# Exploration and Visualization of topics in the Quranic Text using Topic Modeling

Project Report
CS 410 - Fall 2017

Muhammad Ahmed
[mzahmed2@illinois.edu]

Kahtan Al Jewary
[ksa2@illinois.edu]

## Background

Quranic text (English translation or Arabic text) has rich vocabulary, topics and word association. Semantic search in the holy Quran can be supported by finding accurate, coherent topics which help in finding contextual terms related to the user search terms. While one verse may contain multiple topics, another set of verses may comprise one topic. Also, one topic may be repeated in several contexts and in more than one chapter.

The Quran is widely read, recited and searched by Muslims throughout the world. Muslims usually need to search and retrieve relevant information based on more than just simple keywords. This is where discovering Quranic topics would improve the search and retrieval experience. The result of the project provides topics comprised of sets of words. A concatenated subset of any set of words can be used as a search string to discover details about topics and stories spread throughout the Holy Quran and are not contained in only a single chapter.

## Corpora

The Holy Quran is divided into 30 Chapters (Paras) and each chapter is subdivided into sections (Rukus) and each section is further divided into Ayats (Verses). There are a total of 114 chapters (Suras) in Holy Quran. The Sura is identified as Makki and Madani Sura. The Suras and sections are sometimes overlapped. Sura and sections can be continued (run) over chapters, but Suras are complete units. Suras are subdivided into Ayats. The Suras and Ayats and their placements in the Quran are ordained by God. From these 114 Suras of Quran, 89 are Makki Suras and 25 are Madani Suras. Similarly there are 6236 Ayats of which 4725 Ayats are Makki and 1511 are Madani Ayats. The mean number of Ayats per Sura is calculated to be 51.72, whereas Makki Suras have a calculated mean of 41.87 and Madani Suras have the mean 82.56 showing that Madani Suras are larger than Makki Suras with respect to lengths of Ayats. The range of Madani Sura, Ayats size is also larger than Makki Suras which mean that the variation in Madani Suras are more than the Makki Suras. This fact has also been noted by many Mufassirs. The total number of letters in Holy Quran is 322,564 and total words are 86,872 showing the mean number of letters per word is 3.71. In Madani Suras the total number of letters is 124755 and the total number of words is 33,247 showing the mean number of letters per word is 3.75. In Makki Suras, the total number of letters is 197,809 and the number of words is 53,625 showing the mean is 3.69.

The available Quranic text data didn't reflect the actual subdivisions. The data was only divided into 114 chapters. The lengths of the chapters are highly variant resulting in the need to merge all chapters into a single body of text and then dividing it into equal chunks of text. After a process of trial and error, it was discovered that dividing the corpus into 100 "chunks" provided satisfactory results.

The fact that the style of addressing in the Quran are different in Makkah and Madinah gives us a prior knowledge about what sort of topics are covered and may affect model selection and estimation. The Makki Surahs generally consist of subjects pertaining to oneness of God, clarifying this misconception in beliefs, Prophethood, Day of Judgement and words of comfort for the Prophet by narrating incidents from the past. Madani Surahs generally consist of family and social laws, and exposition of limits and duties.

## Dataset

The Quran is contained in one book, therefore, the whole dataset is just one XML file. The XML file contains the Quran book as the root and the chapters as the children. Each child Chapter has Verse children which each contains the text data for that verse. Python ElementTree XML API was used to parse the data. The following is a visual representation of how the data is structured within the XML file:

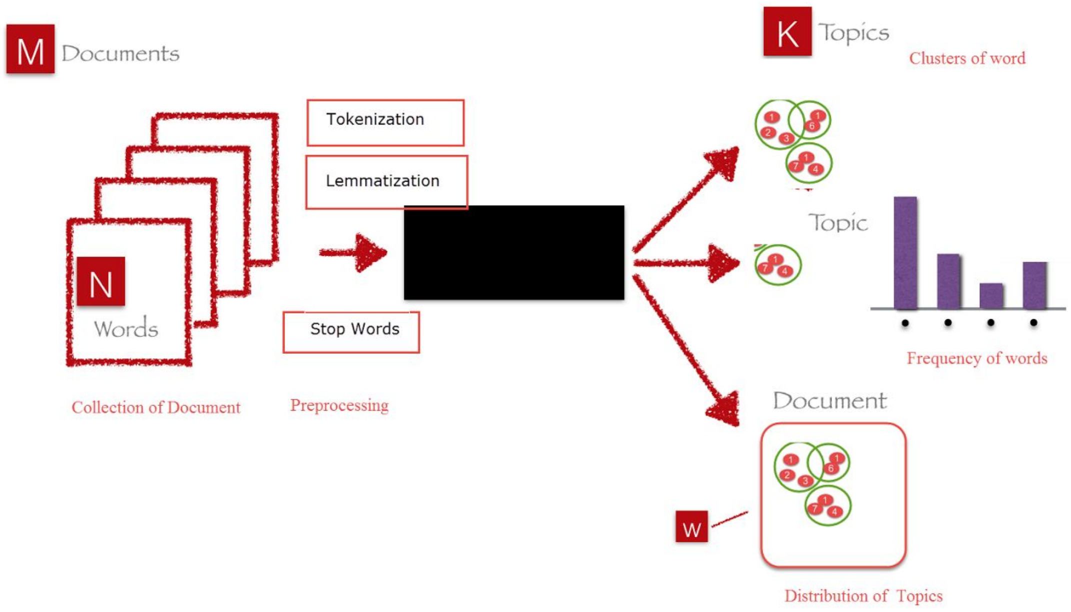
Structure of the Dataset

```
<Dataset>
.     <Chapter 1>
.     .     <Verse 1>
.     .           <Data>
.     .     <Verse 1>
.     .
.     .       . . .
.     .
.     .     <Verse m>
.     .           <Data>
.     .     <Verse m>
.     <Chapter 1>
.
.       . . .
.
.     <Chapter n>
.     .     <Verse 1>
.     .           <Data>
.     .     <Verse 1>
.     .
.     .       . . .
.     .
.     .     <Verse m>
.     .           <Data>
.     .     <Verse m>
.     <Chapter n>
<Dataset>
```

After parsing the XML file , each chapter became a single unit of text which is the concatenation of the text of all its children verses. The end result of the parsing process is 114 chapters. Since the number of verses in each chapter and the length of each verse are not fixed,  the length of each resultant chapter was highly variant.

It was necessary to concatenate all the chapters into a single text unit and then divide it into nearly equal sized documents.

## Conceptual Design



The conceptual design diagram provides a high-level overview of the project components:

1. The Quranic text data is divided into M documents (M represents the number of text chunks within the context of the project).
2. Since all the chapters were merged and then subdivided into equal size text chunks, each document has nearly equal number of words.
3. Once M documents are created, each document is then preprocessed through a pipeline that does the following operations:
    a. Tokenization
    b. Lower casing
    c. PoS filtering
    d. Lemmatization
    e. Stop word removal
    f. Punctuation removal
    g. Special character removal

4. Once all the documents are preprocessed, LDA model is applied using a user provided k number of topics and n number of iterations. It was noticed that the end results are sensitive to n.
5. After obtaining the LDA model results, LDAvis is used to visualize the results.

# Model

Topic modeling is a very popular approach for representing the content of documents. A document is assumed to draw its vocabulary from one or more topics. Topics are represented as probability distributions over the vocabulary, where differing topics give different words high probabilities. We infer a set of topics which can be used to describe the contents of a collection. The high probability topics and words within them can be viewed as a loose description of the collection, with better topic models providing better descriptions. Different topic models specify different document generative procedures, which can lead to very different topics

We use PLSA and LDA as examples to describe the generative process. In the project implementation, LDA was used. Gensim implementation of LDA was used to apply the model to the dataset.

## PLSA

The document collection is represented as term-by-document matrix T(d,w) where $d \in D = \{d_1, ..., d_M\}$ and $w \in W = \{w_1, ...,w_N\}$ Given k number of topics $Z = z_1, ..., z_k$ , the aim is to find the probability distribution of words in a topic and the probability of a document given a topic.

The generative process is as follows:

For each document $d \in D$ with probability P(d),
select a latent topic z with probability P(z|d),
generate a word w with probability P(w|z).

$$
\begin{aligned}
P(d,w) &= X_{z \in Z}\, P(z)P(w|z)P(d|z) \\
&= P(d)\, X_{z \in Z}\, P(w|z)P(z|d).
\end{aligned}
$$

PLSA can not be used to compute probability of new document. It is more suitable in fitting the training documents.

## LDA

LDA is a Bayesian version of PLSA and imposes Dirichlet prior on the model parameters. Given the parameters $\alpha$ and $\beta$, the joint distribution of a topic mixture $\theta$, a set of N topics z, and a set of N words w is given by:

$$p(\theta, z,w|\alpha,\beta) = p(\theta|\alpha) \; \Pi^{N}_{n=1} p(z_n \,|\theta)p(w_n \,|z_n,\beta)$$

Generative process:

1. Choose N
2. Choose $\theta \sim \mathrm{Dir}(\alpha)$
3. For each of the N words $w_n$
    a. Choose a topic $z_n \sim \mathrm{Multinomial}(\theta)$
    b. Choose a word $w_n$ from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic $z_n$

Where: N is the number of words in a document.
$z_n$ the $n_{th}$ topic for the word wn.
$\theta$ the topic distribution for a document.
$\alpha$ the parameter of the Dirichlet prior on the per-document topic distributions
$\beta$ is the parameter of the Dirichlet prior on the per-topic word distribution.

# Algorithm

Initialize Paramters

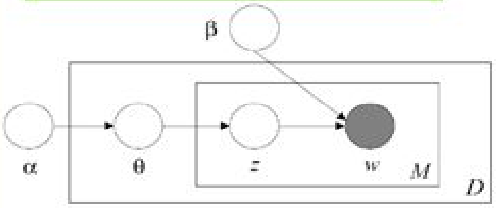


Plate Diagram

1. Set number of topic K
2. Assign word w a random topic 1 2 3 4 w
3. Set number of Iteration
4. Set alpha and beta value

Iterate

For each word in each document:
Resample topic for word, given all other words and their current topic assignments

Given k number of topics Z = z1, ..., zk , the aim is to find:
the probability distribution of words in a topic,
the probability of a document given a topic.

Repeat

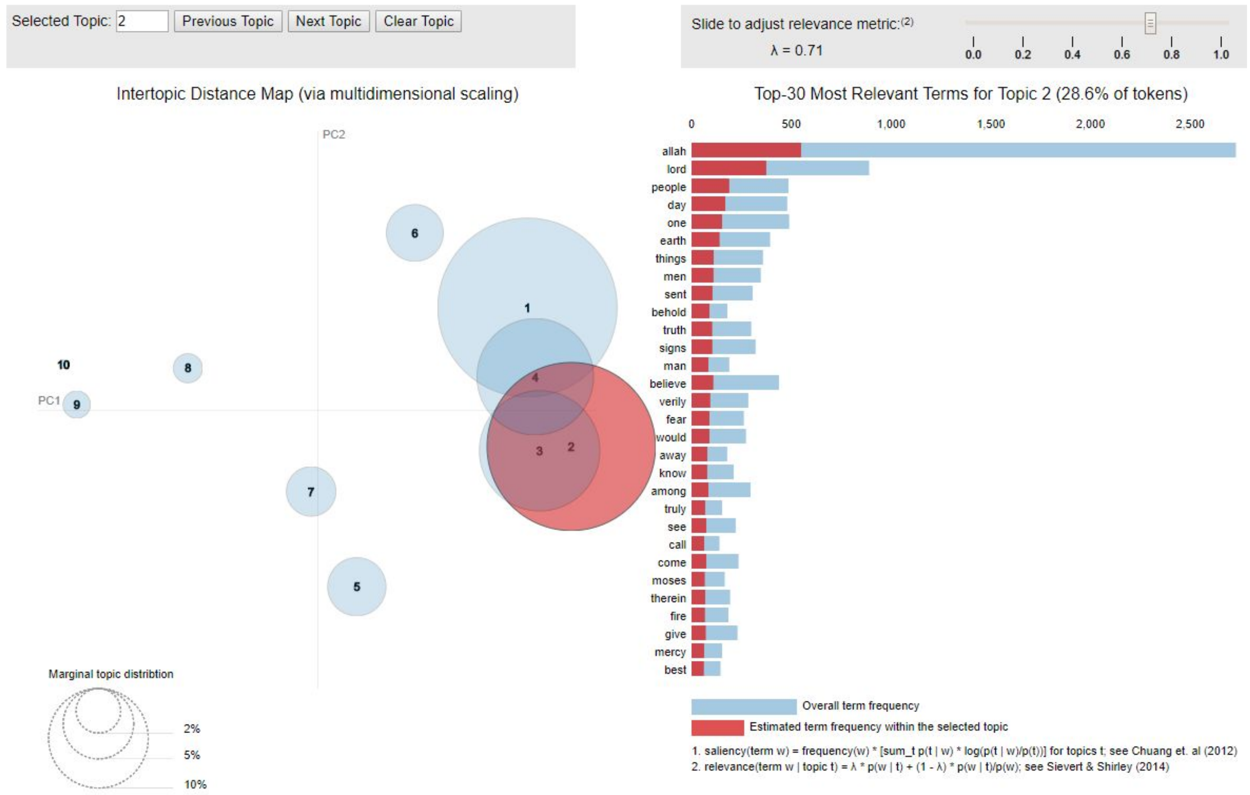
Evaluate

1. Word Intrusion - Choose 5 to 10 high probability words from a random topic, substitute one of the word with intruder (a word with low probability). Present to human and see if human can reliably tell which one is intruder. Assumption Intruder word can easily be identified in coherent topics while in less coherent topics humans will pick a random one.

2. Topic Intrusion - Choose 5 to 10 high probability topics from a random document, substitute one of the topic with intruder (a topic with low probability). Present to human and see if human can reliably tell which one is intruder. Assumption Intruder topic can easily be identified as irrelevant topic in the document

# Output - LDAvis

LDAvis is designed to help users interpret the topics in a topic model that has been fit to a corpus of text data. The package extracts information from a fitted LDA topic model to inform an interactive web-based visualization. The following figure is a sample Quran topic output:



Selected Topic: 2

Slide to adjust relevance metric:(2)

λ = 0.71

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

LDAvis requires five input arguments:

1. φ, the K × W matrix containing the estimated probability mass function over the W terms in the vocabulary for each of the K topics in the model. Note that φkw > 0 for all k ∈ 1...K and all w ∈ 1...W , because of the priors. (Although the software allows values of zero due to rounding). Each of the K rows of φ must sum to one.

2. $\theta$, the $D \times K$ matrix containing the estimated probability mass function over the K topics in the model for each of the D documents in the corpus. Note that $\theta dk > 0$ for all $d \in 1...D$ and all $k \in 1...K$, because of the priors (although, as above, the software accepts zeroes due to rounding). Each of the D rows of $\theta$ must sum to one
3. nd , the number of tokens observed in document d, where nd is required to be an integer greater than zero, for documents d = 1...D. Denoted doc.length in our code.
4. vocab, the length-W character vector containing the terms in the vocabulary (listed in the same order as the columns of $\varphi$).
5. Mw , the frequency of term w across the entire corpus, where Mw is required to be an integer greater than zero for each term w = 1...W . Denoted term.frequency in our code.

## **Definitions of Visual Elements in LDAvis**

There are essentially four sets of visual elements that can be displayed, depending on the state of the visualization. They are:

1. Default Topic Circles: K circles, one to represent each topic, whose areas are set to be proportional to the proportions of the topics across the N total tokens in the corpus. The default topic circles are displayed when no term is highlighted.
2. Red Bars: K × W red horizontal bars, each of which represents the estimated number of times a given term was generated by a given topic. When a topic is selected, we show the red bars for the R most relevant terms for the selected topic, where R = 30 by default
3. Blue Bars: W blue horizontal bars, one to represent the overall frequency of each term in the corpus. When no topic is selected, we display the blue bars for the R most salient terms in the corpus, and when a topic is selected, we display the blue bars for the R most relevant terms.

4. Topic-Term Circles: K × W circles whose areas are set to be proportional to the frequencies with which a given term is estimated to have been generated by the topics. When a given term, w, is highlighted, the K default circles transition (i.e. their areas change) to the K topic-term circles for term w.

In simpler terms:

- The size of a bubble reflects the number of words in a topic.
- The distance between bubbles reflects how different or similar two topics are.
- Hovering over a word highlights other topics the word is a member of.

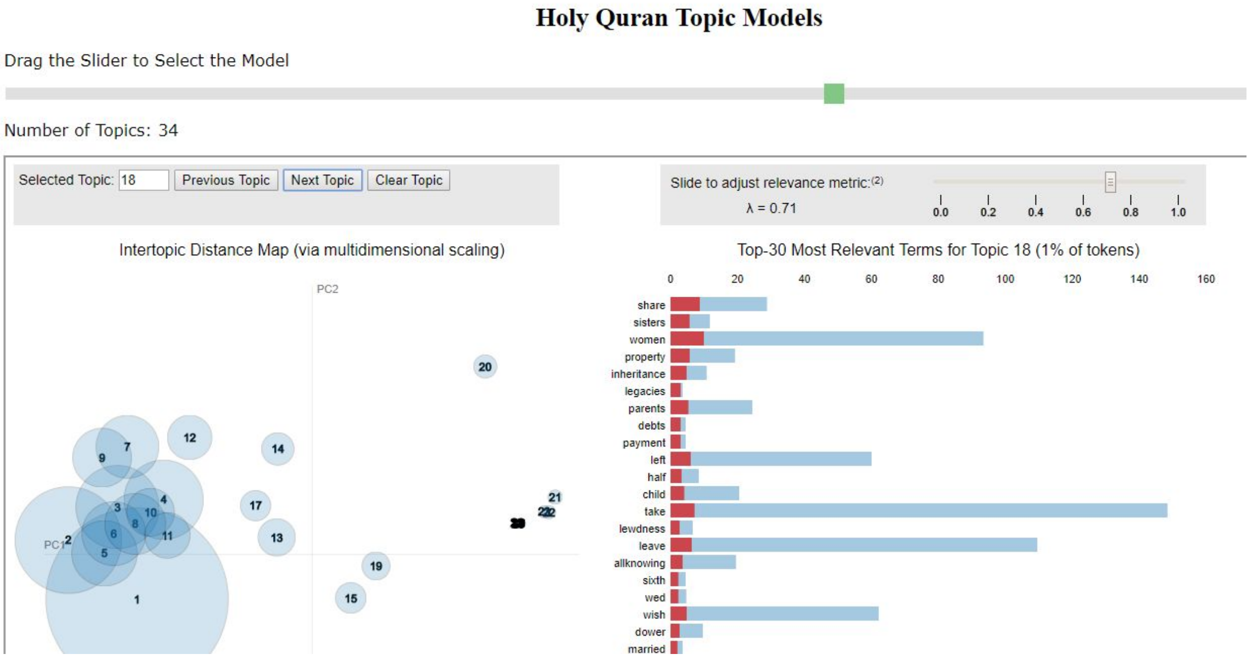
## Implementation

Python libraries were primarily used to implement the project components. The project implementation code and its documentation can be found here.

## Results

The main result of the project is the LDAvis visualization of the Quranic topics. The final form of the visualization is a result of a cyclic process of fitting the model and refining the document cleaning process. The visualization provided clues to which terms occur in the documents with very high counts that removing them could possibly improve the final results. Since the Quran is a translation of Arabic, applying the English stop word filtering and POS tagging didn’t completely eliminate all the stop words. This is where the cyclic process of refining the results became imperative. Over 20 additional words were identified to follow the same statistical pattern of the English stop words. Punctuation also posed a challenge. Some of the translated Arabic words included some special characters to indicate some language specific phonetic guides. Additional code had to be added to handle their occurrences.

In order to determine an optimal number of topics that would result in a good representation of the topics in the Quran, 49 models were applied to the data with k ranging from 2 to 50. All the models were then visualized using LDAviz. To make navigating these models easy, an iframe slider page was created to allow exploring all the models through one window. The following is a figure displaying the iframe slider:



Holy Quran Topic Models

Drag the Slider to Select the Model

Number of Topics: 34

Selected Topic: 18

Slide to adjust relevance metric:(2)

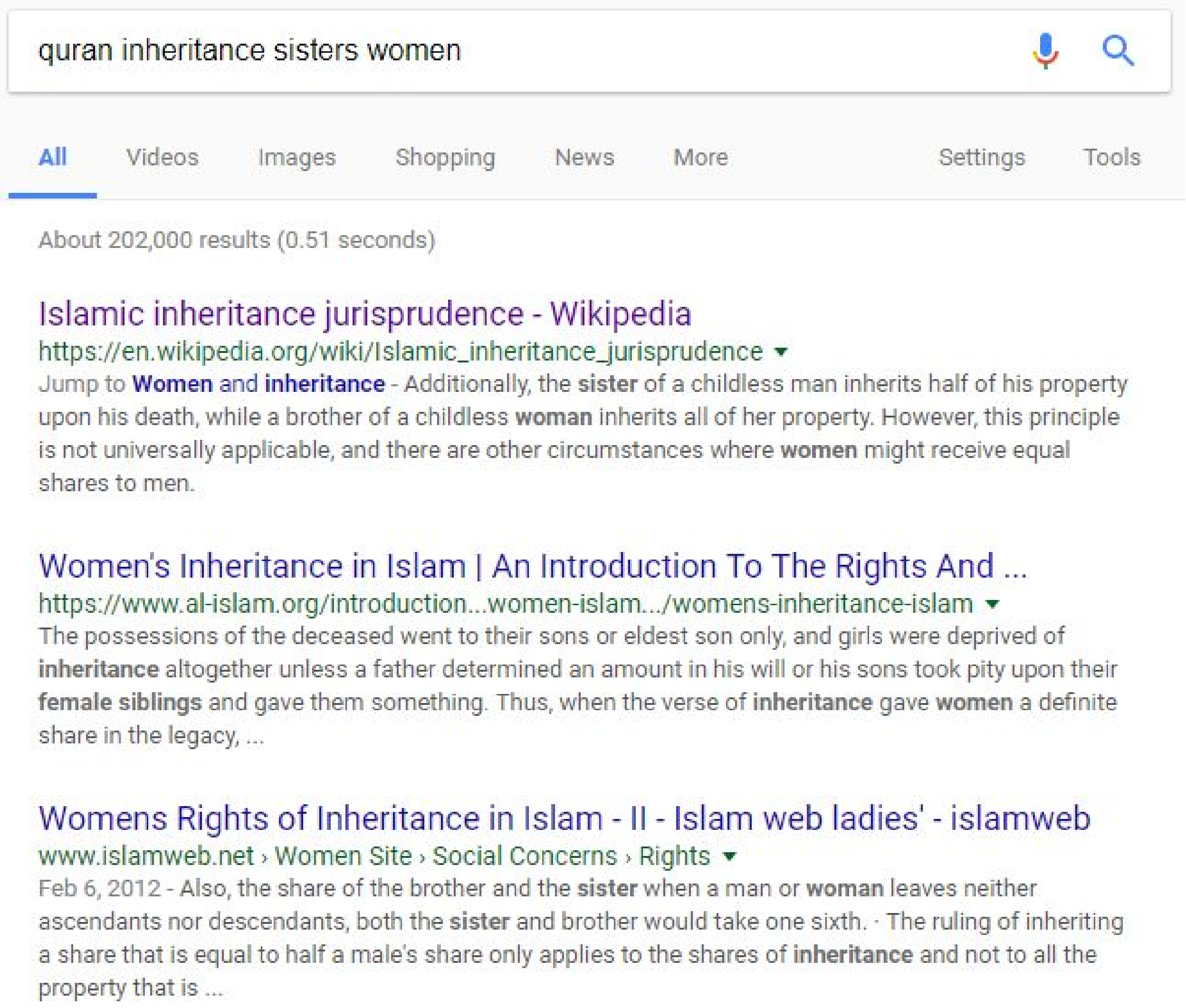
λ = 0.71

After going through the different models and tuning λ, a model with 37 topics and λ value of 0.71 provided satisfying results. Although 37 topics were determined to provide some satisfying results. Most of the other models also provided good results. We kept the 40 models with the iframe slider view in the project repository for other users to explore since deciding which model provides better results is a subjective matter.

An interesting application of the results could be to use the resultant topic words to formulate web search strings. Some words from random topics were used to create search strings. Then, web searches using these strings, prefixed with the word “quran”, were performed on Google. The search results were very interesting. Here are few examples:

**Search String:** quran inheritance sisters women

**Sample Results:**



quran inheritance sisters women

About 202,000 results (0.51 seconds)

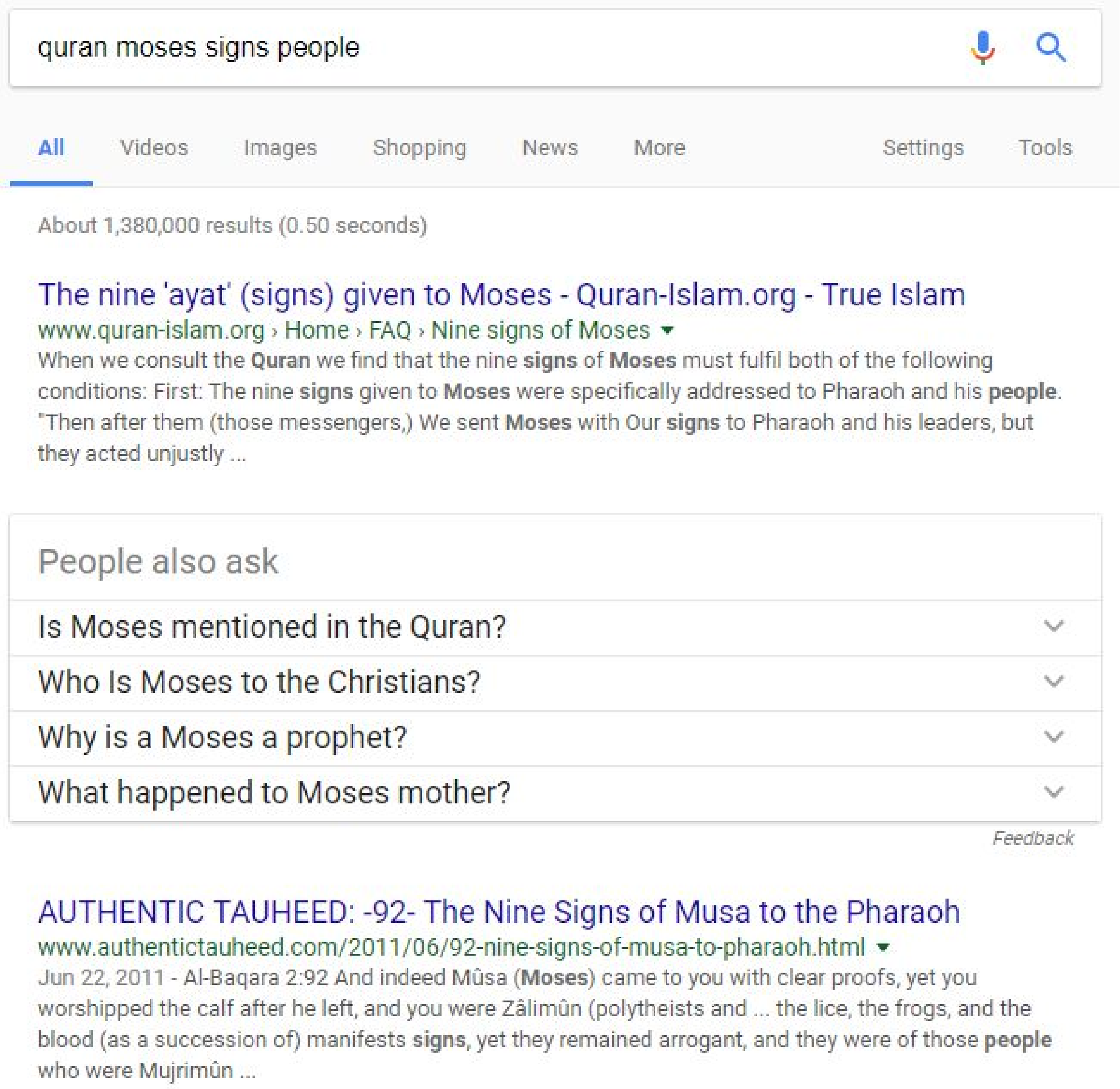
Islamic inheritance jurisprudence - Wikipedia
https://en.wikipedia.org/wiki/Islamic_inheritance_jurisprudence ▾
Jump to **Women** and **inheritance** - Additionally, the **sister** of a childless man inherits half of his property upon his death, while a brother of a childless **woman** inherits all of her property. However, this principle is not universally applicable, and there are other circumstances where **women** might receive equal shares to men.

Women's Inheritance in Islam | An Introduction To The Rights And ...
https://www.al-islam.org/introduction...women-islam.../womens-inheritance-islam ▾
The possessions of the deceased went to their sons or eldest son only, and girls were deprived of **inheritance** altogether unless a father determined an amount in his will or his sons took pity upon their **female siblings** and gave them something. Thus, when the verse of **inheritance** gave **women** a definite share in the legacy, ...

Womens Rights of Inheritance in Islam - II - Islam web ladies' - islamweb
www.islamweb.net › Women Site › Social Concerns › Rights ▾
Feb 6, 2012 - Also, the share of the brother and the **sister** when a man or **woman** leaves neither ascendants nor descendants, both the **sister** and brother would take one sixth. · The ruling of inheriting a share that is equal to half a male's share only applies to the shares of **inheritance** and not to all the property that is ...

**Search String:** quran inheritance sisters women

**Sample Results:**



quran moses signs people

About 1,380,000 results (0.50 seconds)

The nine 'ayat' (signs) given to Moses - Quran-Islam.org - True Islam
www.quran-islam.org › Home › FAQ › Nine signs of Moses
When we consult the **Quran** we find that the nine **signs** of **Moses** must fulfil both of the following conditions: First: The nine **signs** given to **Moses** were specifically addressed to Pharaoh and his **people**. "Then after them (those messengers,) We sent **Moses** with Our **signs** to Pharaoh and his leaders, but they acted unjustly ...

People also ask

Is Moses mentioned in the Quran?

Who Is Moses to the Christians?

Why is a Moses a prophet?

What happened to Moses mother?

AUTHENTIC TAUHEED: -92- The Nine Signs of Musa to the Pharaoh
www.authentictauheed.com/2011/06/92-nine-signs-of-musa-to-pharaoh.html
Jun 22, 2011 - Al-Baqara 2:92 And indeed Mûsa (**Moses**) came to you with clear proofs, yet you worshipped the calf after he left, and you were Zâlimûn (polytheists and ... the lice, the frogs, and the blood (as a succession of) manifests **signs**, yet they remained arrogant, and they were of those **people** who were Mujrimûn ...

As the search results show, these topics are valid and other people have already performed web search on such topics. This shows that the project has achieved the desired and intended results. We hope that other people would greatly benefit from the results of the project.

## References

- http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
- http://semanticsearchart.com/researchLSA.html
- http://semanticsearchart.com/researchpLSA.html
- http://jmlr.org/papers/volume3/blei03a/blei03a.pdf
- https://github.com/bmabey/pyLDAvis
- http://cran.r-project.org/web/packages/LDAvis/vignettes/details.pdf
- https://github.com/cpsievert/LDAvis