UNIVERSITY OF LEEDS

Proceedings Paper:
Alshammeri, M, Atwell, E orcid.org/0000-0001-9395-3764 and Alsalka, MA orcid.org/0000-0003-3335-1918 (2020) Quranic Topic Modelling Using Paragraph Vectors. In: Advances in Intelligent Systems and Computing. 2020 Intelligent Systems Conference (IntelliSys), 03-04 Sep 2020, Online. Springer Verlag , pp. 218-230. ISBN 978-3-030-55186-5

https://doi.org/10.1007/978-3-030-55187-2_19

Quranic Topic Modelling Using Paragraph Vectors

Menwa Alshammeri[1,2], Eric Atwell[1], and Mhd Ammar Alsalka[1]

[1] University of Leeds, Leeds LS2 9JT, UK,
scmhka@leeds.ac.uk, e.s.atwell@leeds.ac.uk, m.a.alsalka@leeds.ac.uk
[2] Jouf University, Saudi Arabia

Abstract. The Quran is known for its linguistic and spiritual value. It comprises knowledge and topics that govern different aspects of people's life. Acquiring and encoding this knowledge is not a trivial task due to the overlapping of meanings over its documents and passages. Analysing a text like the Quran requires learning approaches that go beyond word level to achieve sentence level representation. Thus, in this work, we follow a deep learning approach: paragraph vector to learn an informative representation of Quranic Verses . We use a recent breakthrough in embeddings that maps the passages of the Quran to vector representation that preserves more semantic and syntactic information. These vectors can be used as inputs for machine learning models, and leveraged for the topic analysis. Moreover, we evaluated the derived clusters of related verses against a tagged corpus, to add more significance to our conclusions. Using the paragraph vectors model, we managed to generate a document embedding space that model and explain word distribution in the Holy Quran. The dimensions in the space represent the semantic structure in the data and ultimately help to identify main topics and concepts in the text.

Keywords: Holy Quran, Semantic analysis, Distributional Representation, Topic modeling, Deep learning, Document embedding, Paragraph Vector

1 Introduction

The Holy Quran is a significant resource that is very rich of patterns, topics, and information that make the core of the correct pure knowledge of Muslims. Analyzing the Quran requires special skills and a great deal of effort to get a comprehensive understanding of its meanings, gain useful knowledge, and ultimately build a robust resource for religious scholars, educators, and the public to understand and learn the Quran. The richness of the Quranic text and the deep layers of its meaning offer immense potentials for further study and experiments. Analyzing the Quranic text is not a trivial task due to the overlapping of its meanings. Thus, extracting the implied connections would require deep semantic analysis and domain knowledge.

The Quran has been the subject of many NLP studies. Several studies were related to text mining and topic modeling with the Quran (for a recent survey, see [3]). Previous studies have explored the underlying knowledge of the holy book at different granularities. Scholars and researchers have built applications and tools that exploit such knowledge to allow search in the text. However, they all use different approaches to extract the information needed for their task. Many studies were devoted to topic modeling of the Quran. Most of these works used Latent Dirichlet Allocation LDA as the topic modeling algorithm. Nevertheless, the topic models do not always produce accurate results, and sometimes their findings are misleading [14].

Computational approaches for representing the knowledge encoded in texts play a central role in obtaining the deeper understanding of the content of natural language texts. A recent trend in machine intelligence is the use of distributed representation for words [7] and documents [8] as these representations work well in practice. Several researches have developed distributed word representations in Arabic as well [2]. Word embedding is a modern approach for representing text where individual words are represented as real-valued vectors in a predefined vector space. These representations preserve more semantic and syntactic information on words, leading to improved performance in NLP tasks. They offer richer representation of text that is the base for various machine learning models [20].

Analyzing a text like the Quran requires powerful learning approaches that go beyond word level to achieve phrase level or sentence level representation. Document embedding is a powerful approach, that is a direct extension of word embedding. It maps documents to informative vector representations. Paragraph vectors [8] is a recent breakthrough on feature embedding that has been proposed as an unsupervised method for learning distributed representations for sentences and documents. The model is capable of capturing many document semantics in dense vectors that can be used as input to many machine learning algorithms.

In this work, we used the paragraph vector: an unsupervised document embedding model, to learn an informative representation of Quranic verses. Thus, transforming the text data into features to act as inputs for machine learning models. We utilize the derived features for clustering the verses of the Quran with the final goal of topic modeling of the Quran. Having a good representation of short text like the Quran can benefit the semantic understanding and inferring coherent topics, ultimately identifying inspiring patterns and details that deliver the pure knowledge of the sacred text.

This paper is organized as the following: Section 2. reviews related work. The methodology we follow is described in detail in Section 3. Section 4. is devoted to the experiment and results. Lastly, conclusions and directions for the future are formulated in Sections 5.

2 Related Work

This section describes deep learning methodologies that have achieved state-of-the-art results on challenging NLP problems. It then presents existing works that are related to the Quranic semantic analysis and topic modeling.

2.1 Deep Learning of Word Embedding & Document Embedding

We start by discussing the recent discoveries in feature embedding and the latest approaches for learning the dominant representation of texts. These methods are the inspiration for our work.

Acquiring semantic knowledge and using it in language understanding and processing has always been an active area of study. Researches have resulted in various approaches and techniques related to semantics representation [24]. One well known but simplistic representation is bag of words BOW (or bag of n-gram). However, it lacks the capability to capture the semantics and syntactic order of words in the text. Another common technique is Latent Dirichlet Allocation (LDA) that is usually used for topic modeling. However, it is tough to tune, and results are troublesome to evaluate. Probabilistic topic models [5] such as Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) generate a document embedding space to model and explain word distribution in the corpus where dimensions can be seen as latent semantic structures hidden in the data [4]; [11].

In recent years, machine learning and in particular deep learning with neural networks has played a central role in corpus-based Natural Language Processing NLP [22]. Deep learning related models and methods have recently begun to surpass the previous state-of-the-art results on a variety of tasks and domains such as language modeling, translation, speech recognition, and object recognition in images. One of the most noticeable developments in NLP is the use of machine learning to capture the distributional semantics of words, and in particular deep learning of word embeddings, where words are represented as vectors in a continuous space, capturing many syntactic and semantic relations [21] . Word embeddings and document embedding can be powerful approaches for capturing underlying meanings and relationships within texts, as a step towards presenting the meaningful semantic structure of the text [20]; . Use of deep learning word embeddings has led to substantial improvements in semantic textual similarity and relatedness tasks .

The impressive impact of these models has motivated researchers to consider richer vector representations to larger pieces of texts. In [8] Mikolov and Le released sentence or document vectors transformation. It is another breakthrough on embeddings such that we can use vector to represent a sentence or document. Document embedding maps sentences/documents to informative vectors representation that preserves more semantic and syntactic information. They call it paragraph vectors [8]. Paragraph vectors has been proposed as an unsupervised method for learning distributed representations for pieces of texts. The authors demonstrated the capabilities of their paragraph vectors method on several text

classification and sentiment analysis tasks. [9] also examined paragraph vectors in the context of document similarity tasks.

2.2 Topic Modeling for the Quran

The Quran has been the subject of numerous studies due to its significance. Scholars have studied the Quran for its topics. They have drawn out knowledge and patterns that were the base for many applications to allow search in the holy book. This section provides a review of literature related to text mining and probabilistic topic modeling of the Quran.

Many studies were devoted to text mining and topic modeling with the Quran [3]. Such studies aim at extracting accurate, coherent topics from Quran, which promotes understanding of the text. Latent Dirichlet Allocation (LDA) [5] , as a statistical method, was mainly adopted in most of the works related to Quranic topic modeling [14];[15]; [19] and [13]. However, they were limited to a unigram model, and examined specific chapters and documents of the text [14]. Moreover, most research projects focused on the translation of the Quran in different languages instead of the original text [18].

Latent Dirichlet allocation (LDA) is a generative probabilistic model for a collections of documents (text corpora.) LDA is a topic modeling unsupervised machine learning method that helps discover hidden semantic structures in a text [5]. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words [6]. [11] introduced the semantic representation method that extracts the core of the collection of documents based on the LDA model with the Gibbs sampling algorithm [23]. They demonstrated that the topic model is useful for semantic representation since it can be used in predicting word association and a variety of other linguistic processing and memory tasks.

In [13], the authors presented a method to discover the thematic structure of the Quran using probabilistic topic model. They were able to identify two major topics in the Quran, characterized by the distinct themes discussed in the Makki and the Madani chapters. One limitation to their model was using a unigram language model. However, here, we consider phrases or verses of the Quran as the input for the clustering algorithm and the topic analysis. Another work [12] applied LDA to the Quran. Still, the focus was to compare different term weighting schemes and preprocessing strategies for LDA rather than exploring the thematic structure of the document collection. The Quran was used as the testing corpus, while the model was trained using Bible corpora.

A recent work [14] explored a topic modeling technique to set up a framework for semantic search in the holy Quran. They applied LDA into two structures, Hizb quarters and verses of Joseph chapter. A similar research [15] has used clustering techniques in machine learning to extract topics of the holy Quran. The process was based on the verses of the Quran using nonnegative matrix factorization. However, it was unclear how they linked the keywords of each topic to the associated verses.

[16] have proposed a simple WordNet for the English translation of the second chapter of the Quran. They have created topic-synonym relations between the words in that chapter with different priorities. They have defined different relationships that are used in traditional WordNet. They developed a semantic search algorithm to fetch all verses that contain the query words and their synonyms with high priority. Another work [17] extracted verses from the Quran using web ontology language. They also used the English translation of the Quran. One recent work by [19] proposed topic modeling of the corpus in Indonesian translation of the Quran by generating four main topics that are firmly related to human life. They considered Makki and Madani surahs as the variable for topic modeling categorization. Their results showed Makki's surahs contribute 50% compared to Madani's surahs.

These all together motivated us to further the progress in this field. The primary goal of this work is to exploit a recent trend in machine intelligence, which is the distributed representation of text, to learn an informative representation of the passages of the Quran, potentially allowing for the discovery of knowledge-related connections between its documents. Moreover, we aim at revealing hidden patterns that explain the profound relationship between the verses/passages of the sacred text.

3 Methodology

We use an unsupervised document embedding technique: paragraph vector, to learn vector representation of the verses of the Quran, and potentially revealing significant patterns and inferring coherent topics. The machine learning pipeline is illustrated in figure 1.

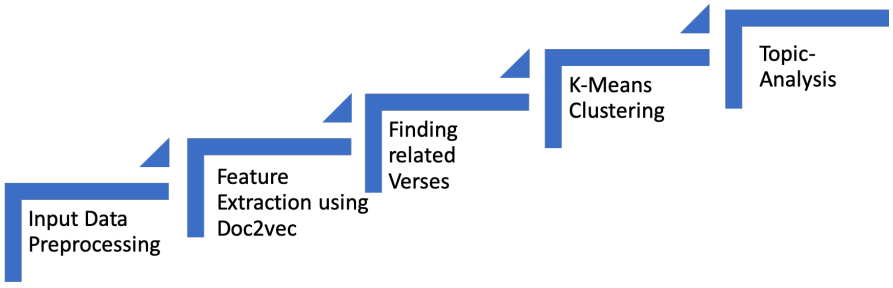

Fig. 1. The ML Pipeline for the clustering and topic analysis

We used paragraph vectors to create fixed-length vector representations for each verse/sentence in the Quran. Paragraph vectors, or doc2vec, were proposed

by Le and Mikolov [8] as a simple extension to word2vec to extend the learning of embeddings from words to word sequences. Doc2vec in Gensim [3], which is a topic modeling python library, is used to generate the paragraph vectors. There are two approaches within doc2vec: a distributed bag of words model and a distributed memory model of paragraph vector. The distributed bag of words model is a simpler model and ignores word order, while the distributed memory model is a more complex model with more parameters. The two techniques are illustrated in figures 2 and 3.

The idea behind the distributed memory model is that word vectors contribute to a prediction task about the next word in the sentence. The model inserts a memory vector to the standard language model, which aims at capturing the topics of the document. The paragraph vector is concatenated or averaged with local context word vectors to predict the next word.

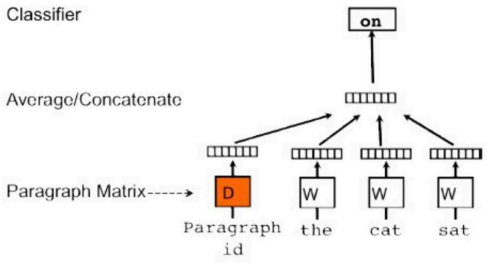

Fig. 2. Paragraph Vector: A distributed memory model(PV-DM) [8]

The paragraph vector can be further simplified when ignoring the context words in the input but forcing the model to predict words randomly sampled from the paragraph in the output. At inference time, the parameters of the classifier and the word vectors are not needed, and back-propagation is used to tune the paragraph vectors. That is the distributed bag of words version of the paragraph vector. The distributed bag of words model works in the same way as skip-gram [8], except that a special token representing the document replaces the input.

From Mikolov et al. experiment [8], PV-DM has proven to be consistently better than PV-DBOW. Thus, in our experiment, we use the distributed memory implementation of the paragraph vector. Besides, we consider the recom-

[3] Doc2vec paragraph embedding was popularised by Gensim - a widely-used implementation of paragraph vectors: https://radimrehurek.com/gensim/

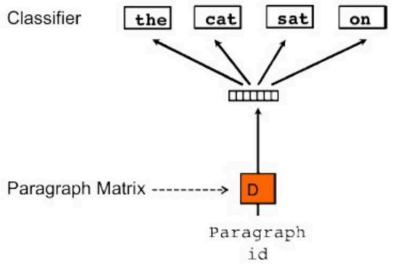

Fig. 3. Paragraph Vector: Distributed Bag Of Words (PV-DBOW) [8]

mendations on optimal doc2vec hyper-parameter settings for general-purpose applications as in [10].

We used Doc2vec implemented in Gensim to learn vector representation of the Quranic verses. We trained the paragraph vectors on the 6,236 verses/passages of the Quran using the original Arabic text from Tanzil project[4]. First, we read the verses from a digitized version of the Quran as a data frame. We preprocess and clean the text using the NLTK library[5]. We removed punctuation, Harakat, and stop-words. Figure 4 shows a snapshot of the the data before it is been processed to be ready for training.

1|6|اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ
1|7|صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ
2|1|بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ الم
2|2|ذَٰلِكَ الْكِتَابُ لَا رَيْبَ ۛ فِيهِ ۛ هُدًى لِلْمُتَّقِينَ
2|3|الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنْفِقُونَ
2|4|وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنْزِلَ إِلَيْكَ وَمَا أُنْزِلَ مِنْ قَبْلِكَ وَبِالْآخِرَةِ هُمْ يُوقِنُونَ
2|5|أُولَٰئِكَ عَلَىٰ هُدًى مِنْ رَبِّهِمْ ۖ وَأُولَٰئِكَ هُمُ الْمُفْلِحُونَ
2|6|إِنَّ الَّذِينَ كَفَرُوا سَوَاءٌ عَلَيْهِمْ أَأَنْذَرْتَهُمْ أَمْ لَمْ تُنْذِرْهُمْ لَا يُؤْمِنُونَ
2|7|خَتَمَ اللَّهُ عَلَىٰ قُلُوبِهِمْ وَعَلَىٰ سَمْعِهِمْ ۖ وَعَلَىٰ أَبْصَارِهِمْ غِشَاوَةٌ ۖ وَلَهُمْ عَذَابٌ عَظِيمٌ
2|8|وَمِنَ النَّاسِ مَنْ يَقُولُ آمَنَّا بِاللَّهِ وَبِالْيَوْمِ الْآخِرِ وَمَا هُمْ بِمُؤْمِنِينَ
2|9|يُخَادِعُونَ اللَّهَ وَالَّذِينَ آمَنُوا وَمَا يَخْدَعُونَ إِلَّا أَنْفُسَهُمْ وَمَا يَشْعُرُونَ
2|10|فِي قُلُوبِهِمْ مَرَضٌ فَزَادَهُمُ اللَّهُ مَرَضًا ۖ وَلَهُمْ عَذَابٌ أَلِيمٌ بِمَا كَانُوا يَكْذِبُونَ

Fig. 4. A snapshot the input data

[4] http://tanzil.net/docs
[5] https://www.nltk.org

Now, the document are ready for training. Next, to produce the verses embeddings, we used the python implementation of doc2vec as part of the Gensim package. We trained the Doc2vec model with different configuration of the hyper-parameters. The data has undergone multiple processes to tune the hyper-parameters, and we drawn on our domain knowledge to poke the model in the right way. Now, we can use the trained model to infer a vector for any verse by passing a list of words to the model.infer_vector function. This vector can then be compared with other vectors via cosine similarity.

4 Experiments and Results

After training doc2vec, document embeddings are generated by the model. The vectors act as features for the Quranic verses. Here, we evaluated the vectors on the task of finding similar verses to examine their effectiveness in capturing the semantics of the verses/passages of the Quran. We inferred the vector for a randomly chosen test document/verse and compared the document to our model. Using intuitive self-evaluation, we were able to locate semantically similar verses and eventually created a dataset of pairs of related verses along with their similarity score. We decide on 50 as the vector size that produced best results in terms of the similarity between the verses in each pair. We used the Qurany ontology browser [6] to verify our results. The Qurany corpus is augmented with an ontology or index of key concepts, taken from a recognized expert source. The corpus allows users to search the Quran corpus for abstract concepts via an ontology browser. It contains a comprehensive hierarchical index or ontology of nearly 1200 concepts in the Quran. Indeed, Doc2vec succeeded in exploring relationships between documents. Examples of our results are illustrated in figure5.

4.1 Clustering with K-Means algorithm

Here, we investigate the structure of the data by grouping the data points (verses of the Quran) into distinct subgroups. With clustering, we try to find Verses that are similar to each other in terms of topics. The objective is to infer patterns in the data that can inform a decision, or sometimes covert the problem to a Supervised Learning problem. The goal of clustering is grouping unlabeled texts in such a way that texts in the same group/cluster are more similar to each other than to those in other clusters. With clustering, we seek to capture in some way the topics or themes in our corpus and the way they are shared between documents (verses) in it. K-Means is considered as one of the most used clustering algorithms due to its simplicity. K-Means puts the observations into k clusters in which each observation belongs to a cluster with the nearest mean. The main idea is to define k centroids, one for each cluster.

We implement our K-Means clustering algorithm in our vectorized documents. We initially determine the number of clusters be 15 (the number of

[6] http://quranytopics.appspot.com [1]

Train Document (955):
«كتاب أنزل إليك فلا يكن في صدرك حرج منه لتنذر به وذكرى للمؤمنين»
[This is] a Book revealed to you, [O Muhammad] - so let there not be in your breast distress therefrom - that you may warn thereby and as a reminder to the believers.

Similar Document (3998, **0.8851784467697144**):
«كتاب أنزلناه إليك مبارك ليدبروا آياته وليتذكر أولو الألباب»
[This is] a blessed Book which We have revealed to you, [O Muhammad], that they might reflect upon its verses and that those of understanding would be reminded.

Train Document (5178):
«هو الذي بعث في الأميين رسولا منهم يتلو عليهم آياته ويزكيهم ويعلمهم الكتاب والحكمة وإن كانوا من قبل لفي ضلال مبين»

It is He who has sent among the unlettered a Messenger from themselves reciting to them His verses and purifying them and teaching them the Book and wisdom – although they were before in clear error –

Similar Document (456, **0.8495432138442993**):
«لقد من الله على المؤمنين إذ بعث فيهم رسولا من أنفسهم يتلو عليهم آياته ويزكيهم ويعلمهم الكتاب والحكمة وإن كانوا من قبل لفي ضلال مبين»

Certainly did Allah confer [great] favor upon the believers when He sent among them a Messenger from themselves, reciting to them His verses and purifying them and teaching them the Book and wisdom, although they had been before in manifest error.

Fig. 5. Pairs of related verses using the paragraph vectors as features of the Quranic verses

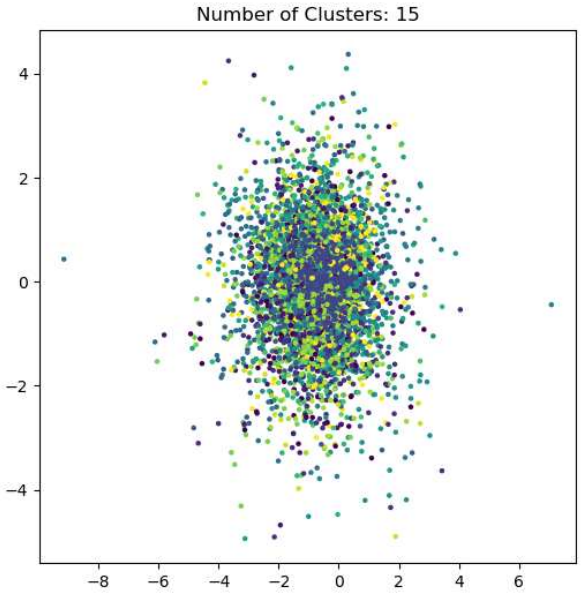

Fig. 6. Visualization of the clustering when number_clusters = 15

main topics from our Qurany corpus), assuming that we have a general sense of the right number of clusters. Another approach would be to do a couple of trial/errors to find the best number of clusters. We tried different values ranging from 2 to 20. We implemented the algorithm in python with the help of SciKit Learn library [7]. We have done the clustering and visualised the clusters, as in figure 6. The list in figure 7 shows examples of the verses in an example cluster.

Chapter	Ayah	Verse
2	30	وإذ قال ربك للملائكة إني جاعل في الأرض خليفة قالوا أتجعل فيها من يفسد فيها ويسفك الدماء ونحن نسبح بحمدك ونقدس لك قال إني أعلم ما لا تعلمون Remember, when your Lord said to the angels: "I have to place a trustee on the earth," they said: "Will You place one there who would create disorder and shed blood, while we intone Your litanies and sanctify Your name?" And God said: "I know what you do not know."
2	33	قال يا آدم أنبئهم بأسمائهم فلما أنبأهم بأسمائهم قال ألم أقل لكم إني أعلم غيب السماوات والأرض وأعلم ما تبدون وما كنتم تكتمون Then He said to Adam: "Convey to them their names." And when he had told them, God said: "Did I not tell you that I know the unknown of the heavens and the earth, and I know what you disclose and know what you hide?"
2	61	وإذ قلتم يا موسى لن نصبر على طعام واحد فادع لنا ربك يخرج لنا مما تنبت الأرض من بقلها وقثائها وفومها وعدسها وبصلها قال أتستبدلون الذي هو أدنى بالذي هو خير اهبطوا مصرا فإن لكم ما سألتم وضربت عليهم الذلة والمسكنة وباءوا بغضب من الله ذلك بأنهم كانوا يكفرون بآيات الله ويقتلون النبيين بغير الحق ذلك بما عصوا وكانوا يعتدون Remember, when you said: "O Moses, we are tired of eating the same food (day after day), ask your Lord to give us fruits of the earth, herbs and cucumbers, grains and lentils and onions;" he said: "Would you rather exchange what is good with what is bad? Go then to the city, you shall have what you ask." So they were disgraced and became indigent, earning the anger of God, for they disbelieved the word of God, and slayed the prophets unjustly, for they transgressed and rebelled.

Fig. 7. A List of Verses located in cluster 1

Figure 8 shows a snapshot of clusters along with individual verses contained in each cluster.

To evaluate our clustering, we used a tagged corpus: Qurany. We compared our results against the corpus to verify the relationships between the verses of the Quran, identify how they are related, and address the concepts covered in each cluster. Our findings confirmed that paragraph vectors representations offered a useful input representation that promoted the clustering performance. Moreover, we use two different metrics to identify how functional is this clus-

[7] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Cluster: 0
495

	Cluster	Verse
0	0	...وإذا لقوا الذين آمنوا قالوا آمنا وإذا خلوا إلى
1	0	... والذين كفروا وكذبوا بآياتنا أولئك أصحاب النار
2	0	...وآمنوا بما أنزلت مصدقا لما معكم ولا تكونوا أول
3	0	...وظللنا عليكم الغمام وأنزلنا عليكم المن والسلوى
4	0	...ولقد علمتم الذين اعتدوا منكم في السبت فقلنا له

Cluster: 1
608

	Cluster	Verse
0	1	...وإذ قال ربك للملائكة إني جاعل في الأرض خليفة ق
1	1	...وعلم آدم الأسماء كلها ثم عرضهم على الملائكة فق
2	1	... قالوا سبحانك لا علم لنا إلا ما علمتنا إنك أنت
3	1	...وإذ فرقنا بكم البحر فأنجيناكم وأغرقنا آل فرعون
4	1	...وإذ قال موسى لقومه إن الله يأمركم أن تذبحوا بق

Cluster: 2
358

	Cluster	Verse
0	2	...أو كصيب من السماء فيه ظلمات ورعد وبرق يجعلون أ
1	2	...الذي جعل لكم الأرض فراشا والسماء بناء وأنزل من
2	2	...هو الذي خلق لكم ما في الأرض جميعا ثم استوى إلى
3	2	...وإذ استسقى موسى لقومه فقلنا اضرب بعصاك الحجر ف
4	2	...ثم قست قلوبكم من بعد ذلك فهي كالحجارة أو أشد ق

Cluster: 3
298

	Cluster	Verse
0	3	...قال يا آدم أنبئهم بأسمائهم فلما أنبأهم بأسمائه
1	3	...وإذ قلنا للملائكة اسجدوا لآدم فسجدوا إلا إبليس
2	3	...وقلنا يا آدم اسكن أنت وزوجك الجنة وكلا منها رغ

Fig. 8. The derived clusters

tering and to measure its performance [8]. First, we consider the inertia metric, which is the within-cluster sum of squares of distances to the cluster center. The algorithm aims to choose centroids that minimize the inertia, which can indicate how internally coherent clusters are. The other metric is Silhouette Score which can be used to determine the degree of separation between clusters. Silhouette Score can take values in the interval [-1, 1]:

- If it is 0 , then the sample is very close to the neighboring clusters.
- It it is 1, then the sample is far away from the neighboring clusters.
- It it is -1, then the sample is assigned to the wrong clusters.

As the coefficient approaches 1, it indicates having good clustering. After we calculated the inertia and silhouette scores, we plotted them and evaluated the performance of the clustering algorithm. Figure 9 shows the result of the two metrics. The inertia score always drops when we increase the number of clusters. From the silhouette curve, As the plots in the figure show, n_clusters=14 has the best average silhouette score and all clusters being above the average shows that it is actually a good choice. The value is close to the number of main topics in the Qurany corpus (15), which indicates we got encouraging results. Joining the elbow curve with the silhouette score curve provides valuable insight into the performance of K-Means.

[8] https://scikit-learn.org/stable/modules/clustering.html

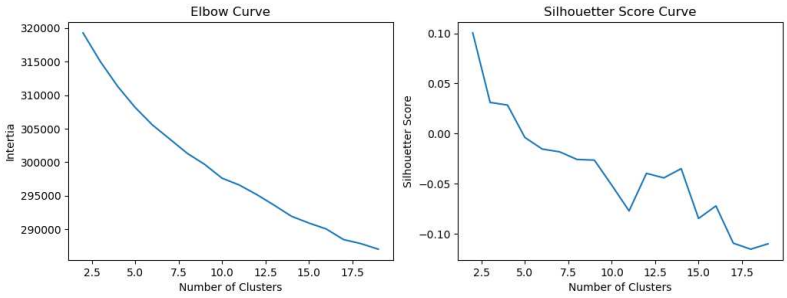

Fig. 9. Evaluation the performance of clustering using Elbow and Silhouette score curve

5 Conclusions & Future Directions

This work presented a new vector representation of the Quranic verses at the paragraph level. These vectors can be used as features and leveraged for the clustering and topic analysis. We then examined the capabilities of paragraph vectors on finding related verses/passages. We were able to locate semantically related verses, and created a dataset of pairs of related verses. We used the Qurany ontology browser to verify our results. Qurany corpus is augmented with an ontology, taken from a recognized expert source, and authenticated by experts with domain knowledge. Next, we fed the features to the clustering algorithm K-Means. The derived clusters suggested groups of related verses that share a common central concept.

In the future, we plan to evaluate the derived clusters against a tagged corpus automatically. We will figure out a classifier that best fits our data, and adequately captures the relations between the data points (verses of the Quran). Eventually, we add more significance to our conclusion and benchmark the derived features of the original Quranic verses.

6 Acknowledgements

Menwa Alshammeri is supported by a PhD scholarship from the Ministry of Higher Education, Saudi Arabia. The author is grateful for the support from Jouf University for sponsoring her research.

References

1. Abbas, Noorhan Hassan. "Quran'search for a concept'tool and website." PhD diss., University of Leeds (School of Computing), 2009.
2. Soliman, Abu Bakr, Kareem Eissa, and Samhaa R. El-Beltagy. "Aravec: A set of arabic word embedding models for use in arabic nlp." Procedia Computer Science 117 (2017): 256-265.
3. Alrehaili, Sameer M., and Eric Atwell. "Computational ontologies for semantic tagging of the Quran: A survey of past approaches." In LREC 2014 Proceedings. European Language Resources Association, 2014.
4. Lu, Yue, Qiaozhu Mei, and ChengXiang Zhai. "Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA." Information Retrieval 14, no. 2 (2011): 178-203.
5. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3, no. Jan (2003): 993-1022.
6. Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis." Discourse processes 25, no. 2-3 (1998): 259-284.
7. Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.
8. Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." In International conference on machine learning, pp. 1188-1196. 2014.
9. Dai, Andrew M., Christopher Olah, and Quoc V. Le. "Document embedding with paragraph vectors." arXiv preprint arXiv:1507.07998 (2015).
10. Lau, Jey Han, and Timothy Baldwin. "An empirical evaluation of doc2vec with practical insights into document embedding generation." arXiv preprint arXiv:1607.05368 (2016).
11. Griffiths, Thomas L., Mark Steyvers, and Joshua B. Tenenbaum. "Topics in semantic representation." Psychological review 114, no. 2 (2007): 211.
12. Wilson, Andrew T., and Peter A. Chew. "Term weighting schemes for latent dirichlet allocation." In human language technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics, pp. 465-473. Association for Computational Linguistics, 2010.
13. Siddiqui, Muazzam Ahmed, Syed Muhammad Faraz, and Sohail Abdul Sattar. "Discovering the thematic structure of the Quran using probabilistic topic model." In 2013 Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, pp. 234-239. IEEE, 2013.
14. Alhawarat, Mohammad. "Extracting topics from the holy Quran using generative models." International Journal of Advanced Computer Science and Applications 6, no. 12 (2015): 288-294.
15. Panju, Maysum H. "Statistical extraction and visualization of topics in the quran corpus." Student. Math. Uwaterloo. Ca (2014).
16. Shoaib, Muhammad, M. Nadeem Yasin, Ullah K. Hikmat, M. Imran Saeed, and Malik Sikandar H. Khiyal. "Relational WordNet model for semantic search in Holy Quran." In 2009 International Conference on Emerging Technologies, pp. 29-34. IEEE, 2009.
17. Yauri, Aliyu Rufai, R. Abdul Kadir, Azreen Azman, and MA Azmi Murad. "Quranic verse extraction base on concepts using OWL-DL ontology." Research Journal of Applied Sciences, Engineering and Technology 6, no. 23 (2013): 4492-4498.

18. Putra, Syopiansyah Jaya, Teddy Mantoro, and Muhamad Nur Gunawan. "Text mining for Indonesian translation of the Quran: A systematic review." In 2017 International Conference on Computing, Engineering, and Design (ICCED), pp. 1-5. IEEE, 2017.
19. Rolliawati, Dwi, Indri Sudanawati Rozas, and Muhamad Ratodi. "Text Mining Approach for Topic Modeling of Corpus Al Qur'an in Indonesian Translation." , 2018.
20. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. MIT press.
21. Sahlgren, M. (2008). The distributional hypothesis. Italian Journal of Disability Studies, 20, 33-53.
22. Goldberg, Y., 2017. Neural network methods for natural language processing. Synthesis Lectures on Human Language Technologies, 10(1), pp. 1–309.
23. Carlo, Chain Monte. "Markov chain monte carlo and gibbs sampling." Lecture notes for EEB 581 (2004).
24. Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C., 2003. A neural probabilistic language model. Journal of machine learning research, Feb, Volume 3, pp. 1137-1155.