Assistant to the Holy Quran

Natural Language Processing | CCAI 413

Final Report

Student	ID
Elaf Almalki	2111732
Lama Ali	2112125
Layan Refaee	2115867

By Dr. Latifa Aljiffry

Abstract:

Sometimes you want to search for a specific verse in the Qur'an, but you will not find it quickly if you do not know the name of the surah, the page number, or any other information about the verse.

Or when you want to find a specific verse but you only remember the beginning of the verse and want to find a continuation of the verse.

Our system solves this problem by using an ML model and a DL model that predicts the completion of the verse and prints the exact location of the verse from the Qur'an to make it easier for you to find it.

- **Introduction:** : This project focuses on enhancing the accessibility of the Holy Quran by leveraging natural language processing techniques. Our goal is to develop a system that allows users to search for specific verses efficiently, even with limited information such as a partial verse or without knowing the surah's name. By utilizing both machine learning (ML) and deep learning (DL) models, our system predicts the continuation of verses and pinpoints their exact locations, significantly improving user interaction with the Quranic text.

- **Background/Related Work:** Our study is grounded in the rich field of NLP applications within religious texts, particularly the Quran. Previous works have utilized various NLP techniques for text retrieval, interpretation, and the semantic analysis of religious texts. However, the unique challenge of predicting and locating verses based on partial information remains less explored. The dataset used includes 6,236 verses across 114 Surahs, sourced from the Quranic Complex's developers platform. This comprehensive dataset allows for robust model training and testing.

Approach: Our approach involves two main models:

1.N-gram model : Utilizing 3-gram models to handle predictive tasks based on statistical probabilities of text sequences.
2.Deep Learning-based Model : Implementing Long Short-Term Memory (LSTM) networks to capture longer dependencies in the text for more accurate predictions of verse completions.
The selection of these models was based on their proven efficacy in similar NLP tasks, where they provide a robust framework for both contextual understanding and text generation.

And the main function:
1.return information function: this function use TF-IDF to return Information (location of verse) to user. we used it in both machine and deep learning code

- **Experiments:** The experiments were structured as follows:

- **Dataset**: The Quran text, structured with metadata including Surah names, verse numbers, and page locations.
- **Configuration**: Models were trained with specified hyperparameters, with the LSTM model requiring longer training due to its complexity.
- **Evaluation Metrics**: Accuracy of verse prediction and location identification was measured, comparing the performance of statistical models against deep learning models.
- **Results**: Both models demonstrated strengths in different aspects of the task. The LSTM and 3-gram models were faster in training and performed well with shorter texts, while the LSTM model excelled in understanding and generating longer text sequences.
- **Comparison**: These are the results of N-gram Model and LSTM Model. As we can see, N-gram Model is much better and predicts better.

Deep learning :

1-

```
start_word = input("أدخل كلمة لبدء أنشاء الايه:")  # Prompt the user to enter a word to start the ayah
generated = generate(model, tokenizer, start_word, num_words=1)

query =generated
search_results = retrieve_verse_information(query, dataset)
print("Generated ayah :",generated)
print(f"Text of the verse: {result['Document']}")
print(f"{result['Information']}")
```

```
أدخل كلمة لبدء أنشاء الايه:إياك نعبد وإياك
Generated ayah : إياك نعبد وإياك فاستعينوه
Text of the verse: إياك نعبد وإياك نستعين
aya in jozz: 1
aya in page: 1
aya number in Quran: 1
this aya in sura: الفاتحة
start lin of aya: 5 - End line of aya: 5
aya number: 5
aya Tafser:  ز له ان يصرف شيئا من انواع العبادة كالدعاء والاستغاثة والذبح والطواف الا لله وحده وفيها شفاء القلوب من داء التعلق بغير الله ومن امراض الرياء والعجب والكبرياء
```

2-

```
start_word = input("أدخل كلمة لبدء أنشاء الايه:")  # Prompt the user to enter a word to start the ayah
generated = generate(model, tokenizer, start_word, num_words=1)

query =generated
search_results = retrieve_verse_information(query, dataset)
print("Generated ayah :",generated)
print(f"Text of the verse: {result['Document']}")
print(f"{result['Information']}")
```

```
أدخل كلمة لبدء أنشاء الايه:قل أعوذ برب
Generated ayah : قل أعوذ برب الصلاة
```

3-

```
start_word = input("أدخل كلمة لبدء أنشاء الايه:")  # Prompt the user to enter a word to start the ayah
generated = generate(model, tokenizer, start_word, num_words=1)

query =generated
search_results = retrieve_verse_information(query, dataset)
print("Generated ayah :",generated)
print(f"Text of the verse: {result['Document']}")
print(f"{result['Information']}")
```

```
أدخل كلمة لبدء أنشاء الايه:قل هو الله
Generated ayah : قل هو الله يبدأ
```

machine learning :

1-

```
start_word = input("أدخل كلمة لبدء انشاء الايه:")  # Prompt the user to enter a word to start the ayah
generated = generate(model, tokenizer, start_word, num_words=1)

query =generated
search_results = retrieve_verse_information(query, dataset)
print("Generated ayah :",generated)
print(f"Text of the verse: {result['Document']}")
print(f"{result['Information']}")
```

```
أدخل كلمة لبدء أنشاء الايه:إياك نعبد وإياك
Generated ayah : إياك نعبد وإياك نستعين
Text of the verse: إياك نعبد وإياك نستعين
aya in jozz: 1
aya in page: 1
aya number in Quran: 1
this aya in sura: الفاتحة
start lin of aya: 5 - End line of aya: 5
aya number: 5
aya Tafser:  ز له ان يصرف شيئا من انواع العبادة كالدعاء والاستعانة والذبح والطواف الا لله وحده وفيها شفاء القلوب من داء التعلق بغير الله ومن امراض الرياء والعجب والكبرياء
```

2-

```
start_word = input("أدخل كلمة لبدء انشاء الايه: ")  # Prompt the user to enter a word to start the aya
#print('The ayah')
aya = generate_aya(start_word)  # Generate the aya using the input word as the starting point

query =aya
search_results = retrieve_verse_information(query, dataset)
print(f"Text of the verse: {result['Document']}")
print(f"{result['Information']}")
```

```
أدخل كلمة لبدء انشاء الايه: قل أعوذ برب
قل أعوذ برب الناس
```

3-

```
start_word = input("أدخل كلمة لبدء انشاء الايه: ")  # Prompt the user to enter a word to start the aya
#print('The ayah')
aya = generate_aya(start_word)  # Generate the aya using the input word as the starting point

query =aya
search_results = retrieve_verse_information(query, dataset)
print(f"Text of the verse: {result['Document']}")
print(f"{result['Information']}")
```

```
أدخل كلمة لبدء انشاء الايه: قل هو الله
قل هو الله أحد
Text of the verse:
```

- **Conclusion:** The project successfully demonstrated the potential of NLP techniques in improving access and interaction with religious texts. Our models effectively predict and locate Quranic verses,

facilitating a better user experience. Future work could explore the integration of these models into mobile applications for wider accessibility and the incorporation of multilingual support to cater to non-Arabic speakers.

Team Contribution:

We all worked on the project together, but each of us

focused on a specific part of the project:

The work was divided among us:

Lama Ali-preprocessing & TF-IDF function

Elaf Almalki-generate model & 3-gram model

Layan Refaee-LSTM model

References:

- https://www.semrush.com/blog/tf-idf/
- https://qurancomplex.gov.sa/techquran/dev/
- https://web.stanford.edu/~jurafsky/slp3/3.pdf