Spring 2024

Dumonym: Crafting and Assessing Lexical Simplification, from Algorithms to Models

Jeremias Brea De Los Angeles
Bard College

Dumonym:
Crafting and Assessing Lexical Simplification, from algorithms to models

Senior Project Submitted to
The Division of Computing
of Bard College

by

Jeremias Brea De Los Angeles

Annandale-on-Hudson, New York
May 2024

Dedication

I dedicate this project to my parents Yudelka De Los Angeles and Richard Hegeman. Your support, faith, and enduring love have been invaluable. I am truly grateful for the life you two have given me.

Acknowledgements

I would like to express my deepest gratitude for my advisor, Professor Kerri-Ann Norton. Thank you for your guidance and support. I am truly grateful for your presence in helping me understand the different components of my project. Your patience and dedication is also greatly appreciated, especially during the times where I may have not understood the topics completely. I cannot express how much I have learned working with you this semester. Kerri-Ann Norton has helped me greatly, in improving my skills and learning how to apply them to real-life scenarios. You have also taught me the value of taking a step back sometimes when things get tough and coming back to it when refreshed. Kerri Has been a mentor to me during this project, teaching me skills that I will use throughout my life, thank you! I would also like to express my gratitude to Professors, Rose Sloan and Sven Anderson. Although I may not have reached out to you two too often, the guidance and advice you have given me along the way has greatly helped the development of this project, and has helped broaden my understanding of computer science. I would also like to thank Kelsey Olivera for helping me keep on track, as well as helping me maintain stability during the difficult moments these past few semesters, thank you! I would finally like to thank David Shein for helping me maintain the focus I needed to complete this project. If you did not push me the way you did this semester I am not so sure I would have completed as much as I have for this project. Thank you everyone for your support!

Abstract

Lexical Simplification is the process of replacing complex words with simpler alternatives in a given text. This project aims to use different approaches in the field of Natural Language Processing to create a series of lexical Simplification models. The framework of lexical simplifiers will also be explored and researched, to give more insight of the mechanisms and approaches used to achieve successful text simplification. I will develop a pipeline of steps, based on my research, with the aim to create a framework for a functional lexical simplification model. I will develop a series of distinct lexical simplification models based on my pipeline, with the expectation of creating an ideal lexical simplification model. The models I develop will be tested and compared to one another to find strengths and weaknesses for potential improvements. The performance of my models were measured utilizing the Chi-Squared metric for three hypothesis tests that aim to assess the impact of my models. After analysis the Text Frequency Thesaurus Simplification model was superior in correctly identifying complex terms as well as potentially having the best approach for generation. Although the tests were very insightful, more testing and analysis can potentially yield better results for comparison. I will also discuss potential improvements intended to solve some of the weaknesses of my models for future iterations.

Table of Contents

List of Tables and Figures:

Introduction

1.1 Personal Anecdote

When interpreting a reading it is crucial to recognize the main point of the text in order to understand and interpret the text. In the case where there is a text that has complex terminology, knowing the main point of the text may pose an obstacle to readers, limiting their ability to analyze the text. The use of complex terminology in texts can be the result of using field specific terms, which can confuse readers outside a field. For instance scientists may be confused by the terms used in classic literature because the terms are specific to the classics field. During a Roman history class I took as a sophomore, I often struggled with locating specific information in our given texts. While there were some texts that were difficult to interpret because of their translation from Roman to English, other texts were difficult to interpret because of my unfamiliarity with the terminology used in the class materials. Therefore I had two sources of confusion, one from the complexity of the text itself and the second being the terminology used in literary analysis. With this confusion, I would often miss the focus of many sentences, limiting my understanding of the material as a result. Perhaps this challenge can be reduced with a simplified version of these texts. A lexical simplifier, in such a situation, could have the potential to clarify any information from a given text that may have been missed by a reader, such as myself.

1.2 What is Complexity?

What is a complex word? For some, it may be viewed as a word that cannot be understood, such scenarios can limit information shared with an audience. An example could be the difference between a phrase such as, “we will endeavor to assist you” and the phrase: “we will try to help you”[3]. These examples have a similar meaning, but one uses a longer and more complex word “endeavor” compared to the shorter, more simple word “try”. Perhaps the difference in word length of these terms can aid in identifying terms that are more difficult or vague. In my research, two notable articles[1][3], seem to relay different interpretations of word difficulty. According to the article *Complex word in English* by Richard Nordquist[1], he mentions that the difficulty of text is defined by two or more morphemes in a word. A morpheme according to *Christy Paluti*[2], is a word that cannot be divided into smaller segments making it the smallest unit of words in a language. Morphemes can be words such as ‘cabin’ in the term ‘cabinet’ or ‘ed’ in the word ‘walked’. In these instances ‘cabin’ would be a morpheme because it is a base word in ‘cabinet’ that cannot be divided, ‘et’ is also an example of a morpheme. Some morphemes cannot be a word unless it’s attached to another. This is intriguing because it reveals a measurable definition of word complexity that shows direct semantic relationships within a single word instead of viewing its comprehensibility. For *Complex and Abstract Words* by Nick Wright[3], his interpretation refers to a degree of word complexity called abstract language, which he defines as being terms that cumulatively affect text by making sentences more vague. Wright's[3] paper aims to reveal the hindering effect that abstract language can have with differing impartments of information. Wright[3] also mentions the use of shorter words in the simplification of some of these abstract statements. In the previous word examples above, between the terms 'endeavor' and 'try' we see that the word 'try' is not only direct, but shorter as well. This begs the question: could changing word size to the best of our ability retain semantics and improve conveyed information? Although this idea of word length simplification is linear and simple, it will be interesting to create an identifier that uses word length and potentially alternative measures to simplify a text. Will the sentence retain its semantic value after simplifying a given text? Combining these different interpretations of word complexity could improve simplification techniques. Lexical simplifiers have the ability to improve the effectiveness of information conveyed or translated to a large audience.

1.3 Potential Solutions

There are many approaches that can be taken to address the issue of term complexity in readings of interest. An initial approach to the problem can be a simplifier that outputs a 'simple' text based on short word replacements. Another approach, can be the use of a simplifier that returns simple text based on frequency in a trained dictionary. The rationale in this method is to identify term complexity as terms with low frequencies and replacing them with terms of higher frequencies. According to research, techniques in the field of Natural Language Processing can help optimize the mentioned approaches or create new methods entirely. These techniques can be anything, from the use of word embeddings to the use of a transformer model to execute the task of lexical simplification. Research in the topic of Lexical Simplification has also revealed that the type of learning that these models used can be significant.

1.4 Natural Language Processing

Natural Language Processing is a field that uses techniques in computer science to help understand characteristics of human language. According to the article *A Complete Guide to Natural Language Processing* by deeplearning.ai[4] the authors refer to the discipline as building machines to manipulate language in a way we can understand. Deeplearning.ai also mentions the proficient use of this discipline in many fields ranging from medicine and retail to the field of entertainment with chatbots. Within NLP there are many different approaches and techniques for language-related tasks. According to the article[4], preprocessing text, feature extraction and modeling are mentioned as crucial tasks of Natural Language Processing. For the development of a lexical simplifier in this project, these factors of Natural Language Processing are necessary.

- **Tokenization**

Later in the article[4], the authors mention data preprocessing as a crucial step for language processing tasks. It's mentioned that before inserting text data into a model it is necessary to preprocess the text dataset(s) before use, to improve performance. There many approaches to preprocessing text data including, but not limited to, stemming, removing stopwords, or digit removal. A processing method that seems very useful for the task of lexical simplification is tokenization. Tokenization is the process of splitting a text into sentences, words or word fragments[4]. This method of preprocessing can potentially be used to format text in a way that allows for analysis of a given text dataset based on specified parameters.

- **Word embedding**

Another crucial factor is the task of feature extraction, it is mentioned[4] that many techniques of machine learning use feature extraction. This factor of NLP typically aims to find any relationships that occur with a preprocessed set of text data. A use for this NLP factor in the project will be the extraction of words that are too long in a data set, or even words that frequently occur in a text. This particular task can help identify the semantic relationship between a selected word and other words within a dataset, allowing for various text manipulations based on those semantic relations. A mentioned[4] example of feature extraction is the word2vec model, which is described as a word embedding that takes context into account when analyzing a given term. This model has the potential to be very useful for different tasks in text simplification.

- **Unigram Bigram and N Gram models**

The article from deeplearning.ai[4] mentions modeling as the third remaining crucial part of Natural Language Processing. The guide also mentions that this step typically structures input data to perform various tasks[4]. In relation, the function of language models involves predicting terms when given a corpus of data. An example of a language model can be a bigram, which predicts the next word from previous terms used as input. Models such as these can help structure a simplifier that uses similar techniques to predict the terms required for a simplified version of given text. Modeling will be a significant component for the creation of a lexical simplification model in this project.

- **Thesaurus**

Although the guide by Deepleaning.ai mentions three crucial factors to language processing, it forgets to mention other resources that can be used in development. These resources can vary from specific functions to entire libraries depending on different tasks. An example can be a thesaurus, as defined by computerhope.com[5], it can be a book, program, or online service that provides alternative or similar words to a given word. The use of resources like a thesaurus can allow access to word semantics in a way that could be more efficient than the use of a previously mentioned word2vec model, for example. With access to resources like a thesaurus, it may allow more room for improvement when developing various models in NLP.

1.5 NLP Libraries

A major resource in the development of this project is the use of libraries specified for natural language tasks. The libraries used are NLTK and Gensim, both of which specialize in natural language or machine learning tasks. *Natural language Processing with python* by Loper and Klein[6] will be the most useful for navigating and implementing different functions and applications in NLTK and Gensim. The main uses for the libraries will be Tokenization, Wordnet and Gensim. Tokenization will be a key piece in this project for analyzing given text data, with NLTK, a built-in function for tokenizing can be called instead of using manual alternatives[6]. During the development of a simplifier there may be a need to implement a word2vec model to perform feature extraction from a given text. The Gensim library allows access to various types of pre-trained word embeddings. Thesaurus' can also be used in the NLTK library with a call to Wordnet, which can give access to various synonyms and antonyms to a given word[6]. With these different features of NLTK and Gensim the task of lexical simplification can be very approachable.

1.6 Papers of Influence

Lexical Simplification

There has been much research in the subject of Lexical simplification. In the study, *A Survey On Lexical Simplification*[7] by Paetzold and Specia, Lexical simplification is described as the process of replacing complex or difficult words in a text with simpler alternatives. The research aims to make text more accessible to a wider range of readers, including those with limited language proficiency or limited exposure to vocabulary. This field is significant, as many people struggle with understanding complex language and vocabulary, which can hinder access to information. There are many approaches to lexical simplification, ranging from rule-based methods that rely on dictionaries, to thesauri that can identify synonyms as alternatives to given terms. There are even more advanced machine learning techniques that use large-scale corpora and possibly neural network or transformer framed models to automatically identify and generate simplified versions of text.

A key challenge identified by Paetzold and Specia[7], is how to accurately identify which words are difficult or complex for a given audience. This can vary depending on factors such as education level, language proficiency, and cultural background. To address this challenge, researchers often rely on human evaluations to gather data on the readability and understandability of different texts. For example, an earlier study by Paetzold[8] used 400 in-person participants to evaluate whether or not they could understand the meaning of words from different parts of speech.

Another challenge is finding balance between simplicity and maintaining the semantics and syntax of a given sentence. Replacing complex words with simpler alternatives can sometimes lead to unintended changes in meanings, which can make text less cohesive or understandable. To address this problem Paetzold[7] mentions various algorithms and models developed to take into account not only the difficulty of individual words, but also the context and syntax of words within sentences. Many of these approaches generally use a framed pipeline of four steps: Complex word identification, Substitution generation, Substitution selection, and Substitution ranking. The pipeline looks like the following:

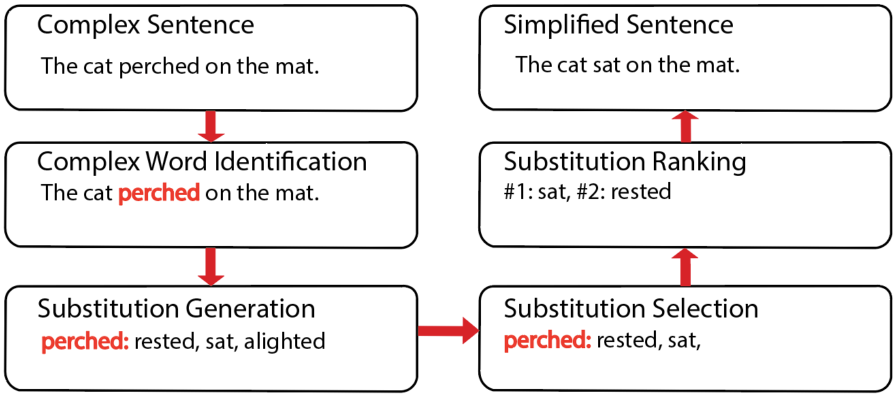

Figure 1: General Lexical Simplification Pipeline[7]

This pipeline will be incredibly useful in the development of this project. Lexical simplification research has the potential to be applied to many fields such as education, healthcare, and public communication. An example could be of simplified passages that can help improve understanding of literature making information more accessible to readers.

Unsupervised Lexical Simplification Model

An interesting approach referenced in the article, *The Survey On Lexical SimplificationI*[7], is the Unsupervised Lexical Simplification model. The authors of the previous study, Paetzold and Specia[7], conducted an older study called *Unsupervised Lexical Simplification* for Non Native Speakers[8], which was quite noteworthy. In the study the researchers aimed to create a model that could help the understanding of texts for non native english speakers. The researchers break down their process of using a four step pipeline, similar to the pipeline of the previous survey study[7]. In each step, the researchers explain their specific approaches for the individual tasks of the model and how they are conducted. A unique component of the study's model[8], is the use of 400 Non-Native English speaking participants for its evaluation of complex text, which can improve simplification quality. The general pipeline of the model incorporates this task as the first step of the model, complex word identification.

The next step in this study's model[8] takes the resulting complex words and finds terms that share the same meanings. The article uses context-aware word embedding models that train on annotated corpora for the step of substitution generation. Paetzold[8] references the use of Part-of-Speech(POS) tags for annotating a corpus. This corpus is then used to train a context-aware word embedding which generates alternatives for each complex word or phrase that is identified. The context-aware word embedding captures the contextual meaning of words and phrases based on the surrounding language used within its training text. The selected candidate terms are retrieved based on a probability of similarity, being applied to the word embedding.

The following step in the pipeline is substitution selection[8], where words that are closer in meaning to the identified terms are selected to feed into the ranking step. In this portion of the model, Paetzold[8] details the use of a technique called boundary ranking as their method of word selection in this piece of the model. Boundary ranking uses classification techniques to find semantic relations between words, labeling them on a number scale. This helps filter synonyms for the better candidates when passing them onto the ranking portion of the model.

The final step used in the model is substitution ranking, which works by using a corpus of movie subtitles to determine the simplest of the selected candidate terms for substitution. These candidates are ranked based on the most frequent terms found in the data set of subtitles. Using a corpus of movie subtitles is a key aspect in substitution ranking, as it provides a large and diverse source of preprocessed text that can be used to find words with the highest frequencies. The selected terms of substitution rank are the final terms that will replace the identified complex words of a given sentence.

Paetzold also speaks on how the model is tested using a dataset of simplified english texts and comparing it to various simplification models. The results showed that the Unsupervised Lexical Simplification Model achieved the most effective performance in comparison to the other models, while also being unsupervised, which means it does not require any labeled training data[8]. Even though the model does not use a neural network or transformer framework, the Unsupervised Lexical Simplification Model is a promising approach to simplifying text for non-native speakers of English. It has the potential to help make information more accessible to a larger audience and improve comprehension of relayed information.

Traditional chinese medicine normalizer

The research study, *Natural Language Processing Algorithms for Normalizing Expressions of Synonymous Symptoms in Traditional Chinese Medicine* by Lu Zhou[9], has greatly aided in improving my understanding of Machine Learning models. The study was created and conducted by a series of well versed programmers aiming to solve an issue of hindered data mining performance regarding records of symptoms within the field of traditional chinese medicine. A main cause of the low performance quality was due to the numerous amounts of synonymous words for symptoms and descriptions on the given records. The researchers of the study[9] proposed using Natural Language Processing techniques to help normalize the expressions of the synonymous symptoms. An approach taken was to create a series of varying natural language models that are then tested on performance for synonym normalization. These models consisted of 1); a text sequence generation model with an encoder and decoder structure based on a bidirectional long short-term memory(Bi-LSTM) neural network(Bi-LSTM transformers) 2); a text classification model based on a Bi-LSTM neural network and sigmoid function(Bi-LSTM classification) 3); another sequence generation model based on a bidirectional encoder representation with sequence to sequence training method of unified language model(BERT-UniLM); 4); A text classification model based on BERT and Sigmoid function(BERT-classification). After extensive testing between both HFDS and TDS the results point to the BERT-Classification model having the best results. The different approaches used for lexical simplification in this study[9] has greatly influenced further interest in researching machine learning approaches for lexical simplification.

Methods

2.1 Summary of methodology

The aim of this project will be to develop a series of simplification models, to find the most optimal model approach for lexical simplification. To do this, I will utilize two python libraries, the first is called Natural Language ToolKit or NLTK, which has access to a variety of natural language resources. The other library I use is called the Generate Similar Model, or Gensim for short, which focuses on prebuilt natural language processing models accessing multiple NLP and machine learning resources as well. The approaches used in this project's models will be based on a general four-step pipeline that is referenced in the *Lexical Simplification Survey* by Paetzold and Specia[7]. Although the pipeline has been a significant influence in this project's development, I will be adjusting it in order to achieve my desired output. The new pipeline I have updated will use the same steps of complex identification and substitution generation, but substitution selection will perform its general function as well as substitution rank's function. The new Substitution Rank will be called substitution replacement, with the main task of replacing the selected complex terms with their simple alternatives. This new pipeline will look like the following:

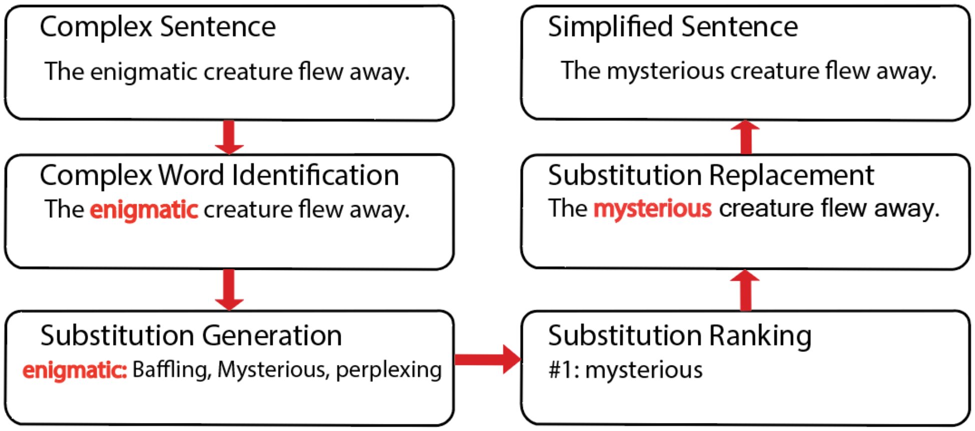
Complex Sentence
The enigmatic creature flew away.
Complex Word Identification
The enigmatic creature flew away.
Substitution Generation
enigmatic: Baffling, Mysterious, perplexing
Substitution Ranking
#1: mysterious
Substitution Replacement
The mysterious creature flew away.
Simplified Sentence
The mysterious creature flew away.

Figure 2: updated Lexical Simplification pipeline

In this project, I will create a series of models that reflect the concepts of word length or word frequency as methods of identifying word complexity. In these models, I will vary the approaches to generating substitutes for the complex words to explore which are most effective and understandable. Every model I create will vary on the approaches taken to fulfill the different tasks of the pipeline.

2.2 Personal framework breakdown

- Complex Word Identification

In previous instances, the task of complex word identification was done using human evaluations as its method. Although human surveys can be very effective for the task, this project's simplest model will identify word complexity based on word length. This is done by searching for words in a text that exceed a certain length of letters and saving it to a list of complex words. This method of complex word identification is not too intricate in nature, but it can be viewed as a more comprehensible method for the task. Although this identification approach is vague, in subsequent models I use more sophisticated approaches for determining word complexity, improving the accuracy of the mechanism in my pipeline. With the use of a corpus of text I will save every word in a dictionary along with a frequency count as a value for every word. With this approach, the various word frequencies may help determine which words are more difficult than others. In the results section, I examine which technique is more effective for identifying word complexity in this new pipeline.

- Substitution Generation

In this portion of the new pipeline, the main objective will be to generate alternative replacement terms with similar meanings to the complex terms. I will be using two different approaches for this task. The first will be the use of a pre-trained word2vec model from the Gensim library[6], which has access to a function that can generate synonyms for a given word. This technique could potentially be very effective within the pipeline. The second approach will be the use of a thesaurus in combination with the Gensim model to possibly improve the quality of the first approach. A thesaurus can be very useful for tasks that may require the use of synonymous or antonymic text generation. NLTK library's WordNet[6] thesaurus, will be used in combination with Gensim for this approach. The thesaurus will be used as a secondary condition in the case where there are no alternatives found in the Gensim word embedding. This method for substitution generation can potentially improve the quality of the pipeline's function.

- Substitution Ranking

Unlike the pipelines mentioned in the previous studies of Paetzold and Specia[7],[8], my modified pipeline will combine the functions of both substitution selection and substitution rank. This new pipeline step will be referred to as only substitution rank in development. The split tasks of selection and ranking will be merged into just one task before the substitution replacement step. In my initial model, my rank mechanism selected words that were the shortest in a given list of alternative words. Although this method is easily computed, it may not select the best terms for replacement. In my later models, I modified my approach to selection and ranking by utilizing the same dictionary of word frequencies that I later use for complex word identification. Instead of identifying complex terms with this saved dictionary, it will be used to identify the simplest words from a given list of alternative words. This new approach to substitution rank can be very effective in the quality of outputs.

- Substitution Replacement

In this step of the new pipeline, the main function will be to replace the complex terms of a given string with the selected alternative simple words. In my simplest model, the steps of replacement and ranking are done in a single function. This function directly replaces the largest terms with their smaller counterparts and returns a new simplified string. At the time of development, this technique for substitution replacement seemed reasonable, but later in my development I decided to separate both tasks. This separation of tasks is necessary as in my later approaches I also officially updated substitution rank as a separate function in my pipeline. In my newer models, substitution replacement only replaces old complex terms with their simple alternatives. With this new addition of substitution replacement, my new pipeline can give a more complete understanding of the lexical simplification process. These newer approaches to substitution replacement could possibly be beneficial for the efficiency of the new pipeline.

2.3 Preprocessing and Dictionary pre-training

Earlier in this project, I mentioned how text preprocessing is a fundamental task in NLP, that has remained true throughout my work. In the development of my dictionaries it was necessary to process my text to remove punctuation, and numbers. The text was also tokenized to account for every unique word. Preprocessing text in this way has allowed for trained data to be more practical and beneficial for simplification. This task has also become more crucial for the simplification of the strings through the simplifier models. Applying the same techniques of pre-processing to the string inputs can improve functionality by accounting for punctuation and integers which both have the capability of reducing the quality and performance of the lexical simplification models.

In a new script, I have created a function called 'cleen' which will perform the different tasks of preprocessing to a given text. For the tasks of the cleen function, I will import NLTK's word tokenize function as well as the string library. This function will be called throughout every program in this project that requires preprocessing. This includes the mentioned dictionary and string input, both of which call on an import of this cleen function. This function takes in a given string and tokenizes it using the word_tokenize function. The tokenized list is then filtered for numerical values, updating an empty list with everything except the identified numerals. This list is then looped through and checked for punctuation. If there are punctuation marks, they are removed, and what is returned is a new list with those adjustments made. With all of the numerical values and punctuation removed effectively, the models can produce better results.

Dictionary of Word Frequencies

In the majority of the models developed, I use a saved dictionary of word frequencies. I went through several stages using different text data to test how effective each text sample would be on the different simplification models. Nearing the end of the project, I think it's best to use a dictionary that encompasses all of the text data previously tested in an effort to cover as many words as possible. A larger dictionary gives the program a wider scope of words and values it can select, and therefore improves different selection tasks throughout the flow of the pipeline. In the majority of the models presented, one stark similarity is that they all utilize the dictionary the same way during the task of complex identification. As well as this, in the latter two models created, the dictionary assists with substitution ranking in the same manner. I kept these steps the same because they were functional and appeared more effective.

The use of a dictionary in this project was inspired by my senior project advisor, who gave me very good advice along the way. I learned through trial and error that utilizing a dictionary of word frequencies would be the best alternative to the method used prior(a list). The frequencies counted within this dictionary allow for algorithms to be developed for the tasks of complex identification and substitution ranking. To train this dictionary, I used multiple sources of text data, the majority of which was mostly geared towards elementary level readers, with the exception of one short story that has a higher reading level. It was quite the journey finding texts to use, as it was important to ensure that the readability was appropriate for creating simplified sentences. Children’s and middle-grade literature such as *Diary of a Wimpy Kid* made the most sense for this process.

To open the text dataset, I make a function which takes in the file path and returns the opened file. The corpus of data is then cleaned with an import function called cleen, which cleans the data for training later on. The preprocessed corpus data is then looped through while counting the frequency of each word. During the loop, every word and frequency count will be updated to an empty dictionary. With the help of a resource called Json I can call on the dump function to save the dictionary into a file for later retrieval and use. This approach to training and saving a dictionary of term frequencies is used for the dictionary calls in every model.

2.4 Model Descriptions

Model 1: Word Length Simplification Model(WLSM)

This model takes the simplest approaches to my four step pipeline, compared to the following models. The Framework of this model follows the steps of the new pipeline but executes them slightly differently. In this model, substitution rank and replacement both share the same function in code. It will be interesting to test whether the function used in this model improves the quality of simplified texts. ‘Word Length Simplification Model’, or WLSM for short, will be the name used to refer to this first model. In WLSM, the concept of word length is used for the identification of complex terms. The task of complex word identification is carried out by a function named identify, which returns a list of words that exceed a certain length from a given string. For the step of substitution generation, this model will use a function called w_embed that returns a listed list of synonyms corresponding to a given list of complex terms.

The generated synonyms can also not exceed the length of its complex counterpart. An example of what this conversion may look like can be;

['automobile','resentful']

[['car', 'vehicle', 'motor'],['envious', 'jealous']].

The synonymous terms generated in this step of the pipeline will come from the use of a Gensim's Library. With access to this resource, I will load in a pre-trained word2vec model and use a built-in function that returns synonyms of a given word. The resulting listed list from the w_embed function will then feed into a function called substitute. As mentioned earlier, this function will implement both tasks of substitution ranking and replacement from the new pipeline. The function named substitute, iterates through a given listed list of alternative words and looks for the smallest words in these lists. Once found, these words will replace the previous complex term, and the terms that don't have alternatives will remain the same. The return of this function will be an updated string with simplified text. The Word Length Simplification Model may directly combine steps of the framework, but it still follows the order of my new pipeline.

Model 2: Frequency Length Simplification Model(FLSM)

The second model I developed, Frequency Length Simplification Model (FLSM), was designed to improve some aspects of the previous model, WLSM. In this model and the following ones, a dictionary was created from a selection of children's novels. I will utilize the saved children's text dictionary containing words with word frequency values. This saved dictionary is called and opened as one of the initial procedures during development. For complex word identification, this model will utilize a new identify function that takes in a string and returns a list of complex words based on frequency. These terms are determined by selecting the words with frequencies under a specified minimum limit or words that cannot be found in the dictionary. The next step in the pipeline will involve feeding the new list of complex words into a function that performs the task of Substitution Generation. For this pipeline step, I will refer to an updated version of a previously used function called w_embed. This function will utilize the same pre-trained word2vec model from the Gensim library used in the WLSM. Slight variations in this version of w_embed include a larger limit for synonyms that can be generated as well no added selection of shorter words. Like its function in the WLSM, this version of w_embed will return a listed list of synonyms reflecting a given list of complex words. For the step of substitution ranking, I created a new function called rank that takes in a listed list of words and returns a new list of candidate terms for later replacement. During the development of this Model, I chose to apply the concept of word length to substitution rank. Perhaps this method for rank can easily identify the simplest term in a list of synonyms. The resulting list from the rank function will be fed into the final step of substitution replacement. To perform this task I will create a new version of the substitute function used in WLSM. In this version of substitute, it will take in a new list of simple words from the rank function and return a new simplified string sentence. Following the framework of my new pipeline, this model has the potential to be a higher quality lexical simplifier. The main takeaways of this model can be using a dictionary of word frequencies and the use of word length in the rank task of the pipeline. Due to these features, this model will be referred to as, ‘Frequency Length Simplification Model’, or FLSM.

Model 3: Text Frequency Simplification Model(TFSM)

The third model, ‘Text Frequency Simplification Model’, will be very similar to the previous model in its use of a dictionary as well as the Gensim pre-trained word2vec model for substitution generation. This model will still differ in one aspect of implementation compared to FLSM’s structure. This model will start by opening the saved dictionary file like in FLSM, for later use in the new pipeline. The identify function will be used for the task of complex word identification. The identify function will operate the exact same as in FLSM where words with low frequencies are selected as complex. The resulting list of the identify function will then feed into the w_embed function which fulfills the task of Substitution Generation. This function will return a listed list of synonyms based on a given list of complex words. The list of words will then be fed into the next step of the new pipeline, Substitution ranking. Up until this point of the pipeline, this model is identical to the previously developed FLSM. What makes this model unique will be the approach used for this step of the pipeline. The new function for this task is called Rink and will take in a listed list of words and return a list of candidate terms. Unlike the rank function used in FLSM, Rink will return candidate terms based on the highest frequencies in the children’s text dictionary. This approach to ranking could have the potential of improving results when compared to previous methods, as using frequencies to determine simplicity appears to be more effective in word selection tasks. The results of the rink function, when following the new pipeline, will be fed into the step of substitution replacement. The function that will be used for replacement is the same substitute function used in the previous FLSM. It will take in a list of replacement words from the Rink function, and return an updated string with all of the complex terms replaced. Due to this model's use of a dictionary instead of word length, this simplifier will be named ‘Text Frequency Simplification Model’, or TFSM.

Model 4: Text Frequency Thesaurus Simplification Model(TFTSM)

For the final model developed in this project, Text Frequency Thesaurus Simplification Model, slight adjustments are made to the TFSM to improve on a few of the tasks in its structure. Firstly, for the task of complex identification, this model will be utilizing the same identify function in the TFSM, which takes in a given string of text and returns a list of selected complex terms based on the term frequencies. The resulting list of complex words will then be fed into a function that performs substitution generation. What distinguishes this step from the TFSM will be the addition of a second feature extraction method. A thesaurus, NLTK's Wordnet, will be added in conjunction with the pre-trained word2vec model. The function used for this task of the pipeline will be the w_embed function used in all the previous models, but with the addition of Wordnet. Wordnet is used conditionally for cases where certain synonyms cannot be generated from Gensim's pre-trained model. This aims to feed those unique words through the thesaurus to act as a failsafe for synonym generation. This addition will allow for more opportunities to find alternative terms for a given list of complex words. The resulting listed list from the w_embed function will then be fed into the rink function used previously in TFSM. The rink function performs the task of substitution ranking, which will select replacement terms. The rink function will utilize the same dictionary used earlier in the model to identify terms with the highest frequencies from a given list. A list of candidate replacement terms will be fed into the substitution replacement step. However, before that point is reached, another function is added to this model to improve results. In Wordnet, some of the terms have underscores present, which can make their way into the resulting final string. To solve this problem, I decided to create a new function called undescr. This function removes the underscores from a given list of alternative replacement terms and returns a new list devoid of underscores. Underscr is positioned between both steps of ranking and replacement to ensure all of the simplified terms are structurally sound. For substitution replacement, I will use the substitute function previously used in the TFSM. This function will return a simplified string from a given list of simple replacement terms. With the use of both Gensim and Wordnet in this model's substitution generation, this model can possibly be the most optimal for lexical simplification. Due to the use of a thesaurus, as well as the dictionary of term frequencies, this model will be called 'Text Frequency Thesaurus Simplification Model' or TFTSM for short.

Results:

3.1 Testing Data

The results of these models will be derived from an input sample of sentences from both a practice SAT exam as well as a practice New York regents exam. The chosen sentences were randomly selected from different aspects of these exams, including excerpts and answers. I decided to use these exams as a resource for difficult or complex sentences that could have the potential to be simplified. It was difficult to choose where to sample sentences from, but after many attempts both of these practice exams seemed to be the best options due to their details and challenging language. The objective of my simplification models will be to effectively simplify strings of text in a manner that can help relay information from specific fields to a wider audience. In the results of these models, I will aim to find different qualities that could be used for future iterations of my models. I will also look for the differences between the models, to better understand the approaches used. These results will help in assessing and gauging the quality of these models, hopefully ruling out the best model(s) for use and future development.

In this phase of the project I will be testing two sampled sentences from both exams. Though this may be a small sample size, testing these sentences will serve as an opportunity to qualitatively analyze my models. For the SAT sample sentences, the first sentence used will be; 'This conjecture informs her interest in future research missions to the moon.'. The use of special vocabulary such as 'conjecture' as well as the missing context to the sentence make it a good one to test and analyze. The second SAT test sentence will be: 'Most of the comedies end in marriage, with characters returning to their socially dictated gender roles after previously defying them, but there are some notable exceptions.'. The contrasting statements made in this sentence can make its structure more complex, making it ideal for testing. One sentence sampled from the New York Regents exam will be: 'This man must have committed a great and still hidden crime; remorse pushes him to philanthropy.'. This sentence was chosen because of its unique structure of utilizing terms with opposing meanings. The second Regents exam sample sentence is going to be: 'Commercial hunters and trigger-happy sportsmen slaughtered them indiscriminately.'. The complexity of this sentence comes from the use of some terms that may be field specific, making it a good sentence for testing. These four sample sentences will allow basic analysis for the performance of my four models. I for each model I will discussing the most intriguing simplified sentences from each exam.

3.2 WLSM result descriptions:

The word length simplification model will identify and simplify word complexity based on the length of a word in a given sentence. In the development of this model I initially anticipated that it would miss some of the complex terms entirely, because of their shorter lengths. In these cases, the changes in the updated sentences could be very minimal, limiting the chances for information to be received by a user. I also expected some scenarios where the replacement terms would have the wrong tense of a word, hindering the syntax of sentences. Even with these potential drawbacks I still expect the WLSM's output sentences to retain most of its semantic value and structure. I will be focusing on the most distinctive result sentence from each of the exams to highlight different qualities of this model.

For the SAT's results, I chose, 'Most of the comedies end in marriage, with characters returning to their socially dictated gender roles after previously defying them, but there are some notable exceptions.'. The simplified result is: 'Most of the films end in marry, with actors return to their morally dictate gender roles after been defy them, but there are some notable caveat.'. In this sentence, there are a few cases where the replacement term selected seemed like a better alternative to its previous counterpart. The replacement of the terms 'comedies' with 'films' and 'exception' with 'caveats', seem to be better alternatives based on their vaguely similar meanings as well as word length logic. Although these terms are better alternatives, they may not necessarily be simpler or helpful. For the substitute word 'film', I found that while it shares a similar meaning to 'comedies', some of the context was lost without it present, which could affect its comprehensibility. In the case of 'caveat', I found the selection of the term interesting as it isn't necessarily a more simple word, it seems to be a more niche or field specific term. The replacements of 'returning' to 'return' and 'previously' to 'been', are instances of how this model does not always account for syntax and grammar. This is important, as these changes can affect the meaning of a sentence which could further inhibit the quality of updated sentences.

Regardless of some flaws in this updated sentence it still retained the general structure of its more complex counterpart.

The most interesting simplification, using the Regent's samples, came from the sentence : 'This man must have committed a great and still hidden crime; remorse pushes him to philanthropy.'. The resulting simplification is: 'This man must have commit a great and still hidden crime; guilt pushes him to philanthropy'. In this result, the sentence seems to retain most of its structure as well as its meaning, with reasonable replacements. The replacement of the word 'committed' with 'commit' is an instance where the syntax of a word has been changed, which could affect the grammar of the statement. In this case, the replacement term 'commit' is not a bad replacement as it still shares the same meaning as its counterpart. A strong instance of simplification is the replacement of the term 'remorse' with 'guilt'. This simplification is strong because both share very similar meanings, in context the sample sentence the replacement term seems like a very good alternative for the sentence, while maintaining its meaning. So far, the changes made in this simplified sentence do not seem to be as bad as the previous result. Unfortunately there was another issue, the term philanthropy was also selected as a complex term, but it did not have any replacements resulting in its reuse within the simplified sentence. This simplification did well in retaining its semantic value and structure compared to the previous result.

3.3 FLSM result descriptions:

The Frequency Length Simplification Model will utilize the same generation method as WLSM with Gensim's pre-trained word2vec model. The key distinctions between this model and the previous WLSM, will be its use of both a pre-trained dictionary of term frequencies and a word length selection feature. I anticipate that this model will outperform the previous, due to its use of these new improvements. With the use of the pre-trained dictionary this model should be able to identify complex terms in a more effective manner than it would have using word length. This approach can possibly reduce the chances of not locating the correct terms for simplification. The word length feature utilized in this model's substitution rank function, could select very reasonable replacements that are short. Unfortunately, this feature does have some limitations, it can select terms that have incorrect syntax or even different meanings as seen in the results for WLSM. Even with the possibility of some flaws, I believe this model should perform better than its predecessor while still retaining structure. The most significant simplified sentence from both exams will be observed to point out strong and weak points in this model.

The most unique result sentence chosen from the SAT sample, came from the sentence: 'This conjecture informs her interest in future research missions to the moon'. The simplified result was: 'This conjecture asks her interest in future research tasks to the moon.'. The resulting sentence from the model has had only two terms updated. One of these replacements seemed very good while the other does not seem to be an appropriate simplification. The replacement of the word 'missions' with 'tasks' is an appropriate update, since both words have very similar meanings, it does not take away from the message of the sentence. On the other hand, the replacement of the word 'informs' with 'asks' is just incorrect. Both terms, though vaguely similar, are two completely different actions which makes this simplification a clear flaw of the model. This replacement doesn't only affect grammar, it also changes the meaning of the sentence, which is not a goal for any of my models. Another issue I anticipated to be less common was the failure to identify real complex terms. In the test sentence as well as the result sentence, the term ‘conjecture’ was not identified at all. This is unfortunate, as the previous model was able to identify this term based on its length. This is not ideal as the word can be seen as a field specific term that may be too abstract for some users. With problems present in these results, this model still did well in retaining the structure of its input sentences.

The sample sentence from Regents was: ‘This man must have committed a great and still hidden crime; remorse pushes him to philanthropy.’. The frequency length simplification was: ‘This man must have committed a great and still hidden crime; remorse puts him to philanthropy.’. In this simplification, there were only two complex terms selected and replaced compared to the previous model’s instance which had three terms identified from this sentence. In this result text, the only visible change found was the replacement of the term ‘pushes’ with ‘puts’. Though I understand where the similarity comes from between both terms, the update is still syntactically incorrect which affects the grammar of the statement. I am hesitant to mention the semantic effect of this result, since the meaning of the statement still seems consistent. The second replacement is not visible, since the selected complex term ‘philanthropy’ is reused in the resulting simplified sentence. This same issue occurred in the previous model due the limit of Gensim’s word2vec model not having access to some terms. This problem is not ideal, the presence of complex terms in a sentence after simplification can still limit its comprehensibility to readers. This model still did a good job of retaining the structure of the given test sentences.

3.4 TFSM result description:

The Text Frequency Simplification Model will utilize the same pre-trained word2vec model that is used in the previous models. The key distinction of this model will be its use of a pre-trained dictionary of word frequencies to perform multiple tasks during the model’s process.

I believe this new change should make this model perform better than its predecessors. The ranking approach of this model will use the pre-trained dictionary to select the best candidates, this new feature will be key to the effectiveness of this model. This new feature will most likely result in the selection of better alternative terms compared to the previous models. Even with these improvements, this newer model will most likely have some issues during its process. I anticipate that there will be some complex terms the identifier of this model will fail to recognize, much like some of the results from FLSM. I also expect that some terms may not be recognized during the substitution generation task of the model, which will result in the same issue of complex terms present in the results of the model. Regardless of these flaws this model will most likely perform well in retaining the structure of its input sentence as well as selecting better alternative terms for replacement. The most significant resulting sentence from each exam sample will be observed closely.

The sample sentence tested from the SAT sentences will be: 'Most of the comedies end in marriage, with characters returning to their socially dictated gender roles after previously defying them, but there are some notable exceptions.'. The simplification of the sentence is: 'Most of the comedy ends in marriage, with characters returning to their morally governed women character after previously defying them, but there are some notable exceptions.'. In this simplification we see multiple cases where replacements do not have good syntax, some other cases seem like appropriate simplifications. An appropriate simplification would be the replacement of the word 'dictated' with 'governed'. In this instance, the replacement word shares a very similar meaning to its complex counterpart, which retains the sentence's semantics. On the contrary, the replacement of terms like 'gender' with 'women' and 'roles' with 'character' are instances where the semantics of the sentences are at risk. Both replacement terms highlight a potential flaw of misunderstanding context within this model. These replacement terms, to a degree, are related in meanings, but these words don't fit as replacements because they don't have context in relation to the meaning of the sentence. Frequent replacements such as these can hinder the grammar and meaning of given sentences which is not ideal for users as well as the purpose of this project. Regardless of these poor conversions, the structure of the sample sentence is still preserved.

The sample sentence tested from the Regents sentences will be: ‘Commercial hunters and trigger-happy sportsmen slaughtered them indiscriminately.’. The sentence is then simplified to: ‘Commercial hunters and trigger-happy sportsmen slaughtered them indiscriminately.’. This result is not optimal since we expect a degree of simplification to occur with results. Upon analyzing the result, the model did identify two complex words from the text, ‘trigger-happy’ and ‘indiscriminately’, but these terms were not altered. This issue most likely occurs in the substitution generation step of this model which uses the pre-trained word2vec from Gensim. As previously mentioned, this pre-trained model has limitations in identifying words, which can reduce the chances for simplification in some cases. While this model has shown promising results before, this particular result points out the challenges and limitations that can occur with these models. Although the result was not altered, it did not alter the structure of the test sentence.

3.5 TFTSM result description:

The Text Frequency Thesaurus Simplification Model is the most similar in structure to the previously tested Text Frequency simplification model. The main difference of this model, will be its additional use of a thesaurus called Wordnet within the substitution generation task of this model. I anticipate this model will surpass all of its previous iterations in the task of lexical simplification. With the addition of Wordnet, this model should be able to address the common issue found in the previous models. The thesaurus will serve as an additional tool for generating alternative terms during the substitution generation phase of the model. This will ideally reduce the likelihood of complex terms remaining in the simplified sentence. I also anticipate this model's use of a Wordnet will result in better alternative terms for simplification. This model's most noteworthy results from each exam sample will be qualitatively observed.

The chosen SAT test sentence is: 'This conjecture informs her interest in future research missions to the moon.'. The simplification resulted in, 'This conjecture informed her interest in future research tasks to the moon'. In this result there were two replacements made that still retain the grammar and meaning of the test sentence. The replacement of the term 'informs' with 'informed' is a reasonable replacement, the syntax of the word is also acceptable. Unfortunately, the use of this word does impact the tense of the statement, which can potentially affect the intended meaning of the sentence. Although this replacement is not ideal, it is a better alternative than the word 'asks' which was found in the result for the FLSM. The second replacement is actually the same selection that was made by the FLSM, choosing 'tasks' as an alternative for 'missions'. Although both of these models have similar generation functions, their approaches to substitution ranking are completely different, which makes their identical results intriguing. The term 'tasks' might be a common occurrence in the pre-trained dictionary, which resulted in its selection during the ranking of this function. Despite this result having some changes to its tense, the overall update seemed grammatical and structured while still retaining the overall meaning of the statement.

From the Regent's samples, I chose the sentence: ‘Commercial hunters and trigger-happy sportsmen slaughtered them indiscriminately.’. Its resulting simplification was: ‘Commercial hunters and fierce sportsmen slaughtered them randomly.’. In this result, there were two visible replacements made to this sentence, both of which seem to be grammatically and semantically correct. The replacement of the term ‘trigger-happy’ with ‘fierce’ is a very reasonable alternative as it still preserves the statement’s meaning. The same can be said for the replacement of the word ‘indiscriminately’ with ‘randomly’, both words share such similar meanings that the use of either still conveys the same message of the sentence’s unaltered form. For results of this test, the model excelled in finding and selecting the correct alternatives as well as retaining the structure of the sentence.

3.6 Data collection method

To evaluate the performance of each of my models, they have each been tested with a set of twenty sentences picked from the New York Regents exam. Within these sentences I personally identified eighty-seven complex terms, each model will be evaluated based on their abilities to identify and simplify these complex terms. The Chi-squared formula will be used to evaluate the identification capabilities of models as well as simplification performance. This performance will be determined by the number words each model is capable of identifying from the sampled sentences. The number of words identified will then be collected for later analysis. It will be very interesting to observe the varying performances of my models. The identification approaches used in these simplification models have only two variations, one that selects via the length of a word, and others that evaluates which word had the highest frequency count. In the case of my two approaches for complex identification, the Chi-squared test will allow me to evaluate which of these approaches would be the most appropriate based on the data collected. In this analysis I anticipate that the approach of word frequency will be the superior method for identifying complex words. It is also expected that the final model, TFTSM, will most likely be superior at replacing complex terms. TFTSM's approach for generating terms will prove to be superior compared to the previous models with its use of Wordnet, allowing more access to terms for replacement.

3.7 Chi-squared Analysis

Chi-squared is a commonly used approach in statistical analysis, which can make this method suitable for making predictions with explicit data returned by the four models. The Chi-square square statistic will utilize the observations made from the collected data of performance frequencies to return a value that helps confirm predictions. The hypothesis made will be based on observations of the varying performance frequencies. The formula that will be used is:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

For my analysis I will make three different hypotheses based on observations of the collected data, each hypothesis will have a table displaying the observations necessary to perform chi-squared analysis. Every Hypothesis will also display the chi-squared value I have simplified to determine the validity of my claims.

Hypothesis 1: The complex word identification method utilizing word length will be more effective than the rival method of word frequency.

Table: 1	Word Length Identification	Word Frequency Identification	Total
identified	70	41	111
Not identified	17	46	63
Total	87	87	174

Table 1: Identification test

Based on visual observations of the table above we clearly can see that the word length is far more effective in the task of complex word identification than the approach using the word frequency. This is intriguing because it disproves my anticipations that word frequency was the better approach for complex word identification. Due to the size of Table 1 the degree of freedom for this analysis will be equal to one which gives it a critical value of: 3.84. After simplifying the Chi-squared value for Table 1 the result is 20.926. With this result I can say the null hypothesis can be rejected, instead I can accept that there is a statistically significant difference between the better performance of word length identification compared to word frequency identification. This is possible due to the Chi-square having a greater value than the critical value of this table. Based on these results the first hypothesis is validated.

Hypothesis 2: Complex word identification using frequency will have a higher chance of correctly identifying complex words.

Table: 2	Word Length Identification	Word Frequency Identification	Total
Non complex	76	21	97

identified			
Complex identified	70	41	111
Total	146	62	208

Table 2: Identification quality test

The observations made on Table 2 are interesting as they show that word frequency identification is more likely to identify complex words than word length identification. This is significant because it can prove that word frequency identification is a more effective approach for identifying complex words. The size of this table is the same as the previous meaning the critical value for this analysis will also be 3.84. The Chi-squared value for Table 2 results in a value of 5.77. This means the null hypothesis can be also rejected for hypothesis 2. Due to the Chi-square having a greater value than the critical value, there is a statistically significant difference between the better performance of word frequency identification compared to word length identification. The significant difference in performance between both models helps validate my second hypothesis.

Hypothesis 3: TFTSM will outperform all the other models in correctly simplifying terms while retaining a sentence's structure.

Table: 3	WLSM	FLSM	TFSM	TFTSM	Total
Good Replacements	27	8	15	19	69
Bad/same Replacement	43	33	26	22	124
Total	70	41	41	41	193

Table 3: Replacement Quality test

Observing the different categories of Table 3 we can see that many if not all my models are not necessarily the best at selecting go replacements during simplification. This is important as it shows the limits to the models I have created. It can show that each of these models has their own set of problems in regards to performance. Due to the larger size of Table 3 the degree of freedom for this analysis will be equal to three which gives a critical value of: 7.82. After simplification, the Chi-squared value for this table is equivalent to: 6.98. Due to Chi-square having a value less than the critical value of this table there is insufficient evidence to reject the null hypothesis. This means that the hypothesis I made cannot be validated, this can be due to the equally poor performance from all the models or because of how close these results are to each other. Hypothesis 3 is the only prediction that has not been validated by the data.

3.8 Interpretations

According to my analysis most of my predictions have been proven successfully, while some others have surprisingly been proven wrong. In the beginning of the analysis I wanted to evaluate the quality of my distinct identification approaches. Earlier in this project I initially anticipated word frequency would outperform identification, but I was proven wrong. Instead word length identification had proven to excel in the task of identification compared to word frequency. This outcome validates my first hypothesis that word length identification is a more effective approach than word frequency identification. This resulted in the WLSM having seemingly the best capability to identify complex words. Later in my analysis I wanted to evaluate which of my identification approaches were the best at correctly identifying complex terms. For my second hypothesis I predicted that word frequency identification would have a better chance of correctly identifying complex words than word length identification. The results of my analysis have proved this claim, showing that word frequency does in fact have a better ability to identify complexity than utilizing word length. In my final hypothesis I claimed that my Text Frequency Thesaurus Simplification Model, or TFTSM for short, would make the best selection of replacement words compared to all the previous models. After my Chi-squared analysis I found that there was insufficient evidence to support my claim, meaning that none of the models made good replacements or that all the models performed at or near the same level. In my analysis, I conclude that word frequency seems to be the most relevant approach for the task of complex word identification. I also conclude that my models all have the capability to select appropriate words for replacement, although more data will be needed to evaluate which these models perform the best in this task.

Discussions

The Word Length Simplification Model has a few promising strengths, but many weaknesses that should shine a light on potential improvements that can be made in future iterations. A strength of this model was its ability to preserve the general structure of its given sentences, which was helpful for maintaining the semantic value of a given sentence to a degree. Although replacements were not the best at times, the appropriately made selections were reasonable alternatives that still retained semantics. A weak point of this model can be seen in both its identification as well as its replacement. Due to the nature of the word length approach used for identification, this model is prone to incorrectly identifying complex terms due to long word length. This is not ideal as the replacements chosen for these identified terms are typically grammatically incorrect and sometimes even semantically incorrect as well. With issues such as these, the simplified sentences may not even have the same meanings as their counterparts which can compromise and limit the information intended to be shared through these sentences. Moving forward potential improvements can be added to this model that account for words that are not complex but long in length, ignoring more cases such as these can potentially improve the quality of identifications made by this model. Another update for this model can be the addition of a mechanism that checks for the syntax or grammar of the replacement words selected right before replacement. The use of a mechanism that checks for syntax can greatly improve the quality of returned sentences for this model, further improving the comprehensibility of the simplifications. Although this model has several flaws it has inspired the progression of my development during this project.

The performance of the Frequency Length Simplification Model seemed to outperform the previous model in some aspects, while in others it seemed to do worse. A strength of this model was its capability to preserve the structure of sentences, this can help in retaining semantic value to a degree. Another strong point of this model was in its ability to select complex words which seemed more reliable than word length, even though there were some instances of simple terms being identified as complex. A weakness of this model can be seen in its identification of complex words, which at times does not seem to be able to identify complex terms within the sampled sentences. This is not ideal, the returned sentences still containing these complex terms are at risk of misinterpretation, these cases can cause confusion for users. Another flaw of this model was the limited text data used in the pre-trained dictionary to account for different word frequencies. This limit can reduce the chances for this model to successfully identify complex terms from the sampled sentences, or even increase the chance of misidentifying simple terms as complex. A potential update moving forward for this model can be the use of more data for the pre-trained dictionary that is used for this model's identification. With more text data added to the dictionary, the model can identify complex terms from a sentence with more accuracy while also misidentifying words as complex to a lesser extent. The flaws of this model have helped me in making improvements for my future implementations.

The performance of the Text Frequency Simplification Model was better than the results of the previous FLSM, with an improvement made in the selection of its replacement terms. A strength of this model is in its ability to select appropriate alternative words for replacement, which is very useful for preserving the grammar of the simplified sentences. In previous models that performed poorly in this aspect, the returned sentences would often contain a few words that were not syntactically correct. In these cases, the grammar of the sentences were often compromised which could limit the understandability of the returned sentence. This model's ability to rank replacement terms is also better than the previous as the alternatives chosen are more grammatically correct. This is interesting as the previous models seemed to struggle with this aspect of replacement, perhaps it was this model's approach to the task of substitution ranking that improved this model's ability. This model also has a weakness of limited text data within the pre-trained dictionary used for identification. This weakness is shared with previous models as both utilize the same dictionary for identifications. This limit of text data can hinder the models ability to correctly identify complex terms, which can result in some complex terms not being identified. This can potentially result in sentences that still contain complex terms reducing its simplicity. A potential solution for this problem moving forwards can be the use of multiple corpus of text during the pre-processing of the dictionary of frequencies, increasing the identification ability of this model. Although this model shares a flaw with the previous model this newer iteration seems to be moving in the right direction for my next implementation.

The Text Frequency Thesaurus Simplification Model is my most current implementation for lexical simplification, this model seems to outperform the previous models that also utilized a dictionary of frequencies to identify complex words. This simplifier also seems to perform identification better than the word length approach used in the WLSM. A strength of this model was in its task of substitution generation, which had access to an extra resource the previous models did not have, the Wordnet thesaurus. With the use of Wordnet in this model's generation function as well as the previously used pretrained word2vec model from Gensim, this model's generation task will have access to a wider range of words that can potentially be saved as replacement terms. With this extra use of a thesaurus this model was able to find a larger array of alternative words that can be used for replacement, allowing more chances for words to be simplified in sample sentences. Unfortunately a weak point of this model was in its ability to select appropriate replacements, although this model does make better replacements than the previous, this model still runs into the problem though not too often. Although the frequency of bad replacements does not occur as often in this model, it is ideal for future iterations to reduce this tendency of replacement. A potential solution for this problem moving forward can be the inclusion of a mechanism that checks for the readability of a simplified sentence. Such a mechanism can account for terms that would need to be replaced to improve the syntax of a sentence while still preserving the meaning of a given statement. This model's ability to make good replacements has a greater probability than all the previous models including the Word Length Simplification Model, which has made more replacements than any of its succeeding simplification models.

In all the simplification models I have made using my four step pipeline, they all seem to share the same problem that would end up revealing itself in the results. For the task of substitution generation all my models shared a similar problem of limited access to data. The pre-trained Word2Vec model I have used in every one of my generation functions has this flaw of limited data. Through the use of the pretrained word embedding from Gensim, did I realize that the data of which this model was trained on had a limit. The pretrained model I have used was trained on google news, which was my intention for the general word generation. It was not until later in my development that I realized the data grabbed from google news may be limiting in some aspects to the generation task in my models. This was clearly seen when some instances of complex words were never replaced due to the Gensim model not having access to a word or its alternatives. During the development of models I came up with a solution that was intended to alleviate this issue of limited data for my generation task. This solution would then be developed into the TFTSM, which I mention has an additional use of a thesaurus as a resource for text data. To a degree this solution worked, this model is currently my best model due to its use of the Wordnet thesaurus. Although this model is my best one, it is far from complete in its performance. For future implementation, I would like to improve my generation approach, maybe by utilizing a word embedding that has been trained on a larger set of data

Conclusions

Based on my statistical analysis, although my WLSM was more effective in identifying complex terms by quantity, the quality of the identifications it made were not nearly as good as the models that identified complexity based on word frequency. My analysis also points out that there was not enough evidence to say which of my models were the best at making simplified replacements, although based on the probabilities my final TFTSM seemed to perform the best at replacement out of all my models. Looking back to my analysis, it is clear that all my models made more bad replacements than good. Although my models may not have been the best at performing lexical simplification, the experience of this entire project has given me a great perspective on the subjects of Natural Language Processing and Machine Learning. Through the bugs and fixes of my models as well as the research I have done during this project, this experience has truly been a great opportunity for me to apply my skills in computer science to real life situations that could potentially benefit people. The ultimate goal of my models is to simplify, but through the development of this project I have seen that it is more than just simplification. It is a process, one that demands an understanding of word complexity as much as it does for simplicity. Through my research, I have seen the general process taken for the task of simplification utilizing a rigid pipeline of four steps. With this pipeline I learned that simplification was an achievable process. Later in my project I decided to tweak this pipeline for the sake of my own compressibility, although some crucial steps may have been lost due to my tweaks, my new pipeline steps ignited my engagement with the process of simplification. Overall this project can benefit from more testing and more access to text data. The models I have created are all good examples of lexical simplification models. With the main goal of extending access to information to a wider audience.

Bibliography

- [1]Richard Nordquist, “What Are Complex Words in English Grammar and Morphology?,” *ThoughtCo*. https://www.thoughtco.com/what-is-complex-word-1689889#:~:text=In%20English%20grammar%20and%20morphology
- [2]Christy Paluti, *Study.com*, 2022. https://study.com/academy/lesson/morphemes-examples-definition-types.html
- [3]Nick Wright, “plainlanguage.gov | Complex and Abstract Words,” *www.plainlanguage.gov*. https://www.plainlanguage.gov/resources/articles/complex-abstract-words/
- [4]DeepLearning.AI, “Natural Language Processing (NLP) - A Complete Guide,” *www.deeplearning.ai*, Jan. 11, 2023. https://www.deeplearning.ai/resources/natural-language-processing/
- [5]Computer Hope,“What is a Thesaurus?,” *www.computerhope.com*. https://www.computerhope.com/jargon/t/thesauru.htm#:~:text=1. (accessed May 02, 2024). [6]S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*. Beijing Etc.: O’reilly, 2009.
- [7]G. H. Paetzold and L. Specia, “A Survey on Lexical Simplification,” *Journal of Artificial Intelligence Research*, vol. 60, pp. 549–593, Nov. 2017, doi: https://doi.org/10.1613/jair.5526.
- [8]G. H. Paetzold and L. Specia, “Unsupervised Lexical Simplification for Non-Native Speakers,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016, doi: https://doi.org/10.1609/aaai.v30i1.9885.
- [9]L. Zhou *et al.*, “Natural Language Processing Algorithms for Normalizing Expressions of Synonymous Symptoms in Traditional Chinese Medicine,” *Evidence-based Complementary & Alternative Medicine (eCAM)*, pp. 1–12, Oct. 2021, doi: https://doi.org/10.1155/2021/6676607.

Code references:

- Geeksforgeeks: “Python | Check for float string,” *GeeksforGeeks*, May 27, 2019. https://www.geeksforgeeks.org/python-check-for-float-string/ (accessed May 02, 2024).
- json: “Reading and Writing JSON to a File in Python,” *GeeksforGeeks*, Dec. 16, 2019. https://www.geeksforgeeks.org/reading-and-writing-json-to-a-file-in-python/
- Zip(): “Python zip() Function,” *www.w3schools.com*. https://www.w3schools.com/python/ref_func_zip.asp
- Update(): “Python Dictionary update() Method,” *www.w3schools.com*. https://www.w3schools.com/python/ref_dictionary_update.asp
- [NLP with Python]: Removing Punctuation | Pre-processing, “Removing Punctuation | Pre-processing | Natural Language Processing with Python and NLTK,” *www.youtube.com*. https://www.youtube.com/watch?v=9CpID8ZL1IQ (accessed May 02, 2024).

Datasets used:

- [corpus 1]Children stories(children level); “Children Stories Text Corpus,” *www.kaggle.com*. https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus?resource=download (accessed May 02, 2024).
- [corpus 2]Short stories from; https://andonovicmilica.wordpress.com/wp-content/uploads/2018/07/short-stories-for-children.pdf
- [corpus 2]Diary of a Wimpy Kid Double down, Jeff Kinney, 2016; https://thebookshelfbeforeme.files.wordpress.com/2020/04/diary-of-a-wimpy-kid-double-down-by-jeff-kinney.pdf
- [corpus 3]the stoic, John Galsworthy 1920(higher level); J. Galsworthy, “The Stoic.” Accessed: May 02, 2024. [Online]. Available: https://theshortstory.co.uk/devsitegkl/wp-content/uploads/2016/02/Short-stories-by-John-Galsworthy.pdf