle, a squashy package of sausages, a poker, a teapot, several chipped mugs, and a bottle of some amber liquid that he took a swig from before starting to make tea.



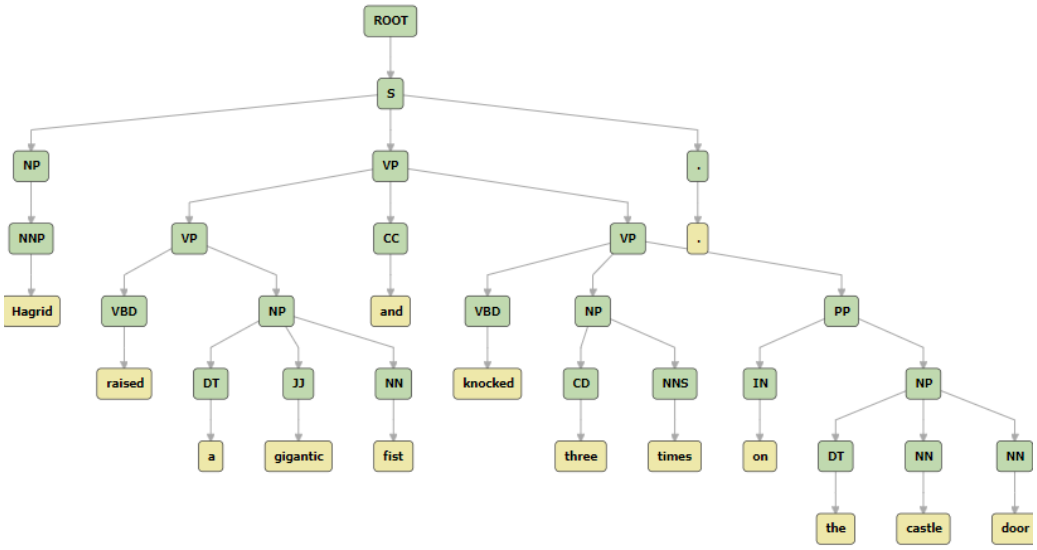
2. You know, I think the ends of Scabbers' whiskers are a bit lighter.



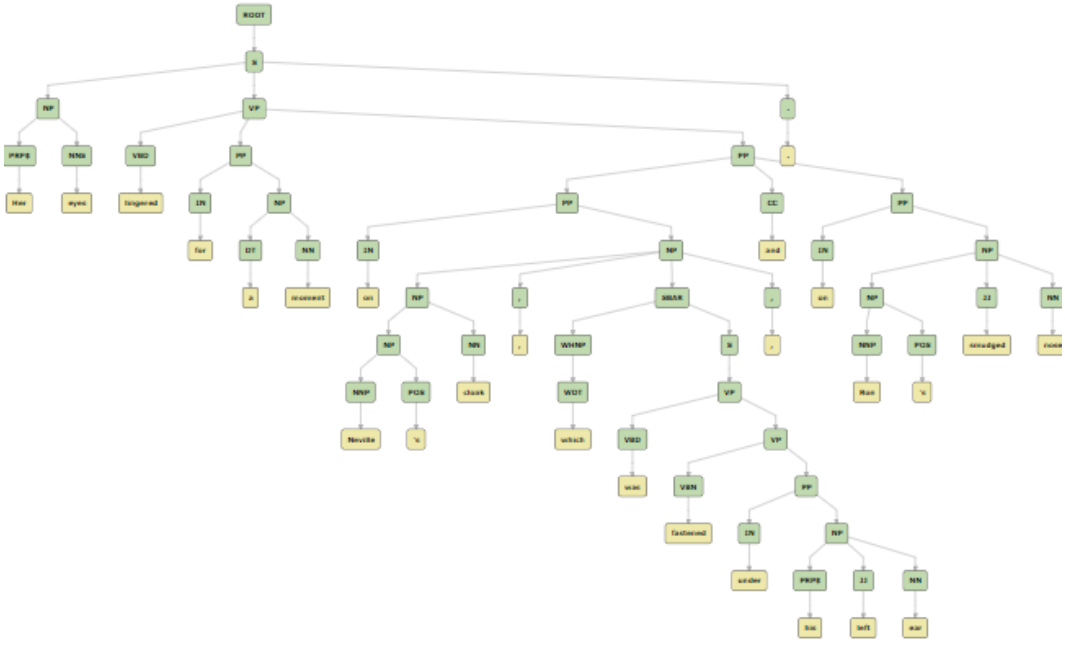
3. You'd better hurry up and put your robes on, I've just been up to the front to ask the conductor, and he says we're nearly there. You haven't been fighting, have you?



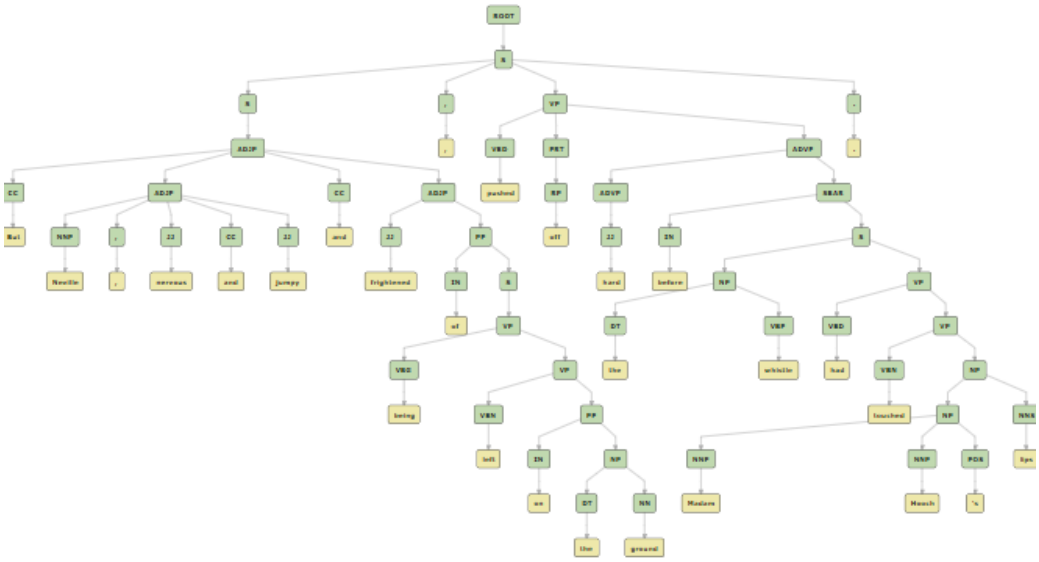
4. Hagrid raised a gigantic fist and knocked three times on the castle door.



5. Her eyes lingered for a moment on Neville's cloak, which was fastened under his left ear, and on Ron's smudged nose.



6. But Neville, nervous and jumpy and frightened of being left on the ground, pushed off hard before the whistle had touched Madam Hooch's lips.


ROOT
S
S
,
VP
.
ADJP
,
VBD
PRT
ADVP
,
CC
ADJP
CC
ADJP
pushed
RP
ADVP
SBAR
But
NNP
,
JJ
CC
JJ
and
JJ
PP
off
JJ
IN
S
Neville
,
nervous
and
jumpy
frightened
IN
S
hard
before
NP
VP
of
VP
DT
VBP
VBD
VP
VBG
VP
the
whistle
had
VBN
NP
being
VBN
PP
touched
NP
NNS
left
IN
NP
NNP
NNP
POS
Eyes
on
DT
NN
Madam
Mooch
's
the
ground

V. TYPED DEPENDENCIES FROM STANFORD USED FOR INFORMATION EXTRACTION

ACOMP: adjectival complement (object of the verb)
ADVCL: adverbial clause modifier (adverbial clause that modifies the verb)
ADVMOD: adverb modifier (adverb that modifies the verb)
APPOS: appositional modifier (definer/modifier of the noun)
AUX: auxiliary (main non-verb of the verb clause)
AUXPASS: passive auxiliary (main passive non-verb of the verb clause)
CC: coordination (relation between word and its conjunction word)
CCOMP: clausal complement (dependent clause with an internal subject, object of the verb)
CONJ: conjunct (conjunction of the phrase)
COP: copula (dependent of the complement)
CSUBJ: clausal subject (subject that is itself a clause)
CSUBJPASS: clausal subject (subject that is itself a clause and also passive)
DEP: dependent (when no precise dependency is detected)
DET: determiner (relationship of head with its determiner)
DOBJ: direct object (direct object of the verb phrase)
EXPL: expletive (deals with there)
GOESWITH: goes with (two words that go together)
IOBJ: indirect object (indirect object of the verb phrase)
MARK: marker (finite clause subordinate to another clause)
MWE: multi word expression (example: because of, well as)
NEG: negation modifier (when a relationship is nullified)
NN: noun compound modifier (noun that modifies another noun, or further defines another)
NPVADVMOD: noun phrase as adverbial modifier (when a noun phrase modifies a verb)
NSUBJ: nominal subject (syntactic subject of the clause)
NSUBJPASS: passive nominal subject
NUM: numeric modifier
NUMBER: element of compound number
PARATAXIS: two sentences places side by side without a coordination/subordination
PCOMP: prepositional complement (complement of a preposition)
POBJ: object of a preposition
POSS: possession modifier (offices, their)
POSSESSIVE: possessive modifier (clothes, ‘s)
QUANTMOD: quantity phrase modifier (200, About)
XCOMP: open clausal complement
XSUBJ: controlling subject

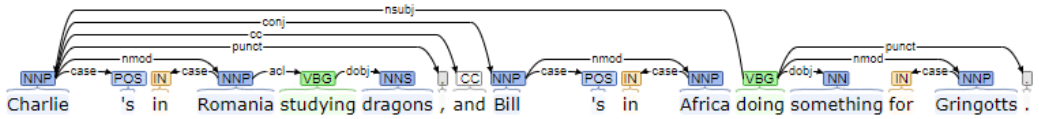
VI. HIERARCHY OF DEPENDENCIES AS PROVIDED BY STANFORD

root - root
dep - dependent
- aux - auxiliary
  - auxpass - passive auxiliary
  - cop - copula
- arg - argument
  - agent - agent
  - comp - complement
    - acomp - adjectival complement
    - ccomp - clausal complement with internal subject
    - xcomp - clausal complement with external subject
    - obj - object
      - dobj - direct object
      - iobj - indirect object
      - pobj - object of preposition
  - subj - subject
    - nsubj - nominal subject
      - nsubjpass - passive nominal subject
    - csubj - clausal subject
      - csubjpass - passive clausal subject
- cc - coordination
- conj - conjunct
- expl - expletive (expletive “there”)
- mod - modifier
  - amod - adjectival modifier
  - appos - appositional modifier
  - advcl - adverbial clause modifier
  - det - determiner
  - predet - predeterminer
  - preconj - preconjunct
  - vmod - reduced, non-finite verbal modifier
  - mwe - multi-word expression modifier
    - mark - marker (word introducing an advcl or ccomp
  - advmod - adverbial modifier
    - neg - negation modifier
  - rcmod - relative clause modifier
  - quantmod - quantifier modifier
  - nn - noun compound modifier
  - npadvmod - noun phrase adverbial modifier
    - tmod - temporal modifier
  - num - numeric modifier
  - number - element of compound number
  - prep - prepositional modifier
  - poss - possession modifier
  - possessive - possessive modifier (’s)
  - prt - phrasal verb particle
- parataxis - parataxis
- goeswith - goes with

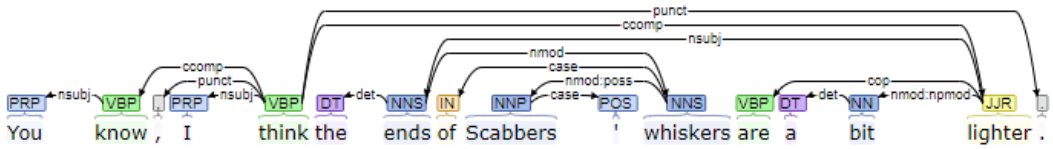
punct - punctuation
ref - referent
sdep - semantic dependent
    xsubj - controlling subject

VII. ADDITIONAL EXAMPLES OF STANFORD BASIC DEPENDENCY

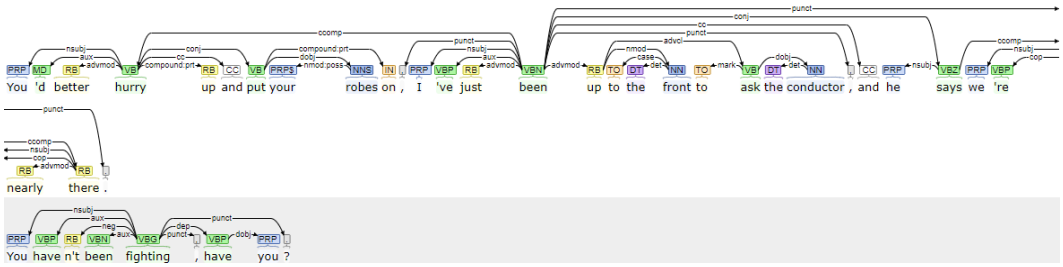
1. Charlie's in Romania studying dragons, and Bill's in Africa doing something for Gringotts



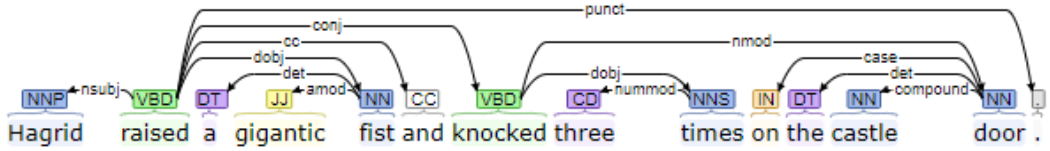
2. You know, I think the ends of Scabbers' whiskers are a bit lighter.



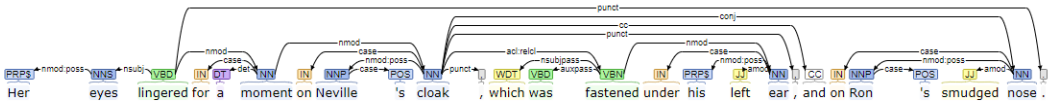
3. You'd better hurry up and put your robes on, I've just been up to the front to ask the conductor, and he says we're nearly there. You haven't been fighting, have you?



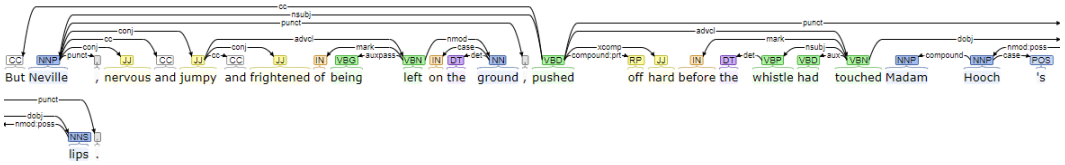
4. Hagrid raised a gigantic fist and knocked three times on the castle door.



5. Her eyes lingered for a moment on Neville's cloak, which was fastened under his left ear, and on Ron's smudged nose.



6. But Neville, nervous and jumpy and frightened of being left on the ground, pushed off hard before the whistle had touched Madam Hooch's lips.



VIII. ADDITIONAL EXAMPLES OF STANFORD COREFERENCE

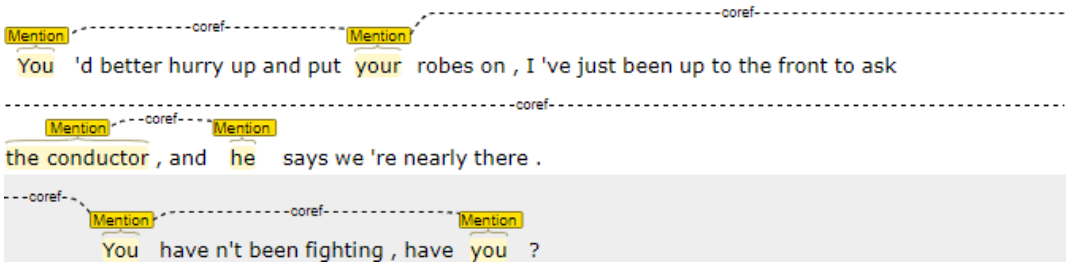
1. Charlie's in Romania studying dragons, and Bill's in Africa doing something for Gringotts

Charlie 's in Romania studying dragons , and Bill 's in Africa doing something for Gringotts .

2. You know, I think the ends of Scabbers' whiskers are a bit lighter.

You know , I think the ends of Scabbers ' whiskers are a bit lighter .

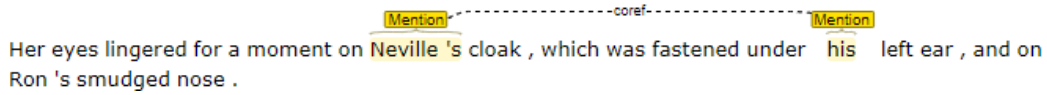
3. You'd better hurry up and put your robes on, I've just been up to the front to ask the conductor, and he says we're nearly there. You haven't been fighting, have you?



4. Hagrid raised a gigantic fist and knocked three times on the castle door.

Hagrid raised a gigantic fist and knocked three times on the castle door .

5. Her eyes lingered for a moment on Neville's cloak, which was fastened under his left ear, and on Ron's smudged nose.



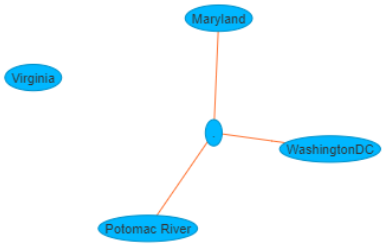
6. But Neville, nervous and jumpy and frightened of being left on the ground, pushed off hard before the whistle had touched Madam Hooch's lips.

But Neville , nervous and jumpy and frightened of being left on the ground , pushed off hard before the whistle had touched Madam Hooch 's lips .

APPENDIX B

ADDITIONAL EXAMPLES OF ALGORITHM OUTPUT

I. ADDITIONAL EXAMPLES OF OPENIE COMPARISON

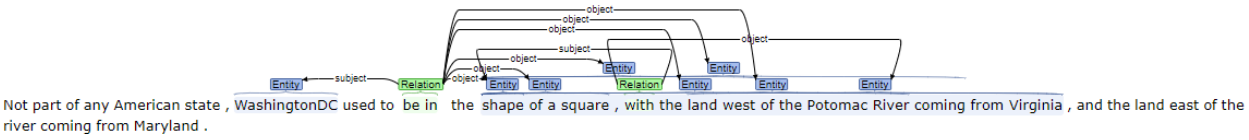


(used (part (state
(American)) shape
(square west (land) land
(east river (coming))))
(WashingtonDC())

(used (part (state
(American)) shape
(square west (land) land
(east river (coming))))
(Potomac River())

(used (part (state
(American)) shape
(square west (land) land
(east river (coming))))
(Maryland())

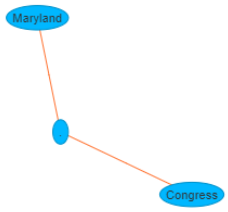
62. Our output for "Not part of any American state, WashingtonDC used to be in the shape of a square, with the land west of the Potomac River coming from Virginia, and the land east of the river coming from Maryland".



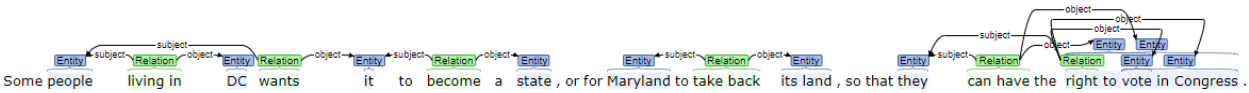
63. Stanford OpenIE output for "Not part of any American state, WashingtonDC used to be in the shape of a square, with the land west of the Potomac River coming from Virginia, and the land east of the river coming from Maryland".

(wants (people (living
(DC)) it become (state)
) (Maryland())

(take (land (its)) so
have (right (vote))
(Congress())

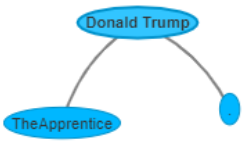


64. Our output for "Some people living in DC wants it to become a state, or for Maryland to take back its land, so that they can have the right to vote in Congress"



65. Stanford OpenIE output for "Some people living in DC wants it to become a state, or for Maryland to take back its land, so that they can have the right to vote in Congress"

(Donald Trump()) earned (year (first show) 50,000 ($ episode)) (TheApprentice())
(Donald Trump()) earned (year (first show) 50,000 ($ episode)) (TheApprentice())
(Donald Trump()) paid (success (Apprentice initial) $ (million (1) episode)))



66. Our output for "For the first year of the show, Donald Trump earned $50,000 per episode of TheApprentice, but following The Apprentice's initial success, Donald Trump was paid $1 million per episode."

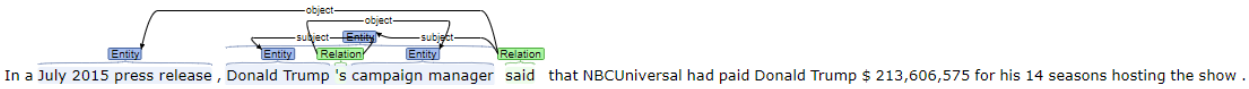
For the first year of the show , Donald Trump earned $ 50,000 per episode of TheApprentice , but following The Apprentice 's initial success , Donald Trump was paid $ 1 million per episode .

67. Stanford OpenIE output for "For the first year of the show, Donald Trump earned $50,000 per episode of TheApprentice, but following The Apprentice's initial success, Donald Trump was paid $1 million per episode."

(said (release (2015 press) manager (campaign)) (Donald Trump())

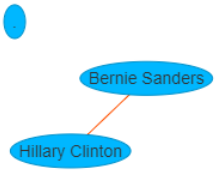
(paid (NBCUniversal 213,606,575 ($ seasons (his 14 hosting (show)))) (Donald Trump())

68.Our output for "In a July 2015 press release, Donald Trump's campaign manager said that NBCUniversal had paid Donald Trump $213,606,575 for his 14 seasons hosting the show"

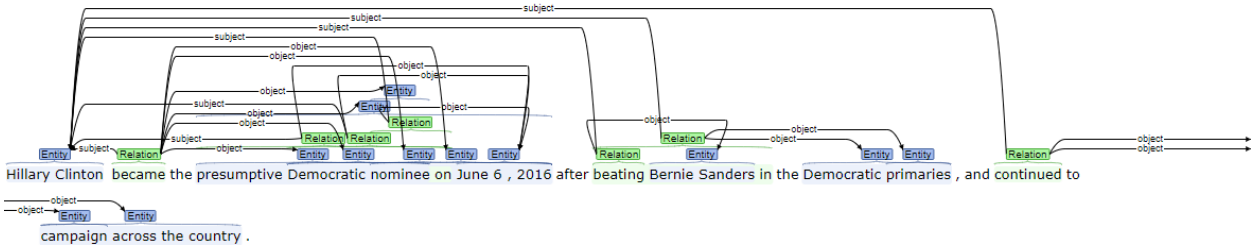


69. Stanford OpenIE output for "In a July 2015 press release, Donald Trump's campaign manager said that NBCUniversal had paid Donald Trump $213,606,575 for his 14 seasons hosting the show"

(Hillary Clinton()) became (nominee (presumptive Democratic)) beating continued (campaign country) (Bernie Sanders(primaries))

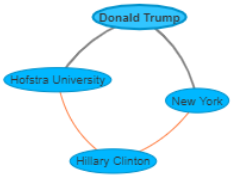


70. Our output for "Hillary Clinton became the presumptive Democratic nominee on June 6, 2016 after beating Bernie Sanders in the Democratic primaries, and continued to campaign across the country"

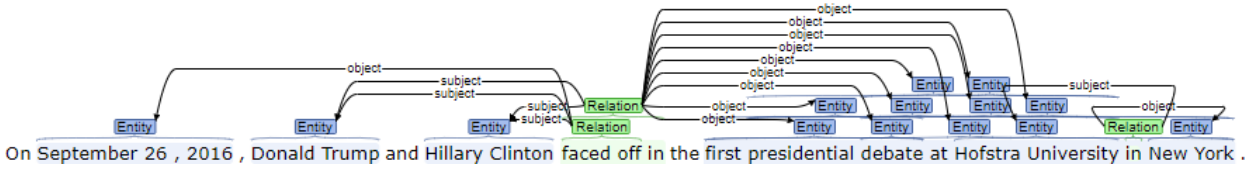


71. Stanford OpenIE output for "Hillary Clinton became the presumptive Democratic nominee on June 6, 2016 after beating Bernie Sanders in the Democratic primaries, and continued to campaign across the country"

(Donald Trump()) faced (debate (first presidential)) (Hofstra University())
(Donald Trump()) faced (debate (first presidential)) (New York())
(Hillary Clinton()) faced (debate (first presidential)) (Hofstra University())
(Hillary Clinton()) faced (debate (first presidential)) (New York())

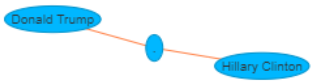


72. Our output for "On September 26, 2016, Donald Trump and Hillary Clinton faced off in the first presidential debate at Hofstra University in New York"

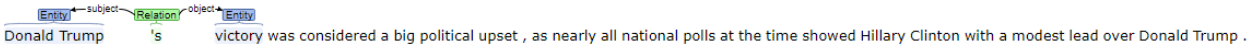


73. Stanford OpenIE output for "On September 26, 2016, Donald Trump and Hillary Clinton faced off in the first presidential debate at Hofstra University in New York"

(considered (victory upset (big political)) (Donald Trump())
(showed (polls (all (nearly) national time) lead (modest)) (Donald Trump())
(showed (polls (all (nearly) national time) lead (modest)) (Hillary Clinton())



74. Our output for "Donald Trump's victory was considered a big political upset, as nearly all national polls at the time showed Hillary Clinton with a modest lead over Donald Trump"



75. Stanford OpenIE output for "Donald Trump's victory was considered a big political upset, as nearly all national polls at the time showed Hillary Clinton with a modest lead over Donald Trump"

II. TEXT FOR OPENIE QUANTITATIVE COMPARISON

1. World War II (History Text)
   (https://en.wikipedia.org/wiki/World_War_II)

On 1 September 1939, Germany invaded Poland after having staged several false flag border incidents as a pretext to initiate the attack. The Battle of Wester Platte is often described as the first battle of the war. Britain responded with an ultimatum to Germany to cease military operations, and on 3 September, after the ultimatum was ignored, France, Britain, Australia, and New Zealand declared war on Germany. France, Britain, Australia, and New Zealand alliance was joined by South Africa (6 September) and Canada (10 September). The alliance provided no direct military support to Poland, outside of a cautious French probe into the Saarland. The Western Allies also began a naval blockade of Germany, which aimed to damage the country's economy and war effort. Germany responded by ordering U-boat warfare against Allied merchant and warships, which would later escalate into the Battle of the Atlantic. On 8 September, German troops reached the suburbs of Warsaw. The Polish counter offensive to the west halted the German advance for several days, but Polish counter was outflanked and encircled by the Wehrmacht. Remnants of the Polish army broke through to besieged Warsaw. On 17 September 1939, after signing a cease-fire with Japan, the Soviets invaded Eastern Poland under a pretext that the Polish state had ostensibly ceased to exist. On 27 September, the Warsaw garrison surrendered to the Germans, and the last large operational unit of the Polish Army surrendered on 6 October. Despite the military defeat, the Polish government never surrendered. Significant part of Polish military

personnel evacuated to Romania and the Baltic countries; many of them would fight against the Axis in other theatres of the war. The Polish government in exile also established an Underground State and a resistance movement; in particular the Polish partisan Home Army would grow to become one of the war's largest resistance movements. Germany annexed the western and occupied the central part of Poland, and the USSR annexed Poland's eastern part; small shares of Polish territory were transferred to Lithuania and Slovakia.

On 6 October, Hitler made a public peace overture to Britain and France, but said that the future of Poland was to be determined exclusively by Germany and the Soviet Union. The proposal was rejected, and Hitler ordered an immediate offensive against France, which would be postponed until the spring of 1940 due to bad weather. The Soviet Union forced the Baltic countries: Estonia, Latvia and Lithuania, the states that were in a Soviet sphere of influence, to sign, mutual assistance pacts, that stipulated stationing Soviet troops in Estonia, Latvia and Lithuania. Soon after, significant Soviet military contingents were moved there. Finland refused to sign a similar pact and rejected to cede part of Finland's territory to the USSR, which prompted a Soviet invasion in November 1939, and the USSR was expelled from the League of Nations. Despite overwhelming numerical superiority, Soviet military success was modest, and the Finno-Soviet war ended in March 1940 with minimal Finnish concessions. In June 1940, the Soviet Union forcibly annexed Estonia, Latvia and Lithuania, and the disputed Romanian regions of Bessarabia, Northern Bukovina and Hertza. Meanwhile, Nazi-Soviet political

rapprochement and economic co-operation gradually stalled, and both states, Nazi-Soviet began preparations for war. In April 1940, Germany invaded Denmark and Norway to protect shipments of iron ore from Sweden, which the Allies were attempting to cut off. Denmark capitulated after a few hours, and Norway was conquered within two months despite Allied support. British discontent over the Norwegian campaign led to the appointment of Winston Churchill as Prime Minister on 10 May 1940. On the same day, Germany launched an offensive against France. To circumvent the strong Maginot Line fortifications on the Franco-German border, Germany directed its attack at the neutral nations of Belgium, the Netherlands, and Luxembourg. The Germans carried out a flanking maneuver through the Ardennes region, which was mistakenly perceived by Allies as an impenetrable natural barrier against armored vehicles. By successfully implementing new blitzkrieg tactics, the Wehrmacht rapidly advanced to the Channel and cut off the Allied forces in Belgium, trapping the bulk of the Allied armies in cauldron on the Franco-Belgian border near Lille. Britain was able to evacuate a significant number of Allied troops from the continent by early June, although abandoning almost all of their equipment. On 10 June, Italy invaded France, declaring war on both France and the United Kingdom. The Germans turned south against the weakened French army, and Paris fell to them on 14 June. Eight days later France signed an armistice with Germany; it was divided into German and Italian occupation zones, and an unoccupied rump state under the Vichy Regime, which, though officially neutral, was generally aligned with Germany. France kept its fleet, which Britain attacked on 3 July in an attempt to prevent its seizure by Germany. The Battle of Britain began in early July with Luftwaffe attacks

on shipping and harbors. Britain rejected Hitler's ultimatum, and the German air superiority campaign started in August but failed to defeat RAF Fighter Command. Due to this the proposed German invasion of Britain was postponed indefinitely on 17 September. The German strategic bombing offensive intensified with night attacks on London and other cities in the Blitz but failed to significantly disrupt the British war effort and largely ended in May 1941. Using newly captured French ports, the German Navy enjoyed success against an over-extended Royal Navy, using U-boats against British shipping in the Atlantic. The British Home Fleet scored a significant victory on 27 May 1941 by sinking the German battleship Bismarck. In November 1939, the United States, who were taking measures to assist China and the Western Allies, amended the Neutrality Act to allow, cash and carry, purchases by the Allies. In 1940, following the German capture of Paris, the size of the United States Navy was significantly increased. In September the United States further agreed to a trade of American destroyers for British bases. Still, a large majority of the United States public continued to oppose any direct military intervention in the conflict well into 1941. In December 1940 Roosevelt accused Hitler of planning world conquest and ruled out any negotiations as useless, calling for the United States to become an, arsenal of democracy, and promoted Lend-Lease programs of aid to support the British war effort. The United States started strategic planning to prepare for a full-scale offensive against Germany. At the end of September 1940, the Tripartite Pact formally united Japan, Italy and Germany as the Axis Powers. The Tripartite Pact stipulated that any country, with the exception of the Soviet Union, which attacked any Axis Power would be forced to go to war against all three,

Japan, Italy and Germany. The Axis expanded in November 1940 when Hungary, Slovakia and Romania joined. Romania and Hungary would make major contributions to the Axis war against the USSR; in Romania's case partially to recapture territory ceded to the USSR. In December 1940 Roosevelt accused Hitler of planning world conquest and ruled out any negotiations as useless, calling for the US to become an arsenal of democracy and promoted Lend-Lease programs of aid to support the British war effort. The US started strategic planning to prepare for a full-scale offensive against Germany. At the end of September 1940, the Tripartite Pact formally united Japan, Italy and Germany as the Axis Powers. The Tripartite Pact stipulated that any country, with the exception of the Soviet Union, which attacked any Axis Power would be forced to go to war against all three. The Axis expanded in November 1940 when Hungary, Slovakia and Romania joined. Romania and Hungary would make major contributions to the Axis war against the USSR. In Romania's case partially to recapture territory ceded to the USSR.

In early June 1940 the Italian Regia aeronautics attacked Malta, and a siege of this British possession started. In late summer – early autumn, Italy conquered British Somaliland and made an incursion into British-held Egypt. In October Italy attacked Greece, but the attack was repulsed with heavy Italian casualties; the campaign ended within days with minor territorial changes. Germany started preparation for an invasion of the Balkans to assist Italy, to prevent the British from gaining a foothold in the Balkans, which would be a potential threat for Romanian oil fields, and to strike against the British dominance of the Mediterranean. In December 1940 British Commonwealth forces began counter-

offensives against Italian forces in Egypt and Italian East Africa. The offensives were highly successful; by early February 1941 Italy had lost control of eastern Libya, and large numbers of Italian troops had been taken prisoner. The Italian Navy also suffered significant defeats, with the Royal Navy putting three Italian battleships out of commission by a carrier attack at Taranto and neutralizing several more warships at the Battle of Cape Matapan. Italian defeats prompted Germany to deploy an expeditionary force to North Africa, and at the end of March 1941 Rommel's Afrika Korps launched an offensive which drove back the Commonwealth forces. In under a month, Axis forces advanced to western Egypt and besieged the port of Tobruk. By late March 1941 Bulgaria and Yugoslavia signed the Tripartite Pact. However, the Yugoslav government was overthrown two days later by pro-British nationalists. Germany responded with simultaneous invasions of both Yugoslavia and Greece, commencing on 6 April, 1941. Yugoslavia and Greece were forced to surrender within the month.

The airborne invasion of the Greek island of Crete at the end of May completed the German conquest of the Balkans. Although the Axis victory was swift, bitter and large-scale partisan warfare subsequently broke out against the Axis occupation of Yugoslavia, which continued until the end of the war. In the Middle East, in May Commonwealth forces quashed an uprising in Iraq which had been supported by German aircraft from bases within Vichy-controlled Syria. In June–July they invaded and occupied the French possessions Syria and Lebanon, with the assistance of the Free French. With the situation in Europe and Asia relatively stable, Germany, Japan, and the Soviet Union made preparations. With the Soviets wary of mounting tensions with Germany and the

Japanese planning to take advantage of the European War by seizing resource-rich European possessions in Southeast Asia, the two powers signed the Soviet–Japanese Neutrality Pact in April 1941. By contrast, the Germans were steadily making preparations for an attack on the Soviet Union, massing forces on the Soviet border. Hitler believed that Britain's refusal to end the war was based on the hope that the United States and the Soviet Union would enter the war against Germany sooner or later. Hitler therefore decided to try to strengthen Germany's relations with the Soviets, or failing that, to attack and eliminate them as a factor. In November 1940, negotiations took place to determine if the Soviet Union would join the Tripartite Pact. The Soviets showed some interest, but asked for concessions from Finland, Bulgaria, Turkey, and Japan that Germany considered unacceptable. On 18 December 1940, Hitler issued the directive to prepare for an invasion of the Soviet Union. On 22 June 1941, Germany, supported by Italy and Romania, invaded the Soviet Union in Operation Barbarossa, with Germany accusing the Soviets of plotting against them. They were joined shortly by Finland and Hungary.

The primary targets of this surprise offensive were the Baltic region, Moscow and Ukraine, with the ultimate goal of ending the 1941 campaign near the Arkhangelsk-Astrakhan line, from the Caspian to the White Seas. Hitler's objectives were to eliminate the Soviet Union as a military power, exterminate Communism, generate Lebensraum (living space) by dispossessing the native population and guarantee access to the strategic resources needed to defeat Germany's remaining rivals. Although the Red Army was

preparing for strategic counter-offensives before the war, Barbarossa forced the Soviet supreme command to adopt a strategic defense. During the summer, the Axis made significant gains into Soviet territory, inflicting immense losses in both personnel and materiel. By the middle of August, however, the German Army High Command decided to suspend the offensive of a considerably depleted Army Group Centre, and to divert the 2nd Panzer Group to reinforce troops advancing towards central Ukraine and Leningrad. The Kiev offensive was overwhelmingly successful, resulting in encirclement and elimination of four Soviet armies, and made possible further advance into Crimea and industrially developed Eastern Ukraine.

The diversion of three quarters of the Axis troops and the majority of their air forces from France and the central Mediterranean to the Eastern Front prompted Britain to reconsider its grand strategy. In July, the United Kingdom and the Soviet Union formed a military alliance against Germany. The British and Soviets invaded neutral Iran to secure the Persian Corridor and Iran's oil fields. In August, the United Kingdom and the United States jointly issued the Atlantic Charter. By October Axis operational objectives in Ukraine and the Baltic region were achieved, with only the sieges of Leningrad and Sevastopol continuing. A major offensive against Moscow was renewed; after two months of fierce battles in increasingly harsh weather the German army almost reached the outer suburbs of Moscow, where the exhausted troops were forced to suspend their offensive. Large territorial gains were made by Axis forces, but their campaign had failed to achieve its main objectives: two key cities remained in Soviet hands, the Soviet capability to resist was not broken, and the Soviet Union retained a considerable part of

its military potential. The blitzkrieg phase of the war in Europe had ended. By early December, freshly mobilized reserves allowed the Soviets to achieve numerical parity with Axis troops. This, as well as intelligence data which established that a minimal number of Soviet troops in the East would be sufficient to deter any attack by the Japanese Kwantung Army, allowed the Soviets to begin a massive counter-offensive that started on 5 December all along the front and pushed German troops 100–250 kilometers west.

2. A Walk to Remember (Chapter 1) – Fictional Novel text

Beaufort is located on the coast near Morehead City, North Carolina. In 1958, Beaufort was a place like many other small southern towns. Beaufort was the kind of place where the humidity rose so high in the summer that walking out to get the mail made a person feel as if he needed a shower, and kids walked around barefoot from April through October beneath oak trees draped in Spanish moss. People waved from their cars whenever they saw someone on the street whether they knew him or not, and the air smelled of pine, salt, and sea, a scent unique to the Carolinas. For many of the people in Beaufort, fishing in the Pamlico Sound or crabbing in the Neuse River was a way of life, and boats were moored wherever you saw the Intracoastal Waterway. Only three channels came in on the television, though television was never important to those of people who grew up in Beaufort. Instead, Beaufort's people's lives were centered around the churches, of which there were eighteen within the town limits alone. The churches went by names like the Fellowship Hall Christian Church, the Church of the Forgiven People, the Church of Sunday Atonement, and the Baptist churches. When I was growing up, it was far and away the most popular denomination around, and in Beaufort were Baptist churches on practically every corner of town, though each church considered itself superior to the other churches. There were Baptist churches of every type. Beaufort had Freewill Baptists, Southern Baptists, Congregational Baptists, Missionary Baptists and Independent Baptists. Back then, the big event of the year was sponsored by the Baptist church downtown, Southern, in conjunction with the local high school. Every year Baptist church put on their Christmas pageant at the Beaufort Playhouse, Beaufort

Playhouse was actually a play that had been written by Hegbert. Hegbert is a minister, who'd been with the church since Moses parted the Red Sea. Hegbert was old enough that you could almost see through the Hegbert skin. It was sort of clammy all the time, and translucent. Kids would swear they actually saw the blood flowing through Hegbert veins. Hegbert hair was as white as those bunnies you see in pet stores around Easter. Hegbert wrote this play called Angel because Hegbert did not want to keep on performing that old Charles Dickens classic called A Christmas Carol. In Hegbert mind, Scrooge was a heathen, Scrooge came to his redemption only because Scrooge saw ghosts, not angels and who was to say whether they'd been sent by God, anyway? And who was to say Scrooge would not revert to his sinful ways if they had not been sent directly from heaven? Angel did not exactly tell you, in the end, it sort of plays into faith and all but Hegbert did not trust ghosts if ghosts were not actually sent by God, which was not explained in plain language, and this was Hegbert big problem with Angel. A few years back, Hegbert would change the end of the play sort of followed it up with his own version, complete with old man Scrooge becoming a preacher and all, heading off to Jerusalem to find the place where Jesus once taught the scribes. Angel did not fly too well not even to the congregation, who sat in the audience staring wide-eyed at the spectacle and the newspaper said things like, though Angel was certainly interesting, Angel was not exactly the play we've all come to know and love. So Hegbert decided to try his hand at writing his own play. Hegbert would write his own sermons his whole life, and some of them, we had to admit, were actually interesting, especially when Hegbert talked about the wrath of God coming down on the fornicators and all that good stuff.

That really got Hegbert blood boiling when Hegbert talked about the fornicators. That was Hegbert's real hot spot. When we were younger, my friends and I would hide behind the trees and shout that he is a fornicator when we saw Hegbert walking down the street, and we'd giggle like idiots like we were the wittiest creatures ever to inhabit the planet. Hegbert would stop dead in his tracks and his ears would perk up. I swear to God, they actually moved. Hegbert would turn this bright shade of red, like Hegbert had just drunk gasoline and the big green veins in his neck would start sticking out all over, like those maps of the Amazon River that you see in National Geographic. Hegbert would peer from side to side, his eyes narrowing into slits as he searched for us, and then, just as suddenly, Hegbert would start to go pale again, back to that fishy skin, right before our eyes. Boy, it was something to watch, that's for sure. So we'd be hiding behind a tree and Hegbert would stand there waiting for us to give ourselves up as if Hegbert thought we'd be that stupid. We'd put our hands over our mouths to keep from laughing out loud, but somehow Hegbert would always zero in on us. Hegbert would be turning from side to side, and then Hegbert would stop, those beady eyes coming right at us, right through the tree. I know who you are, Landon Carter, Hegbert would say, and the Lord knows, too. Hegbert would let that sink in for a minute or so, and then Hegbert would finally head off again, and during the sermon that weekend Hegbert would stare right at us and say something like, God is merciful to children, but the children must be worthy as well. And we'd sort of lower ourselves in the seats, not from embarrassment, but to hide a new round of giggles. Hegbert didn't understand us at all, which was really sort of strange, being that Hegbert had a kid and all. But then again, Hegbert was a girl. More on that,

though, later. Anyway, like I said, Hegbert wrote Angel one year and decided to put on that play instead. Angel itself wasn't bad, actually, which surprised everyone the first year it was performed. Angel is basically the story of a man who had lost his wife a few years back. Tom Thornton, used to be real religious, but Tom Thornton had a crisis of faith after his wife died during childbirth. Tom Thornton's raising this little girl all on his own. Tom Thornton hasn't been the greatest father. What the little girl really wants for Christmas is a special music box with an angel engraved on top, a picture of which she'd cut out from an old catalog. Tom Thornton searches long and hard to find the gift, but Tom Thornton can't find it anywhere. So it's Christmas Eve and Tom Thornton is still searching, and while Tom Thornton is out looking through the stores, Tom Thornton comes across a strange woman Tom Thornton has never seen before, and she promises to help Tom Thornton find the gift for his daughter. First, though, they help this homeless person then they stop at an orphanage to see some kids, then visit a lonely old woman who just wanted some company on Christmas Eve.

3. Biography Text (https://en.wikipedia.org/wiki/Donald_Trump)

America is the second largest country in North America. America is made up of 50 states, a federal district, and five territories. America has great influence over world finance, trade, culture, military, politics, and technology. America is a federal republic. America consists of 50 states, 5 territories and 1 district called WashingtonDC. States can make laws about things inside the state, but federal law is about things dealing with more than one state or dealing with other countries. In some areas, if the federal government makes

laws that say different things from the state laws, people must follow the federal law because the state law is not a law any more. Each state has a constitution of its own, different from the federal (national) Constitution. Each of these is like the federal Constitution because they say how each state's government is set up, but some also talk about specific laws. The federal and most state governments are dominated by two political parties, the Republicans and the Democrats. There are many smaller parties. The largest of these are LibertarianParty and GreenParty. People help in political campaigns that they like. They try to persuade politicians to help them. This is called lobbying. All Americans are allowed to do these things, but some have and spend more money than others, or in other ways do more in politics. Some people think this is a problem, and lobby for rules to be made to change it. Since 2017, the president is a Republican, and Congress is also Republican controlled, so the Republicans have more power in the federal government. There are still many powerful Democrats who can try to stop the Republicans from doing things that they believe will be bad for the country. Also, members of a party in power do not always agree on what to do. If enough people decide to vote against Republicans in the next election, they will lose power. In a republic like America, no party can do whatever they want. All politicians have to argue, compromise, and make deals with each other to get things done. They have to answer to the people and take responsibility for their mistakes. America's large cultural, economic, and military influence has made the foreign policy of America, or relations with other countries, a topic in American politics, and the politics of many other countries. America conquered and bought new lands over time, and grew from the original 13 colonies in the east to the

current 50 states, of which 48 of them are joined together to make up the contiguous America. These states, called TheLower, can all be reached by road without crossing a border into another country. TheLower go from the Atlantic east to the Pacific in the west. There are two other states which are not joined to the lower 48 states. Alaska can be reached by passing through British Columbia and the Yukon. British Columbia and the Yukon are part of Canada. Hawaii is located in the middle of the Pacific Ocean and is so far from the rest of America that Hawaii can only be reached by airplane.

WashingtonDC, the national capital, is a federal district that was split from the states of Maryland and Virginia in 1791. Not part of any American state, WashingtonDC used to be in the shape of a square, with the land west of the Potomac River coming from Virginia, and the land east of the river coming from Maryland. In 1846, Virginia took back its part of the land. Some people living in DC wants it to become a state, or for Maryland to take back its land, so that they can have the right to vote in Congress.

Donald Trump was born on June 14, 1946. Donald Trump is the 45th and current President of America. Before becoming president, Donald Trump was a businessman and television personality. Donald Trump was also the chairman and president of The Trump Organization. Much of Donald Trump's money was made in real estate in New York City, Las Vegas, and Atlantic City. Donald Trump used to own the MissUniversePageant. Donald Trump was the star in his own reality show The Apprentice. In June 2015, Donald Trump announced that Donald Trump would run for President of America in the 2016 presidential election. Starting mid-July, polls showed that Donald Trump was the front-runner in the Republican field. This was true even after much criticism from the

party due to Donald Trump's comments on illegal immigration, Muslims, and ISIS. Donald Trump's campaign has gained support from mostly middle-class families. It has gained opposition from Democrats, some Republicans, business people, some world leaders and the pope. Donald Trump was born in Queens, New York City. Donald Trump is the son of Fred Trump and Mary Anne. Fred Trump married Mary Anne in 1936. Mary Anne was born on the IsleOfLewis, off the west coast of Scotland. Donald Trump was one of five children. Donald Trump's oldest brother, FredJr, died in 1981 at the age of 43. Donald Trump's sister is Maryanne. Maryanne is a judge in New York. Donald Trump's father's parents were German immigrants. Donald Trump's grandfather was Frederick Trump. Frederick Trump immigrated to America in 1885. Frederick Trump became a naturalized American citizen in 1892. Frederick Trump married Elisabeth Christ at Kallstadt, on August 26, 1902. Frederick Trump and Elisabeth Christ had three children. Frederick Trump studied at Fordham University until transferring to the University of Pennsylvania. In 2003, Donald Trump became the executive producer and host of the NBC reality show TheApprentice. In TheApprentice, a group of competitors battled for a high-level management job in one of Donald Trump's commercial enterprises. In 2004, Donald Trump filed a trademark application for the catchphrase YoureFired. For the first year of the show, Donald Trump earned $50,000 per episode of TheApprentice, but following The Apprentice's initial success, Donald Trump was paid $1 million per episode. In a July 2015 press release, Donald Trump's campaign manager said that NBCUniversal had paid Donald Trump $213,606,575 for his 14 seasons hosting the show. On February 16, 2015, NBC announced that NBC would be renewing

TheApprentice for a 15th season. Donald Trump was replaced by former Governor of California and actor, Arnold Schwarzenegger. After becoming the presumptive Republican nominee, Donald Trump's focus shifted to the general election, urging remaining primary voters to save their vote for the general election. Donald Trump began targeting Hillary Clinton. Hillary Clinton became the presumptive Democratic nominee on June 6, 2016 after beating Bernie Sanders in the Democratic primaries and continued to campaign across the country. Hillary Clinton had established a significant lead in national polls over Donald Trump throughout most of 2016. In early July, Hillary Clinton's lead narrowed in national polling averages following the FBI's conclusion of its investigation into her ongoing email controversy. On September 26, 2016, Donald Trump and Hillary Clinton faced off in the first presidential debate at Hofstra University in New York. Lester Holt is an anchor with NBC News. Lester Holt was the moderator. This was the most watched presidential debate in American history. On November 8, 2016, Donald Trump won the presidency with 304 electoral votes to Hillary Clinton's 227 votes. Donald Trump won a smaller share of the popular vote than Hillary Clinton. Donald Trump is the fifth person to become president without winning the popular vote. The final popular vote difference between Hillary Clinton and Donald Trump is that Hillary Clinton finished ahead by 2.86 million or 2.1 percentage points, 48.04% to 45.95%, with neither candidate reaching a majority. Donald Trump's victory was considered a big political upset, as nearly all national polls at the time showed Hillary Clinton with a modest lead over Donald Trump. State polls showed Hillary Clinton with a modest lead to win the Electoral College. In the early hours of November 9, 2016, Donald Trump

received a phone call in which Hillary Clinton conceded the presidency to Donald Trump. Donald Trump then delivered his victory speech before hundreds of supporters in the Hilton Hotel in New York City. On January 20, 2017, Donald Trump was sworn in by Chief Justice John G. Roberts as President of America at his inauguration ceremony at the UnitedStatesCapitolBuilding. Within Donald Trump's first hour as president, Donald Trump signed several executive orders, including an order to minimize the economic burden of AffordableCareAct. AffordableCareAct is also known as Obamacare. On the Saturday following Donald Trump's inauguration there were massive demonstrations protesting Donald Trump in America and WorldWide. On January 23, 2017 Donald Trump signed the executive order withdrawing America from TransPacific. TransPacific is a trade agreement between PacificRim and America. PacificRim also contains Australia. PacificRim also contains Chile, Japan, Peru, Singapore and Vietnam. PacificRim also contains Brunei, Canada and Zealand. PacificRim also contains Malaysia and Mexico. PacificRim would have created a free-trade zone for about 40 percent of the world's economy. Two days later, Donald Trump ordered the construction of the MexicoBorderWall. Donald Trump reopened the Keystone XL and Dakota Access pipeline construction projects. On January 27, Donald Trump signed an order suspended admission of refugees for 120 days and denied entry to citizens of Iraq, Iran, Libya, Somalia, Sudan, Syria and Yemen for 90 days, citing security concerns about terrorism. Later, the administration seemed to reverse a portion of part of the order, effectively exempting visitors with a green card. FederalJudges issued rulings that curtailed parts of the immigration order, stopping Donald Trump from deporting visitors already affected.

On January 30, 2017, Donald Trump fired Attorney General Sally Yates because of Attorney General Sally Yates's criticisms of Donald Trump's immigration suspension. On January 31, 2017, Donald Trump nominated Judge Neil Gorsuch to the American Supreme Court to replace the late Justice Antonin Scalia. Michael T. Flynn was National Security Advisor. After The Wall Street Journal reported that Michael T. Flynn was under investigation by America counterintelligence agents for his communications with Russian officials, Michael T. Flynn resigned on February 13, 2017. Andrew Puzder was Donald Trump's secretary of Labor-nominee. Two days later on February 15, Andrew Puzder withdrew his nomination due to not having support from Democrats or Republicans to confirm his nomination. On April 7, 2017, Donald Trump ordered the launch of 59 Tomahawk cruise missiles from the Mediterranean Sea into Syria. Tomahawk aimed at Shayrat Air Base because of the chemical attack by Khan Shaykhun. Donald Trump married his first wife, Ivana Zelníčková on April 7, 1977, at the Marble Collegiate Church in Manhattan. Ivana Zelníčková was a Czech model. Donald Trump and Ivana Zelníčková had three children. Donald Trump and Ivana Zelníčková had a son named Donald Trump Jr. on December 31, 1977. Donald Trump and Ivana Zelníčková had a daughter named Ivanka Trump on October 30, 1981. Donald Trump and Ivana Zelníčková had another son named Eric Trump on January 6, 1984. Ivana Zelníčková became a naturalized citizen of America in 1988. By early 1990, Donald Trump's troubled marriage to Ivana Zelníčková and Donald Trump's affair with actress Marla Maples had been reported in the tabloid press. Donald Trump and Ivana Zelníčková were divorced in 1992. Donald Trump married, his second wife, actress Marla Maples in 1993.

Donald Trump and Marla Maples had a daughter Tiffany on October 13, 1993. Donald Trump and Marla Maples were separated in 1997 and later divorced in 1999. In 1998, Donald Trump began a relationship with Slovene model Melania Knauss. Melania Knauss became Donald Trump's third wife. Melania Knauss and Donald Trump were engaged in April 2004 and were married on January 22, 2005, at Bethesda-by-the-Sea Episcopal Church, on the island of Palm Beach, Florida. In 2006, Melania Knauss became a naturalized citizen of America. On March 20, 2006, Melania Knauss and Donald Trump had a son, whom they named Barron Trump. Mohammad Bin Salman Al Saud was born on 31 August 1985. Mohammad Bin Salman Al Saud is the Crown Prince of Saudi Arabia, First Deputy Prime Minister of Saudi Arabia and the youngest minister of defense in the world. Mohammad Bin Salman Al Saud is also chief of the HouseOfSaud Royal Court, and chairman of the CouncilForEconomicAndDevelopmentAffairs. Mohammad Bin Salman Al Saud has been described as the power behind the throne of his father, King Salman. Mohammad Bin Salman Al Saud was appointed Crown Prince in June 2017 after King Salman's decision to remove Muhammad Bin Nayef from all positions, making Mohammad heir apparent to the throne.