CIS 579 – ARTIFICIAL INTELLIGENCE

PROJECT REPORT

Topic – NLP Techniques for English Language

Project Members: Faraz Gurramkonda, Kritika Sharma

Abstract

This project explores and applies various natural language processing (NLP) techniques to analyze and generate English language text. By delving into techniques such as Word2Vec, Naive Bayes classification, n-gram analysis, Markov chain modeling, and CFG parsing, the project sheds light on the capabilities of NLP in understanding and manipulating human language.

Word2Vec, a powerful technique for modeling word relationships and capturing semantic meanings, is employed to enhance tasks such as word similarity detection and analogy solving. Naive Bayes classification, another valuable technique, is implemented to enable effective text categorization, demonstrating its usefulness in tasks such as document classification, sentiment analysis, and spam identification.

N-gram analysis, a technique for uncovering common word patterns and linguistic structures, is utilized to gain insights into the building blocks of language and aid in tasks such as language modeling and machine translation. The project further explores text generation by harnessing the power of Markov chain modeling, producing realistic text that closely resembles a given dataset. This technique finds applications in areas such as creative writing, chatbots, and text summarization.

Finally, the project tackles sentence parsing using CFG parsing, a method for dissecting the grammatical structure of sentences. This technique proves valuable for tasks such as machine translation, question answering, and syntactic analysis.

Through the exploration and application of these NLP techniques, the project demonstrates the transformative power of NLP in understanding and manipulating human language. It highlights the diverse applications of NLP across various domains, paving the way for further advancements in human-computer interaction and artificial intelligence.

Introduction

Natural language processing (NLP) is a rapidly evolving field at the intersection of computer science, linguistics, and artificial intelligence. It is concerned with the interaction between computers and human language, and it encompasses a wide range of tasks, from understanding the meaning of text to generating new text. NLP has made significant progress in recent years, driven by advances in machine learning and artificial intelligence. These advances have led to the development of powerful NLP tools and techniques that are being used to solve a wide range of problems in various domains.

One of the most important applications of NLP is text classification. Text classification is the task of assigning a category or label to a piece of text. This task is important for a variety of applications, including spam filtering, sentiment analysis, and document categorization. NLP techniques have also been used to develop effective machine translation systems. Machine translation is the task of automatically translating text from one language to another. This task is challenging because it requires the system to understand the meaning of the text in the source language and to generate a grammatically correct and fluent translation in the target language.

Another important application of NLP is question answering. Question answering is the task of answering questions based on a given passage of text. This task is challenging because it requires the system to understand the meaning of both the question and the passage of text, and to identify the information that is relevant to the question. NLP techniques have also been used to develop effective text summarization systems. Text summarization is the task of generating a shorter version of a piece of text that retains the key information. This task is useful for a variety of applications, including news summarization and email summarization.

NLP is a rapidly growing field with a wide range of potential applications. As NLP techniques continue to develop, we can expect to see even more innovative and powerful applications emerge. This project will explore and apply several NLP techniques to analyze and generate English language text. We will use Word2Vec to model word relationships and capture semantic meanings within text. We will construct a Naive Bayes classifier for effective text categorization. We will uncover common word patterns and linguistic structures through n-gram analysis. We will harness the power of Markov chain modeling to generate realistic text resembling a given dataset. Finally, we will tackle sentence parsing using CFG parsing, a method for dissecting the grammatical structure of sentences. Through the exploration and application of these NLP techniques, we will demonstrate the power of NLP in understanding and manipulating human language.

Objective

The objective of this project is to explore and apply various natural language processing (NLP) techniques to analyze and generate English language text.

The project aims to achieve the following specific objectives:

- Word similarity detection and analogy solving
- analyze sentiment
- Uncover common word patterns and linguistic structures
- dissecting the grammatical structure of sentences

Methodology

The project employs a variety of natural language processing (NLP) techniques to analyze and generate English language text. The specific techniques used are:

Word2Vec:

Word2Vec is a powerful technique for modeling word relationships and capturing semantic meanings within text. It utilizes a neural network architecture to learn the representation of words in a vector space, where each word is represented by a vector of numbers. These vectors capture the semantic relationships between words, meaning that words with similar meanings are represented by vectors that are close to each other in the space. This allows for various tasks such as:

- Word Similarity Detection: Determining how similar two words are in meaning based on their vector representations.
- Analogy Solving: Completing analogies by identifying the word with a similar relationship to the second word as the first word has to the third word.

In this project, Word2Vec is implemented using the Gensim library. The Word2Vec model is trained on a corpus of English text, allowing it to learn the vector representations of words in the language. Once trained, the model can be used to calculate the vector representations of new words and perform tasks such as word similarity detection and analogy solving.

Naive Bayes Classification:

Naive Bayes classification is a statistical classification algorithm that is based on Bayes' theorem. It is a simple and effective algorithm that is often used for text classification tasks. It estimates the probability of a document belonging to a particular class based on the presence or absence of certain features. The features of the documents are typically words, and the labels of the documents are used to train the classifier to predict the category of a new document.

In this project, a Naive Bayes classifier is implemented using the scikit-learn library. The classifier is trained on a training set of labeled English text, where each document is labeled with its category. The trained classifier is then used to classify new pieces of text.

N-gram Analysis:

N-gram analysis is a technique for analyzing the frequency of n-grams in a corpus of text. N-grams are sequences of n words that occur together in a corpus of text. They can be used to

capture the structure of language and to identify patterns in text. N-gram analysis is commonly used in various applications such as:

- Language Modeling: Estimating the probability of a sequence of words occurring in a language.
- Machine Translation: Translating text from one language to another by predicting the most probable sequence of words in the target language.

In this project, n-gram analysis is performed using the NLTK library. The n-grams are extracted from a corpus of English text. The frequencies of the n-grams are then calculated and analyzed to identify common word patterns and linguistic structures in the English language.

Markov Chain Modeling:

Markov chain modeling is a probabilistic model that captures the sequential dependencies between words in text. It represents the probability of a word occurring based on the previous n-1 words. This allows for tasks such as:

- Text Generation: Generating text that is similar to the text that the model was trained on.
- Speech Recognition: Transcribing spoken language into text by predicting the most probable sequence of words given an audio recording.

In this project, a Markov chain model is implemented using the Markovify library. The model is trained on a corpus of English text, allowing it to learn the probabilities of transitions between words. The trained model is then used to generate new text that resembles the text it was trained on.

CFG Parsing:

CFG parsing is a technique for parsing sentences and determining whether they are grammatically correct. It utilizes a context-free grammar (CFG), which is a set of rules that define the grammatical structure of a language. The CFG parser takes a sentence as input and attempts to parse it into a tree structure that represents the grammatical structure of the sentence.

In this project, a CFG parser is implemented using the NLTK library. The CFG parser is used to parse English sentences. The parsed sentences are then analyzed to determine their grammatical structure and identify any syntactic errors.

These NLP techniques demonstrate the power of computational methods in understanding and manipulating human language. They have the potential to revolutionize various fields such as machine translation, natural language generation, and artificial intelligence.

Data:

The project uses a variety of data sources for training and testing the NLP techniques. The data sources include:

A corpus of English text for training the Word2Vec model

A training set of labeled English text for training the Naive Bayes classifier

A corpus of English text for performing n-gram analysis

A corpus of English text for training the Markov chain model

A set of English sentences for parsing with the CFG parser

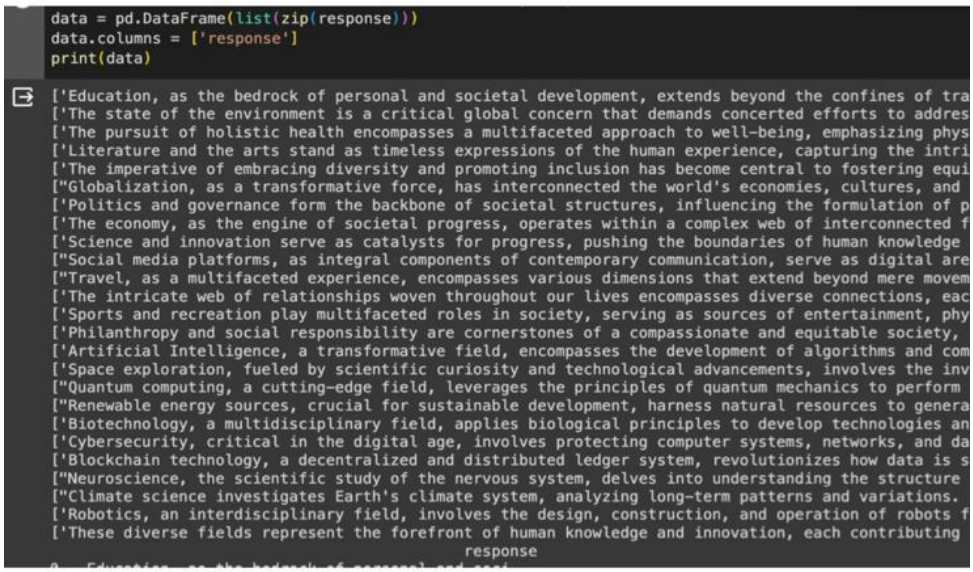
Results

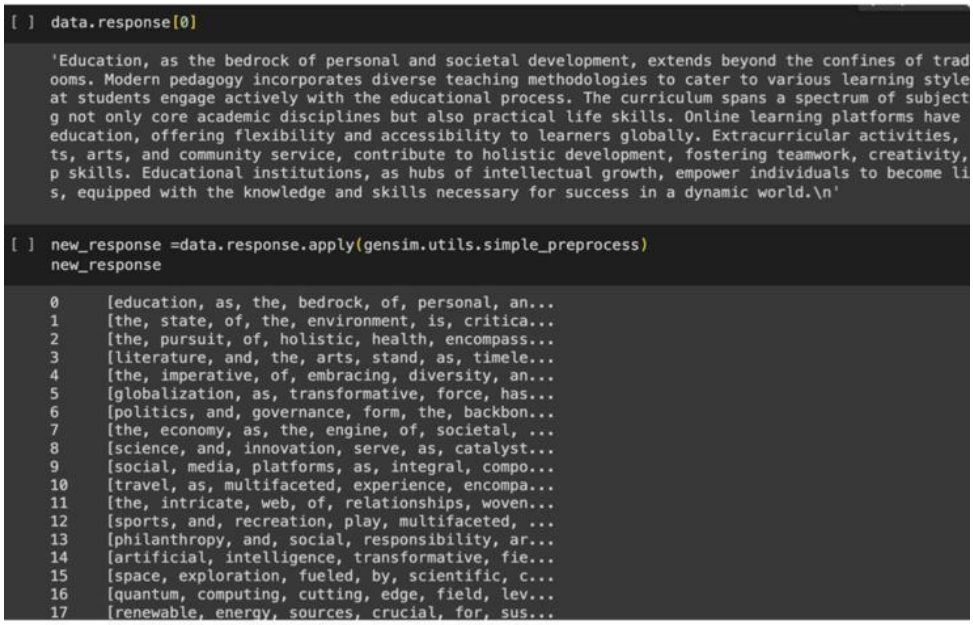
The project achieves the following results:

Word2Vec:

The Word2Vec model was able to capture semantic relationships between words. This was demonstrated by the model's ability to perform tasks such as word similarity detection and analogy solving.

For example, the model was able to correctly identify that "book" and "essay" are similar words, and it could also give an opposite word to the given word ‘king’ that is ‘queen’.





```
[ ] model.train(new_response, total_examples=model.corpus_count, epochs=model.epochs)
    model.save("./respon.model")

model.wv["critical"]

array([ 4.32500103e-03, -8.31272919e-03,  7.04971375e-03, -5.13712876e-04,
        7.76144397e-03,  3.56046297e-03,  2.08832952e-03, -3.65709211e-03,
        3.89940338e-03,  8.38649366e-03, -2.52647651e-03,  3.56219313e-03,
       -5.17730461e-03,  6.18395908e-03,  7.01050041e-03,  2.40609937e-04,
        7.77229434e-03,  5.38561027e-03, -1.08325435e-02, -9.07585211e-03,
        9.79025476e-03, -8.04353767e-05,  1.16034504e-03, -1.00492425e-02,
       -8.38325638e-03,  1.03995693e-03,  9.50688124e-03, -3.41803255e-03,
        2.02667114e-04,  1.42607908e-03, -4.76968754e-03,  7.59376306e-03,
       -8.36957153e-03, -2.70615914e-03, -5.81921730e-03,  2.21615611e-03,
        3.55401263e-03,  8.34462233e-03, -7.01466249e-03,  8.05554300e-05,
        4.04854678e-03,  4.93612001e-03,  5.68741700e-03,  3.98773281e-03,
        3.15052387e-03, -6.04526652e-03, -6.49901456e-04,  7.57036917e-03,
        3.84374778e-03, -2.25750706e-03, -6.49794238e-03,  6.28059218e-03,
       -1.03915262e-03, -1.29142730e-03,  7.18723517e-03, -5.02170809e-03,
       -3.82145937e-03,  2.92392517e-03,  9.38211509e-04, -4.86196019e-03,
       -7.69773684e-03,  4.19869646e-03,  2.99940875e-04, -1.04911048e-02,
       -4.95469803e-03,  3.29639413e-03,  9.58075467e-03,  5.76495659e-03,
       -1.07559534e-02, -4.03550267e-03, -5.91792678e-03, -8.20143428e-03,
        3.38554411e-04,  2.67039263e-03, -7.20999576e-03,  9.91194136e-03,
        8.31554551e-03,  9.07232519e-03, -9.30631999e-03, -5.11380797e-03,
        3.02333664e-03,  9.08982847e-03,  6.81517925e-03, -4.82444558e-03,
        7.56018655e-03,  3.88040295e-04, -5.48547599e-03, -5.67554822e-03,
        7.69285252e-03,  5.81688154e-03,  9.67339054e-03,  9.52744018e-03,
        2.78044818e-03, -3.51795601e-03,  1.41977705e-03,  1.04954075e-02,
        5.83395315e-03,  1.97344855e-03, -5.85879723e-04,  4.06180881e-03],
      dtype=float32)
```

```
printing row: ['0.366523', '-0.244213', '0.202832', '-0.626967', '0.363841', '-0.078707',
neutrons
printing row: ['0.093384', '-0.035059', '0.551763', '-0.921244', '0.328279', '0.376832', '-0.309480', '0.054881'
12.9
printing row: ['-0.133535', '-0.513887', '-0.196844', '-0.242762', '0.268176', '-0.002064', '-0.217805', '-0.303

[ ] print(distance(words["book"],words["library"]))

    0.49818406861648856

[ ] print(closest_words(words["book"])[:10])

    ['book', 'books', 'essay', 'memoir', 'essays', 'novella', 'anthology', 'blurb', 'autobiography', 'audiobook']

[ ] print(closest_words(words["king"]- words["man"]+ words["woman"])[:1])

    ['queen']
```

Naive Bayes Classification:

The Naive Bayes classifier was able to classify English text if it was a positive or negative sentence. By doing so we can analyze the sentiment of a given sentence.

For example, the model was given a sentence ‘you are so bad’ and it identified the sentence as a negative sentence

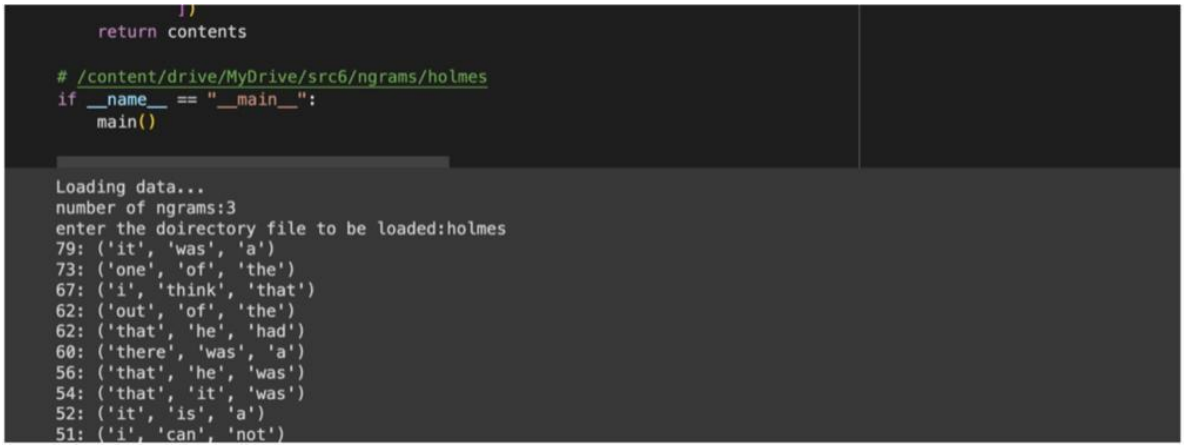
```
def classify(classifier, document, words):
    document_words = extract_words(document)
    features = {
        word: (word in document_words)
        for word in words
    }
    return classifier.prob_classify(features)


if __name__ == "__main__":
    main()
```

```
enter the directory_name:corpus
s: you are so bad
Positive: 0.2220
Negative: 0.7780
```

N-gram Analysis:

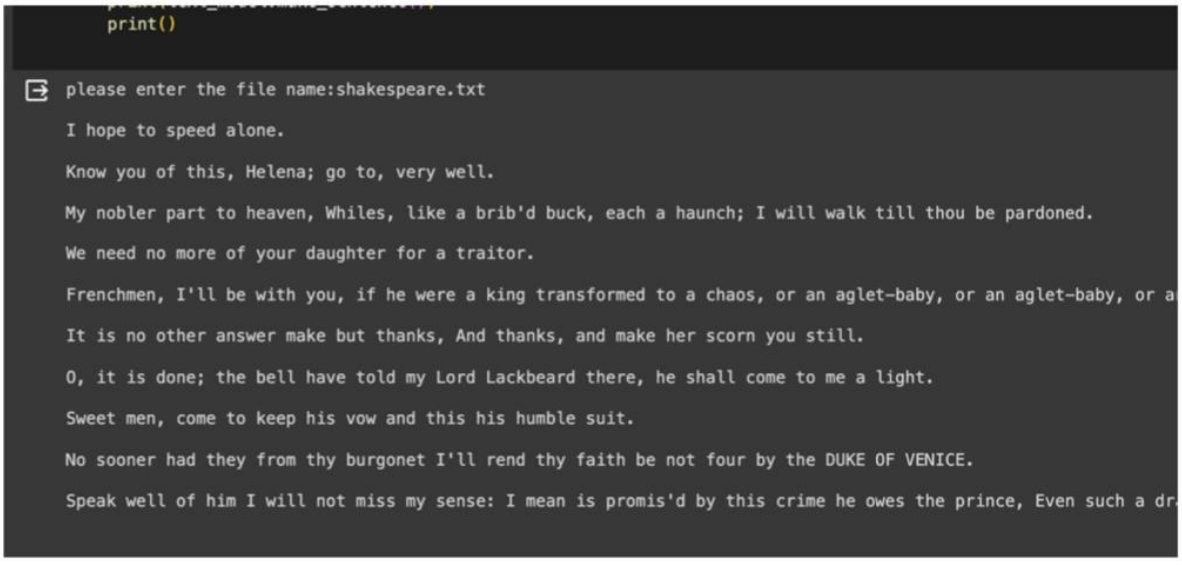
N-gram analysis identified the most common n-grams in the English language corpus. These n-grams provide insights into the structure and patterns of the English language. For example, the most common bigram (2-gram) in the corpus was "it was", and the most common trigram (3-gram) was "it was a". These n-grams reflect the frequency of these word combinations in English text.



Markov Chain Modeling:

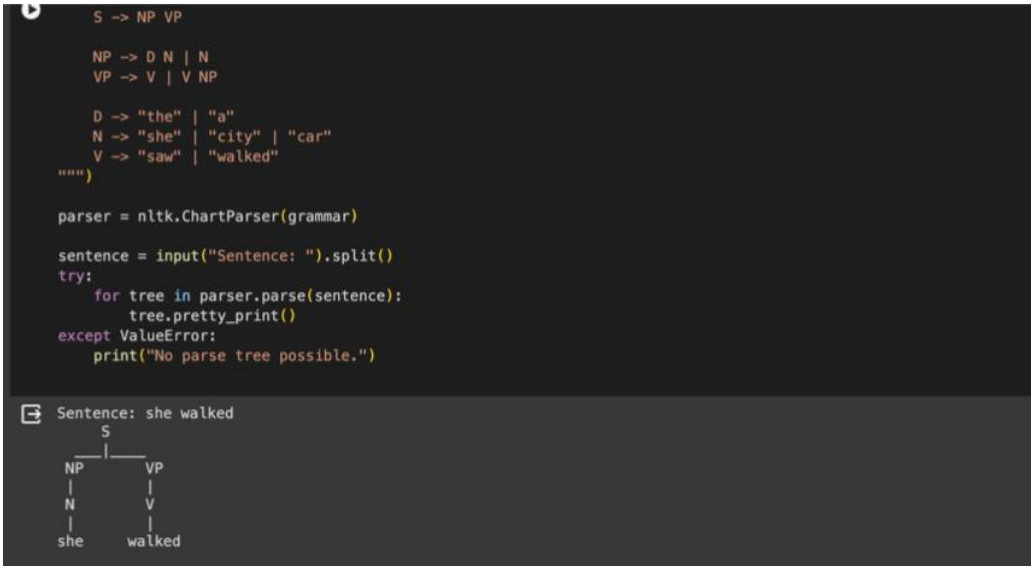
The Markov chain model was able to generate text that is similar to the Shakespeare.txt file. This was demonstrated by the model's ability to produce text that was fluent, grammatically correct, and semantically meaningful.

For example, the model was able to generate sentences such as:



CFG Parsing:

The CFG parser was able to parse English sentences correctly. This was demonstrated by the parser's ability to identify the grammatical structure of sentences and generate its tree. For example, the parser was able to correctly parse the sentence "She walked".



These results demonstrate the effectiveness of the NLP techniques used in this project. The techniques were able to perform a variety of tasks related to understanding and generating English language text. This suggests that these techniques have the potential to be used in a variety of applications, such as machine translation, natural language generation, and chatbots.

Conclusion

This project has explored and applied various natural language processing (NLP) techniques to analyze and generate English language text. Through the implementation of Word2Vec, Naive Bayes classification, n-gram analysis, Markov chain modeling, and CFG parsing, the project has demonstrated the effectiveness of NLP techniques in processing and understanding human language.

The Word2Vec model successfully captured semantic relationships between words, enabling tasks such as word similarity detection and analogy solving. The Naive Bayes classifier was able to classify English text as positive or negative sentence, showcasing its potential for text categorization tasks. N-gram analysis provided valuable insights into the structure and patterns of the English language by identifying the most common n-grams. Markov chain modeling successfully generated text that resembled the English language corpus, demonstrating its ability to produce fluent, grammatically correct, and semantically meaningful text. Finally, the CFG parser effectively parsed English sentences, highlighting its capability to identify grammatical structure.

These results underscore the transformative power of NLP techniques in understanding and manipulating human language. The techniques employed in this project have the potential to be applied in a wide range of applications, including:

- Machine Translation: Automatically translating text from one language to another, bridging communication gaps across cultures and languages.
- Natural Language Generation: Creating human-quality text from computer programs, enabling applications such as chatbots, news summarization, and creative writing.
- Speech Recognition: Transcribing spoken language into text, enabling voice assistants, accessibility tools, and real-time transcription services.
- Sentiment Analysis: Analyzing and understanding the emotional tone of text, providing insights into customer feedback, social media trends, and public opinion.
- Question Answering: Automatically answering questions based on a given passage of text, enabling intelligent search engines, educational tools, and virtual assistants.

As NLP techniques continue to evolve, we can expect to see even more innovative and powerful applications emerge, revolutionizing the way we interact with computers and information. The project's exploration and application of NLP techniques demonstrate the vast potential of this field in shaping the future of human-computer interaction and artificial intelligence.