
# CIS 600: NATURAL LANGUAGE PROCESSING

# QUESTION ANSWERING SYSTEM USING WORD2VEC AND GLOVE

## TEAM MEMBERS


| | |
|---|---|
| Akshay Shirahatti | 341001392 |
| Rahul Parande | 904792531 |
| Yugmi Bhatt | 918182440 |
| Sakshi Sheth | 703086135 |
| Yash Patel | 517958851 |

# Abstract

In the age of digital information overload, efficient retrieval and interpretation of textual data remain an important challenge. This project report describes the creation of an advanced Question Answering (QA) system that employs Natural Language Processing (NLP) approaches, notably the Word2Vec and GloVe models, to improve the accuracy and efficiency of automated question-answer systems. The implemented system's goal is to model semantic links between words in big text corpora, allowing for the extraction of short, relevant replies to user queries across multiple disciplines.This project creates an advanced question answering system employing Natural Language Processing (NLP) techniques, with a primary focus on Word2Vec and GloVe models, in order to increase information retrieval accuracy and efficiency. Using these models, our system generates high-dimensional vector spaces, allowing us to represent text in a way that captures semantic similarities between words. This allows the system to determine the most relevant replies to user inquiries by comparing the cosine similarity of the vectorized form of the questions to prospective answers stored in a comprehensive knowledge base. The method begins with meticulous text preparation to remove noise and standardize incoming data, which is then transformed into contextually relevant embeddings. These embeddings are then utilized to determine similarities and select the most relevant answers. Our solution demonstrates the use of neural network-based embedding techniques to build a strong framework that can considerably improve the accuracy and reaction time of automated question answering systems. The system is designed to be customizable and scalable, making it appropriate for integration into a variety of applications that require automated user interaction, such as digital assistants and customer care. This project describes the architecture, implementation obstacles, and promise for embedding approaches to revolutionize automated answers in NLP applications.

# 1. INTRODUCTION

The information era has resulted in an exponential growth in data creation and access, posing both benefits and challenges for data management and retrieval. Among these issues, the efficient and reliable extraction of important information from massive digital text collections is an essential requirement in a variety of industries, including business, education, and customer service. This need has sparked substantial interest and progress in the subject of Natural Language interpreting (NLP), particularly in the development of systems capable of understanding and interpreting human language in order to offer quick, reliable responses to user requests. Our effort contributes to this emerging field by creating a comprehensive question-answering system that combines two of the most advanced text processing models: Word2Vec and GloVe.

Our methodology is predicated on the idea that a more profound semantic comprehension of text can greatly improve the precision and applicability of the answers that automated systems provide. Google researchers created Word2Vec, a ground-breaking model in this field. Words are embedded into a high-dimensional vector space using a neural network architecture, which captures semantic and contextual word relationships based on co-occurrence in large text corpora. This particular model excels at identifying word similarities, even when there isn't a clear syntactic or usage pattern connection between them.

"Global Vectors for Word Representation," or GloVe, is the model we use in parallel. GloVe, created at Stanford University that combines the advantages of local context window techniques and global matrix factorization, the two primary word embedding methodologies. In this way, it effectively utilizes the statistical data of word co-occurrences throughout the whole corpus to generate embeddings that capture global statistics as well as local contextual information. This dual approach improves the model's capacity to identify minute differences in word usage across contexts, while also enriching the word vectors with more comprehensive contextual meanings.

By integrating these models, our system optimizes the quality of the input text data for analysis by preprocessing it through a series of steps. Stopwords, punctuation, and other non-informative elements are eliminated as part of this preprocessing, and the text is then tokenized and normalized. After the data has been processed, it is fed into the Word2Vec and GloVe models to create word embeddings. These embeddings convert the text into a format that can be used for semantic analysis and comparison.

The core mechanism for calculating the cosine similarity between the vectorized form of the query and the vectors of possible answers is the fundamental method for retrieving answers to user queries. By using this technique, the system is guaranteed to choose the answer that most closely matches the query semantically, increasing the answer's relevance and accuracy. Our system is able to comprehend and react to direct queries as well as interpret the intent and contextual subtleties behind them thanks to this complex model architecture and processing

pipeline. This allows us to provide responses that are both semantically rich and appropriate for the context.

Our NLP system's versatility and applicability in different domains where precise and speedy information retrieval is essential are enhanced by these capabilities. The potential applications are numerous and significant, ranging from powering customer service chatbots that can handle a high volume of inquiries at once to assisting students and researchers in navigating sizable digital libraries. We want to investigate additional integrations, like voice input and multi-modal data processing, as we develop and enhance our system. These could lead to new directions in artificial intelligence and natural language processing research and applications.

## 2. SIGNIFICANCE

The ability to quickly and effectively navigate large repositories and extract relevant information is crucial in an era where data is everywhere. Our project develops a question-answering system that precisely and quickly meets this need by utilizing state-of-the-art Natural Language Processing (NLP) technologies, specifically Word2Vec and GloVe. This development is significant in a number of ways, improving user experience, operational efficiency, and widespread accessibility for information consumption. The following are the key effects of our system:

**Enhanced Information Accessibility:** By quickly obtaining the most relevant and important responses to user inquiries, the system significantly increases the accessibility of information. This is especially important in settings like clinical decision support systems, customer service, and academic research where accurate and timely information is essential.

**Operational Efficiency:** The system minimizes the effort required of customer service representatives and information analysts by automating the answer retrieval process, thereby reducing the need for manual search efforts. As a result, businesses save money and use their resources more effectively.

**Improved User Experience:** The system improves user satisfaction by responding quickly and accurately, which makes it easier for users to use various digital platforms. For businesses looking to enhance customer engagement and interaction through responsive digital services, this capability is priceless.

**Scalability Across Industries:** The system is scalable and suitable for international markets because of the adaptability of the NLP models (Word2Vec and GloVe) used. These models can be customized for a variety of industries and languages. This covers possible applications in e-commerce, learning, medical, and other fields.

**Creative NLP Applications:** This project sets the standard for future NLP research and development by illustrating the usefulness of sophisticated NLP techniques. It offers a starting point for investigating more intricate NLP systems that might include multimodal data integration or multilingual capabilities.

**Data-Driven Insights:** The system can also produce insights into user behavior and preferences by analyzing interactions and queries. These insights can be used to improve service offerings, adjust business strategies, and personalize content.
Education and Learning: By quickly and accurately responding to questions about complicated subjects, this system can help teachers and students in educational settings by giving them instant access to information.

## 3. PROPOSED WORK

The main goal of our project is to develop a next-generation natural language processing system for the purpose of information retrieval and the formulation of user queries. The system will dramatically enhance the interaction of users with a database, with an answer to a query presented in seconds—a crucial factor for today's fast-paced, digital information-driven world. We combined the use of state-of-the-art NLP techniques and robust machine learning models with the sophistication of application frameworks using Flask and Firebase for optimal data storage and management.

**Word2Vec**: Word2Vec is an intrinsic portion of our NLP system. Using a machine learning model and neural network architecture, the model learns the co-occurrences of words from a large corpus of text. Word2Vec produces a vector space of a few hundred dimensions in which all the individual words in the corpus are embedded; it usually uses either one of its two architectures, Continuous Bag of Words (CBOW) or Skip-Gram. Words sharing common contexts in the corpus are placed close in the space, that is, semantically similar words are embedded near each other. The proximity feature of our system allows it to distinguish and leverage both semantic and syntactic similarities between words for enhancing the relevance of search results far beyond mere lexical matches.

**GloVe**: GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm on the aggregated global word-word co-occurrence statistics from a corpus used to learn vector representations of words. The model captures both local and global contextual information effectively during the training process, thus developing word vectors that hypothesize meaningful dimensions in which co-occurring words are often mapped together. The system will be able to infer not just the words in direct contexts but will also use the inferred semantic relationships from the collective statistics of the corpus to improve the quality and relevance of the answer that is retrieved.

**Flask:** We use Flask, a small web development framework, written in Python, to support the operational needs of such deep computational tasks. This allows us to create a lightweight, powerful web application that can handle user requests seamlessly. It supports integration with other Python libraries and tools, which will help you in building high-performing applications that can scale and require real-time data processing and response delivery. Its modular and minimalist design allows for small enhancements and modifications without the overhead of heavier frameworks, therefore adhering to our agile development process.

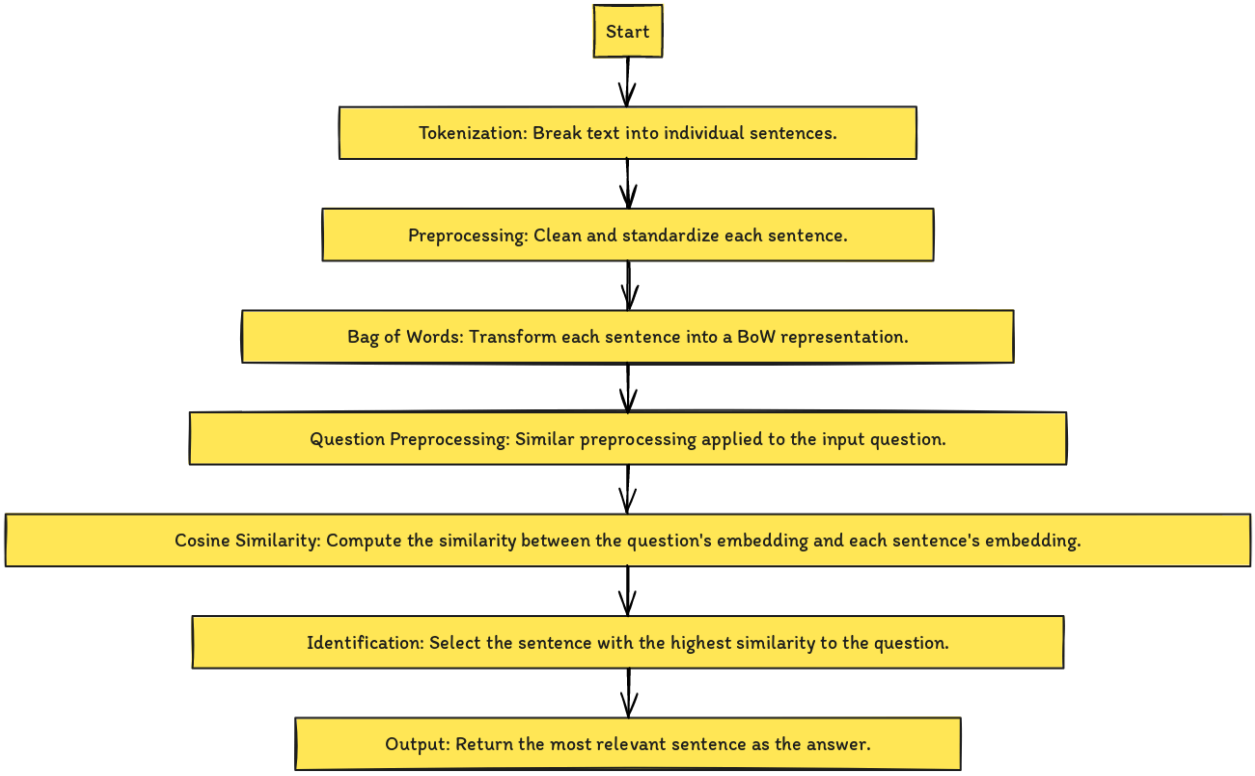
**Firebase:** Firebase is what powers the storage and management of our data. It offers a series of cloud services, including real-time databases and cloud storage, which are important for ensuring the integrity and availability of data that goes through the system. Strong, scalable storage solutions that can handle the dynamic nature of user-generated data and queries with the use of Firebase are availed. More so, its real-time database features make sure that the data is updated in its operations, which is very crucial for the accuracy and relevance of the responses the system gives to its users.

The methodology we have proposed is quite comprehensive and starts at the initial data handling where input from users and data coming from the system are acquired, tokenized to further break long texts down into small and more manageable and analyzable parts. After the data has been pre-processed to remove inconsistencies or irrelevance that may degrade the quality of the system outputs, further analysis is done on the data.

The next stages involve the application of NLP techniques to analyze and understand the content and transform raw data into usable insights. The system concludes with the user interaction phase: the transformed information is delivered back to the user in a concise, precise answer to their query.
This unified approach guarantees that our NLP system is more than simply operational; it is also robust, scalable, and capable of meeting the complex and diverse demands of modern data environments.

**NLP Techniques and Workflow**

The workflow of our question-answering system comprises several detailed steps:




1. **Text Tokenization**

   Text tokenization is a very important first step of our NLP pipeline in dealing with the decomposition of complex textual documents into manageable and analyzable elements. Our system is developed using the Natural Language Toolkit in Python. Text from PDFs is processed, which is mostly further split into sentences. Such an elementary segmentation forms the base for raw data, which is to be further handled for NLP tasks.

   Breaking the big blocks of text down by sentence enables the application of the following NLP techniques to every sentence on its own; hence, our analyses are granular, making our inferences more accurate. This ensures the system's performance correctness in correctly comprehending and addressing a user's question, due to its capability to handle the integrity of the semantic structures provided by text only in individual chunks.

2. **Sentence Preprocessing**

Once the text is tokenized to sentences, every sentence is pre-processed with a full routine that aims at increasing the quality and relevancy of the data entering the main analytical stages of our NLP system. The pre-processing consists of some key steps, each with a reason to refine the text so that it can enable the system to perform better at later stages like vectorization or semantic analysis. Here's a brief description of the steps constituting the pre-processing of a sentence:

**2.1. Removal of Stop Words**

To eliminate common words that appear frequently in the language but do not contribute significantly to the overall semantic meaning of the sentences. Words such as "and", "the", "in", and similar articles, conjunctions, and prepositions are typically removed during this process.

This reduces the data's dimensionality and focuses the analysis on words that have more substantial meanings and are more likely to contribute to the semantic relevance of the text. By concentrating on key words, we enhance the system’s ability to derive meaningful insights from the text.

**2.2. Stripping Non-Alphabetic Characters**

To clean sentences by removing any characters that are not letters, such as numbers, punctuation marks, special symbols, and extraneous spaces. This includes converting contractions and possessives into simpler forms (e.g., turning “children’s” into “children”).

This step helps in standardizing the text format across the dataset, ensuring that only textual data is analyzed. It prevents the algorithm from mistakenly attributing importance to syntactical nuances that do not influence the content's meaning.

**2.3. Text Standardization**

To homogenize the appearance and format of all text data by converting all characters to lowercase. This ensures that the same words in different cases are recognized as identical (e.g., "House" and "house" are treated as the same word).

Standardization eliminates case sensitivity issues in the processing stages, simplifying computational requirements and helping in accurate word frequency counting. It is crucial for maintaining consistency and reliability in subsequent NLP tasks like tokenization and vectorization.

**2.4. Normalizing Text Encoding**

To unify the character encoding across all text data to a standard format (such as UTF-8), ensuring that all characters are represented uniformly in the system.

This step prevents encoding errors that could lead to data loss or misinterpretation during the analysis, especially with non-English characters or special symbols.

**2.5. Correcting Spelling Errors**

To detect and correct typos and misspellings in the text data. This can be achieved using automated spelling correction tools that compare words against a standard dictionary.

Spelling correction enhances the quality of the input data and ensures that semantic analysis does not get distorted by orthographic mistakes. It helps in improving the accuracy of keyword-based searches and contextual analyses.

By systematically cleaning and standardizing the text, we ensure that the subsequent stages of the system, such as vector representation and similarity calculations, are built on a solid foundation of clear and consistent data. This meticulous approach to preprocessing not only streamlines the entire NLP workflow but also significantly boosts the system's analytical accuracy and efficiency.

3. **Bag-of-Words (BoW) Representation**

The Bag of Words (BoW) model is the basic natural language processing (NLP) technique of our system used for the conversion of preprocessed sentences to structured vector format. This model simply breaks the textual content as a collection of individual words, independent of word order and the grammatical relationship between the words, yet still considering the frequency in the text.

It is implemented by a system in which all unique words in the whole text corpus are at first gathered and combined to form a vocabulary list. Every entry in the vocabulary list is associated with a number, which is then termed as the index of that word. Vector is generated for every sentence in the corpus. These vectors have entries up to the amount of words found in the dictionary and they are constructed in such a way that every index in the dictionary is shown. The value in any given dimension of the vector shows the frequency of the particular word's appearance in that sentence. For example, if the word "apple" appears three times in a sentence, and "apple" is indexed at position 5 in the dictionary, the fifth entry in the sentence's vector will be 3.

A bit simplistic, this is deceptively powerful for a few reasons. First, it permits the system to engage in its ability to compare texts at high efficiency. Because every sentence gets mapped onto a vector of numbers, standard mathematical apparatuses can be applied to quickly reckon similarities and divergences. This is crucial for such tasks as document classification, sentiment analysis, and similarity searches.

The model renders texts comparable in a computational format to get quantitative attributes of the content without the need for too much computational complexity for a more profound linguistic analysis. This, therefore, is an approach driven by BoW for enabling the system to function more accurately and effectively in performing the required operations and interpretations of large volumes of text data.

| | |
|---|---|
| Document D1 | *The child makes the dog happy*<br>the: 2, dog: 1, makes: 1, child: 1, happy: 1 |
| Document D2 | *The dog makes the child happy*<br>the: 2, child: 1, makes: 1, dog: 1, happy: 1 |

↓

| | child | dog | happy | makes | the | BoW Vector representations |
|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 2 | **[1,1,1,1,2]** |
| D2 | 1 | 1 | 1 | 1 | 2 | **[1,1,1,1,2]** |

4. **Preprocessing the Input Question**

   Parallel to processing sentences, the input question from the user undergoes a similar preprocessing pipeline. This includes the cleaning and transformation of the query into a BoW model format, ensuring that the question is compatible for direct comparison with the document's text. By applying the same preprocessing and vectorization standards to both the input questions and the document text, our system ensures a level playing field for all textual data, thereby maximizing the accuracy of similarity calculations between user queries and stored information.

5. **Cosine Similarity Calculation**

   After sentence and the user question are transferred into their vectorized forms via the Bag-of-Words (BoW) model, the most important measure step our system does now is to measure the cosine similarity between the question vector to each of the sentence vectors in the document. Cosine similarity is a common measure within text analysis, calculating the cosine of the angle between two vectors. This results in a calculated value between -1 and 1, where 1 indicates absolute alignment or similarity, 0 indicates orthogonality or no similarity, and -1 indicates absolute dissimilarity.

   This metric fits our needs since it is based on vector orientation, not magnitude, and can thus handle texts of varying length and scales. The system will thereafter be able to

measure the semantic similarity between these texts by checking how close two vectors are in orientation with one another, which will represent the user's question and document sentences.

Cosine similarity is quite good in bringing to the fore the most relevant context-sensitive sentences in relation to the user's query, irrespective of length and word frequency in the sentences. It does so by computing the angular proximity of the vectors within a higher-dimensional space, where a word from the dictionary applied in the BoW model represents each dimension. Sentences are selected that are closer in angle to the query vector compared to other sets of vectors.

It will be of great help in information retrieval and document indexing environments, where picking out the most semantically relevant sentences will greatly improve the accuracy and user satisfaction of the query response system. In other words, the system can pick up not only the relevant information but also provide ranks according to the semantic closeness for refined and efficient response to user inquiries.

6. **Identifying the Most Similar Sentence**

   Following the calculation of cosine similarities, the system identifies the sentence with the highest similarity score as the most relevant to the user's query. This step is critical as it pinpoints the exact piece of information that most closely matches the user's needs, based on the semantic content of the query. The selection of the most similar sentence is instrumental in providing precise and contextually appropriate responses to users, which enhances user satisfaction and system usability.

7. **Returning the Answer**

   Finally, the sentence identified as the most relevant is presented back to the user as the answer to their query, completing the search cycle. This step not only marks the culmination of the NLP process but also symbolizes the system's success in effectively bridging the gap between user queries and accurate information retrieval. By delivering concise and precise answers, our system fulfills its purpose of enhancing the user experience in digital information environments, making it a valuable tool in any data-driven application.

# 4. LITERATURE REVIEW

The rise of Natural Language Processing (NLP) has completely changed how we interact with computers, bringing about a new era where machines can communicate with us in a way that feels remarkably human. Over the years, researchers and professionals alike have dived deep into the world of NLP, eager to see how it can improve different areas, from making tasks easier to enhancing our experiences and sparking new ideas.

One key aspect of NLP is word embedding techniques, which are crucial for turning text into numbers so computers can understand meaning and make comparisons. Among these techniques, Word2Vec and GloVe have become quite popular, allowing us to capture the relationships between words in large bodies of text. Word2Vec, introduced by [i] Mikolov and his team in 2013, learns about words based on how they're used in context, while GloVe, suggested by [ii] Pennington and his colleagues in 2014, looks at global patterns in word usage.

Building on this foundation, our project aims to create a sophisticated Question Answering System (QAS) using NLP techniques, specifically focusing on finding answers to common questions. By carefully preparing text and using Word2Vec and GloVe, we're working on making a system that can quickly give accurate answers to questions.

These systems are incredibly important for making tasks easier and improving how things work. Research by [iii] Lee and others in 2018 shows how NLP is changing customer support, making it easier for companies to answer questions and solve problems, which ultimately makes customers happier.

Looking ahead, our project wants to keep exploring better ways of embedding words. We're interested in advanced models like FastText, as suggested by [iv] Bojanowski and his team in 2017, or contextual embeddings like BERT and ELMO, suggested by [v] Devlin et al. and [vi] Peters et al. in 2018. We believe these techniques can help us understand language even better and give more accurate answers.

We also want to connect our system with chatbots, so it's easier for people to get answers no matter where they are. Research by [vii] Liu and others in 2016 shows how chatbots can make interactions more personal and helpful.

And that's not all – we're also thinking about personalizing recommendations based on what people ask and what they like, as suggested by [viii] Hu et al. in 2008. By tailoring answers to each person's preferences, we can make sure they're getting the most useful information.

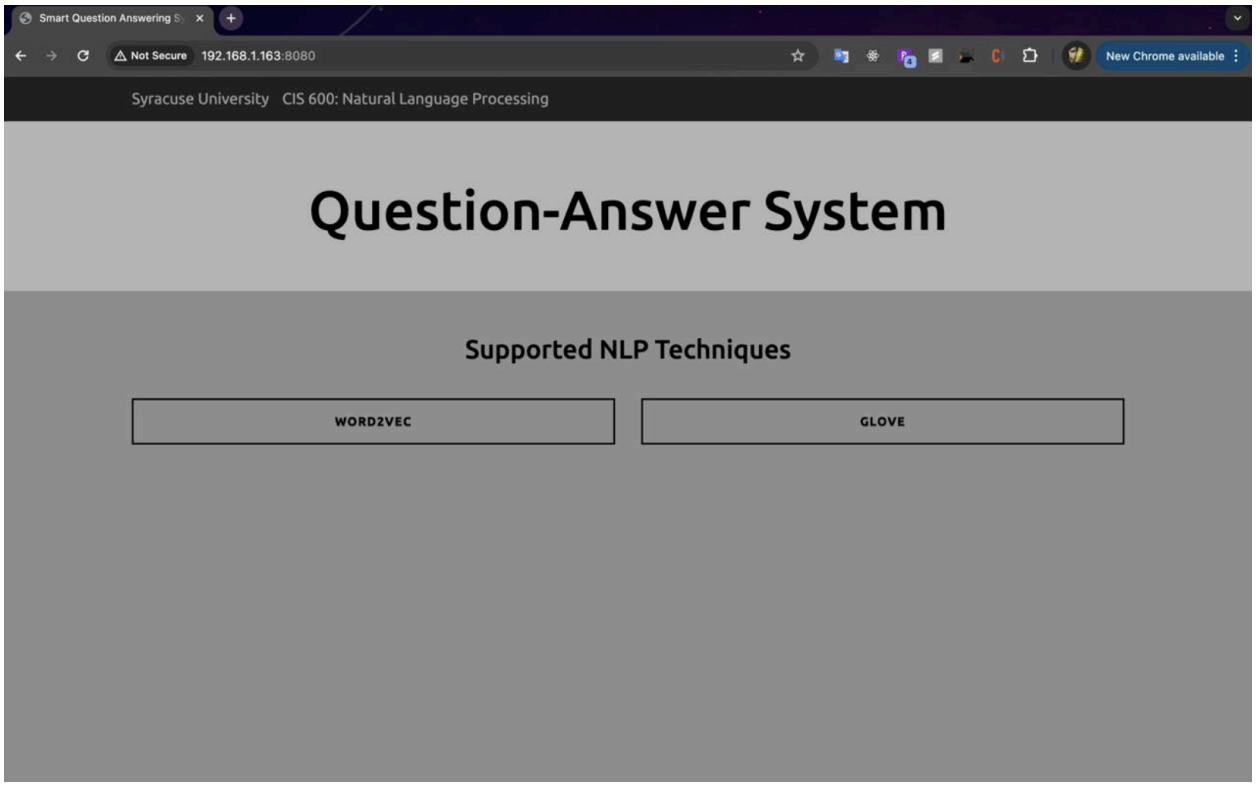
Lastly, we're looking into ways to handle different types of information, like text, pictures, and sound, so we can give even richer answers. Research by [ix] Kiela and others in 2018 shows how combining different types of data can make interactions more meaningful.

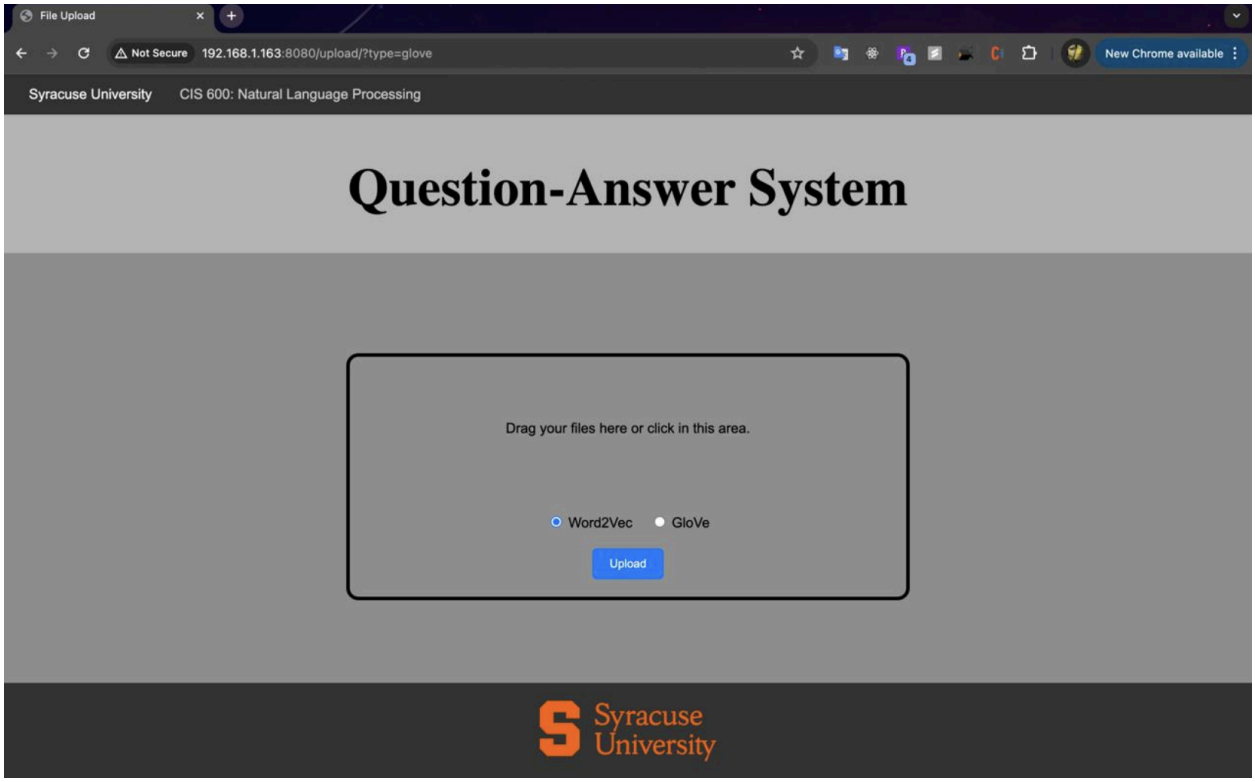
# 5. RESULTS

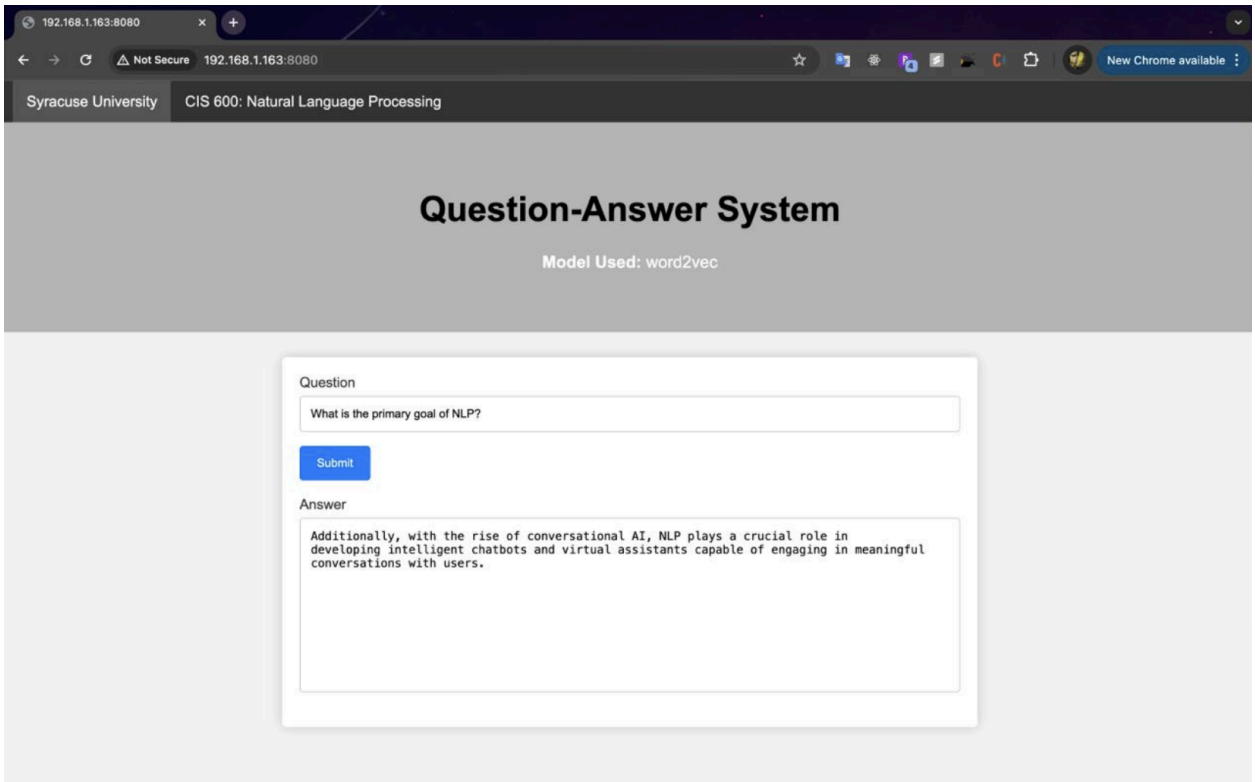
This section covers the project's results, which include the development and evaluation of a Question Answering System that makes use of cutting-edge Natural Language Processing (NLP) methods. Through meticulous trial and analysis, how our system efficiently extracts relevant responses to user inquiries from a database of frequently requested questions.After further testing, it was evident that GloVe performed better than Word2Vec. GloVe's broader perspective and global statistics-driven approach, in contrast to Word2Vec, which depended on local context, produced more coherent and accurate responses by comprehending deeper semantic linkages across the entire dataset. This highlights how important it is to use appropriate word embedding strategies based on the features of the dataset in order to get the best results in NLP applications.

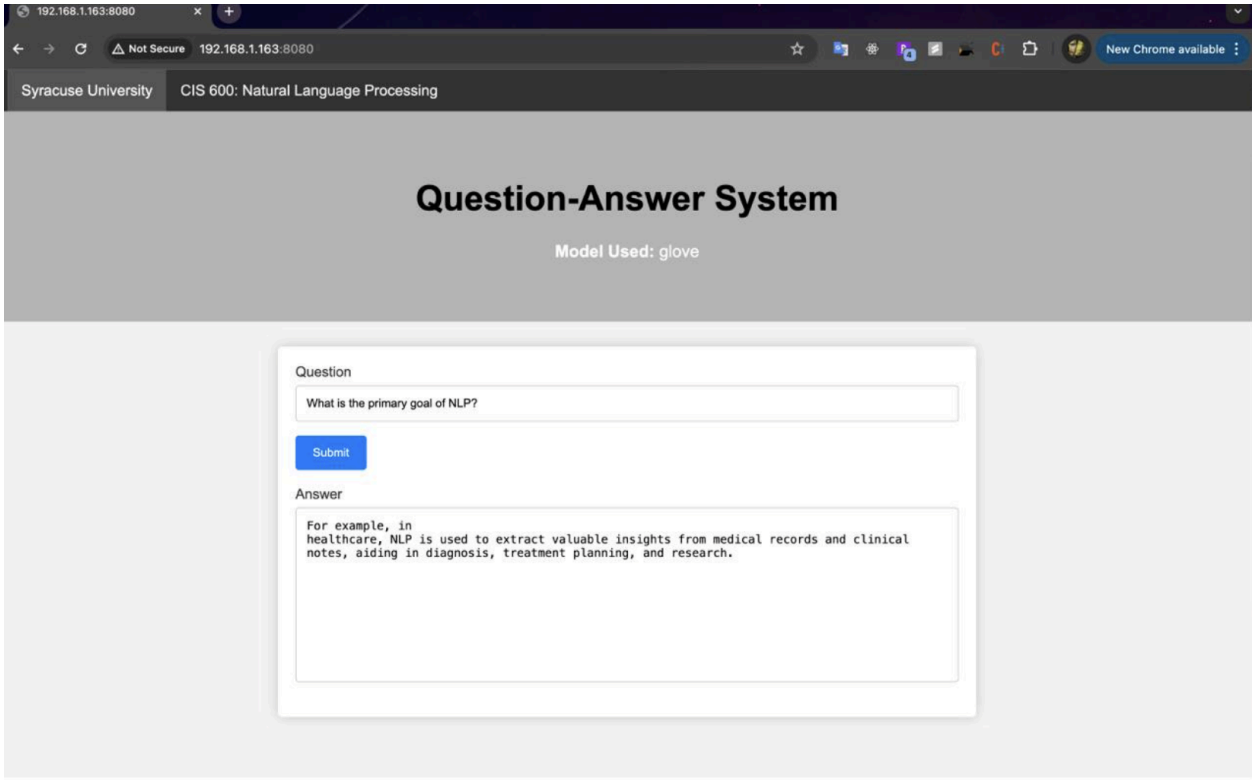
We have developed a Flask application which serves as a question-answering system that allows users to upload PDF documents and ask questions about the content of those documents. The application supports two types of models for answering questions: Word2Vec and GloVe.

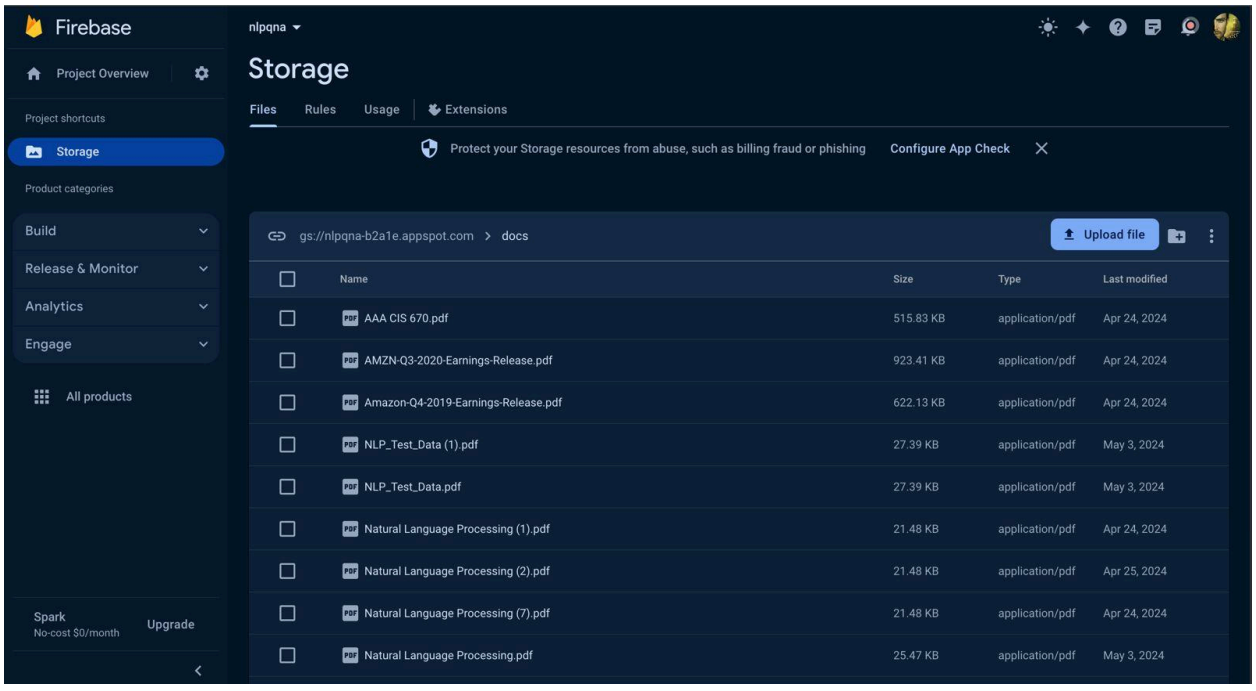
Below are the screenshots of our simple interface:




File Upload
Not Secure
192.168.1.163:8080/upload/?type=glove
New Chrome available
Syracuse University
CIS 600: Natural Language Processing
Question-Answer System
Drag your files here or click in this area.
Word2Vec
GloVe
Upload
Syracuse University


192.168.1.163:8080
Not Secure
192.168.1.163:8080
New Chrome available
Syracuse University
CIS 600: Natural Language Processing
Question-Answer System
Model Used: word2vec
Question
What is the primary goal of NLP?
Submit
Answer
Additionally, with the rise of conversational AI, NLP plays a crucial role in developing intelligent chatbots and virtual assistants capable of engaging in meaningful conversations with users.


192.168.1.163:8080
Not Secure
192.168.1.163:8080
New Chrome available
Syracuse University
CIS 600: Natural Language Processing
Question-Answer System
Model Used: glove
Question
What is the primary goal of NLP?
Submit
Answer
For example, in
healthcare, NLP is used to extract valuable insights from medical records and clinical
notes, aiding in diagnosis, treatment planning, and research.


Firebase
nlpqna
Project Overview
Project shortcuts
Storage
Product categories
Build
Release & Monitor
Analytics
Engage
All products
Spark
No-cost $0/month
Upgrade
Storage
Files
Rules
Usage
Extensions
Protect your Storage resources from abuse, such as billing fraud or phishing
Configure App Check
gs://nlpqna-b2a1e.appspot.com > docs
Upload file
Name
Size
Type
Last modified
AAA CIS 670.pdf 515.83 KB application/pdf Apr 24, 2024
AMZN-Q3-2020-Earnings-Release.pdf 923.41 KB application/pdf Apr 24, 2024
Amazon-Q4-2019-Earnings-Release.pdf 622.13 KB application/pdf Apr 24, 2024
NLP_Test_Data (1).pdf 27.39 KB application/pdf May 3, 2024
NLP_Test_Data.pdf 27.39 KB application/pdf May 3, 2024
Natural Language Processing (1).pdf 21.48 KB application/pdf Apr 24, 2024
Natural Language Processing (2).pdf 21.48 KB application/pdf Apr 25, 2024
Natural Language Processing (7).pdf 21.48 KB application/pdf Apr 24, 2024
Natural Language Processing.pdf 25.47 KB application/pdf May 3, 2024

# 6. CONCLUSION

As we conclude, our project epitomizes the profound influence of Natural Language Processing (NLP) in revolutionizing how we streamline tasks, optimize operations, and enhance user interactions across various domains. By skillfully leveraging state-of-the-art word embedding techniques like Word2Vec and GloVe, we've created a robust Question Answering System capable of navigating through a pool of FAQs with finesse to retrieve relevant answers.

The practical implementation of our system, exemplified through its intuitive interface and seamless integration with Firebase database, underscores its tangible advantages and user-friendly design. By empowering users to effortlessly pose queries and receive accurate responses, our system significantly simplifies the information retrieval process, leading to newfound efficiencies and productivity.

Moreover, our journey has unveiled exciting prospects for further exploration. Delving deeper into advanced embedding methodologies and contemplating their fusion with chatbot platforms opens up boundless opportunities. By incorporating contextual embeddings like BERT and FastText, we envision enhancing the understanding of user queries, thereby refining the accuracy and relevance of our responses.

Moreover, the possibility of incorporating customized recommendation systems and extending our data processing skills to include multi-modal inputs is extremely promising. This provides a means of customizing responses based on user preferences and needs, which in turn raises user satisfaction and encourages more in-depth interaction.

All things considered, our work is a shining example of advancement in the field of natural language processing-based Q&A systems. It displays the progress we've made in increasing automation, operational effectiveness, and user pleasure in a variety of applications. Our unwavering passion to expanding the frontiers of innovation in Natural Language Processing continues to propel us forward, motivated by our desire to provide meaningful answers to pressing real-world problems.

# 7. ACKNOWLEDGMENT

We express our sincere gratitude to Professor Edmund Yu and the Teaching Assistants for their unwavering support and guidance throughout the assessment process. Their invaluable input and expertise greatly influenced both the overall direction and the specific content of this work. Without their assistance, this project would not have been possible.

Furthermore, we extend our thanks to the faculty and staff of the Engineering and Computer Science Department at Syracuse University for fostering an environment conducive to project development and providing us with the necessary resources to fulfill our study objectives.

# 8. References

[i] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[ii] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[iii] Lee, J., Kim, K., Lee, S., & Lee, G. G. (2018). Question answering system for customer service based on deep learning. Applied Sciences, 8(7), 1245.

[iv] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 135-146.

[v] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Bidirectional encoder representations from transformers. arXiv preprint arXiv:1810.04805.

[vi] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

[vii ]Liu, C., Lowe, R., Serban, I. V., Noseworthy, M., Charlin, L., & Pineau, J. (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. arXiv preprint arXiv:1603.08023.

[viii] Hu, R., Pu, P., & Chen, L. (2008). A study of user satisfaction in using natural language for e-mail querying. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 17-20).

[ix] Kiela, D., Firoiu, L., & Riedel, S. (2018). Efficient large-scale multimodal classification. arXiv preprint arXiv:1802.02977.