The Application of the Topic Modeling to Question Answer Retrieval

Jelica Vasiljevic*, Tom Lampert**, Milos Ivanovic*
* Faculty of Science, University of Kragujevac, Serbia
** Laboratoire Quantup, Strasbourg, France
jeca.kg@hotmail.com
mivanovic@kg.ac.rs
t.lampert@laboquantup.eu

Abstract*—Topic modeling (TM) is used for the extraction of the information from unstructured documents. The aim of this study is to investigate the application of the Latent Dirichlet Allocation Topic modeling algorithm to question answer retrieval. The most appropriate answer is automatically selected from a database of answers based on a combination of several similarity measures. The primary hypothesis assumed in this study is that a question and its correct answer are thematically similar. All TM results were compared to a simple word count approach, employed as the reference model. Results show that the topic modeling approach performs better than the reference model as the number of the documents increase. It is also proved that the difference in results is statistically significant. Nevertheless, basic LDA turned out to be insufficient for efficient question answering. It is therefore hypothesized that additional expert knowledge would greatly improve its performance.

I. Introduction

Community Question Answering [1] sites, such as StackExchange and Yahoo! Answers, have become very important sources of information on the internet. They enable users to exchange knowledge by posing questions and offering answers, and as these exchanges are stored, such sites have created vast stores of valuable knowledge. A significant portion of these databases can be used to answer new question if they are related to the information that already exist in the database.

Question Answer retrieval refers to the selection of one or more correct answers to a given question. An answer is selected from a set of potential answers that already exist. Typically, a question is considered to be lexically similar to the correct answer. Thus, the question is processed and important lexical characteristics are extracted which are then used in order to retrieve the correct answer. Commonly used preprocessing methods in the lexical processing chain are: lowercase transformation, stop word removal, stemming, lemmatization, and so on. However, it is possible that a question and its correct answer do not have any words in common. Thus, lexical similarity may not always be sufficient to select the correct answer to a given question. Because of this, it is important to encode a question's semantics when inferring the correct answer.

Using Latent Dirichlet Allocation (LDA) topic modeling [1] it is possible to encode the semantics of a given document. In [2], the authors propose a number of LDA based similarity measures which could be applied to the question answering problem. Furthermore, a novel statistical topic model for the question answering problem in community archives is proposed in [3].

In this paper, we explore the limits of the classic LDA topic modeling algorithm. We combine several similarity measures in order to rank answers according to their semantic similarity. We also test the influence of synonyms, stemming, and lemmatization on the final result.

This remainder of this manuscript is organized as follows: Section II presents the methods used within the presented study, Section III the experimental setup and results, and finally our conclusions are drawn in Section IV.

II. Method

A. Topic Modeling Approach

Latent Dirichlet Allocation is described in [1]. It is an unsupervised, statistical approach to document modeling that discovers latent semantic topics in a large collection of text documents [1]. Each document is represented as a distribution over a fixed number of topics, while each topic is represented as a distribution over words. In the presented work, these two probability distributions are used to calculate the similarity between a question and all possible answers. The answer which is the most similar to the question is then proposed to be the correct answer.

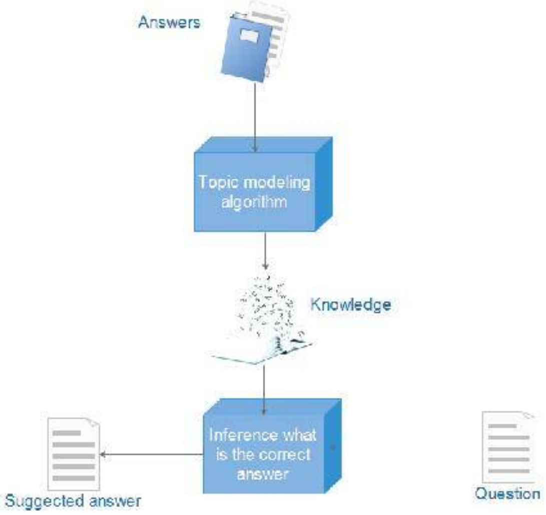

Figure 1. Conceptual overview of the LDA Topic Modeling approach.

Figure 1 presents a conceptual overview of the proposed solution. An LDA topic model is constructed using a training set composed of all the answers that currently exist in a database. Therefore, topic distributions over all possible answers are known, and word distributions over all topics are known. This model can then be used to infer the distribution over topics and distributions over words of a new, user defined question, and the most similar answer that exists in the database can be selected.

B. Similarity Measures

The output of LDA is a multinomial distribution over topics for each document in the (answer) database and a multinomial distribution over words for each topic. The nature of the algorithm also allows for the inference of topic distributions for unseen documents. In the presented experimental setup, questions are the unseen documents for which a topic distribution is determined using LDA, according to the topic model learnt using the database of answers. A similarity measure is therefore required to infer the answer with the most similar topic distribution to the question, and in this work several are evaluated.

B.1 Cosine Similarity

The cosine similarity measures the cosine of the angle between two vectors. As the measure tends towards 1, two vectors are more similar. The cosine similarity between two vectors a and b is measured as follows

$$\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|a\| \cdot \|b\|}. \tag{1}$$

In topic modeling, the distribution over topics is discrete, so it can be represented as a vector. The first coordinate of that vector is the probability of the first topic in the document, the second coordinate is probability of the second topic in the same document and so on. Since the distribution over topics for each document D is known, it is possible to measure the similarity of two documents using (1).

Given a question, the answer that results in a cosine similarity closest to 1 is selected as the correct answer.

B.2 Similarity Measure Based on Query Likelihood Probability

It has already been stated that the distribution over topics is known for each document. Furthermore, for each topic the distribution over words is known. Let K be total number of topics, w a particular word, and θ_d the topic distribution in document D. The probability of word w appearing in document D can be expressed as [5]

$$P_{lda}(w|D) = \sum_{z=1}^{K} P(w|z) P(z|\theta_D), \tag{2}$$

where:

- $P(w|z)$ is the probability of word w in topic z,
- $P(z|\theta_D)$ is the probability of topic z in document D.

Equation (2) can be interpreted as the probability of generating word w given document D.

Using (2), the probability of a set of words Q belonging to document D can be defined as

$$P(Q|D) = \prod_{w \in Q} P_{lda}(w|D). \tag{3}$$

Equation (3) can be interpreted as the probability of generating the set of words Q given document D.

Specifically, if we take Q to be a question and D a particular answer, (3) gives the probability of generating the question from the answer. As the probability increases and approaches 1, it is more likely that the answer is correct. Therefore, (3) can also be used to measure the similarity between a question and an answer.

Besides (2), the probability of word w appearing in document D can be expressed in terms of classical probability, as follows

$$P(w|D) = \frac{f_{w,D}}{|D|}, \tag{4}$$

where:

- $f_{w,D}$ is number of occurrences of the word w in document D,
- $|D|$ is total number of words in document D.

It is not guaranteed that word w belongs to document D, and therefore it is possible that (4) is zero. This would problematically cause (3) to also be zero. In the proposed application, this is not logical. For example, two documents could have many words in common and one that is not. According to (3), the similarity between two such documents would, incorrectly, be zero.

The solution to this problem is to use pseudo-counts. A pseudo-count is the default number of occurrences of words that do not exist in a document [5]. By extending (4) pseudo-counts are introduced as follows

$$P(w|D) = \frac{f_{w,D} + \mu \frac{c_w}{|C|}}{|D| + \mu}, \tag{5}$$

where:

- μ is the pseudo count and is determined experimentally,
- C is set of all possible answers,
- c_w is the number of occurrences of word w in C.

It is still possible that (5) results in zero, which is the case when word w from a question does not occur in any answer, i.e. $c_w = 0$. In that case, we sample a hyper-parameter β from a discrete Dirichlet distribution, as the default probability [1].

Equations (2) and (5) both define the probability of a word appearing in a document, but with different physical meanings. Equation (2) exploits topic similarity, while (5) exploits lexical similarity. Both similarities are important for correct answer selection, but not necessarily with the same importance, and therefore they can be combined as follows

$$P_{lda}(w|D) = \lambda \left(\frac{f_{w,D} + \mu \frac{c_w}{|C|}}{|D| + \mu} \right) + (1 - \lambda) \sum_{z=1}^{K} P(w \mid z) P(z|\theta_D) \tag{6}$$

The influence of each term is controlled by the parameter λ which takes values from the range $[0,1]$. By increasing λ, lexical similarity becomes more important.

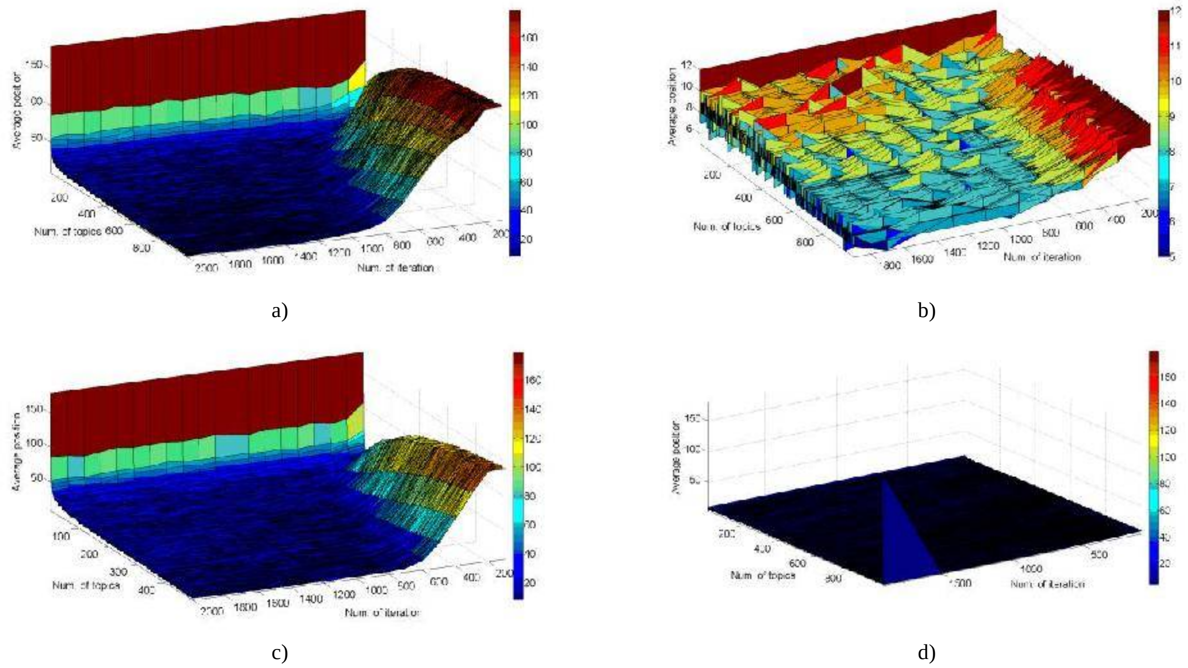

Figure 2. Average position of the correct answer (height) as a function of the number of iterations (x-axis) and the number of topics (y-axis): a) when using the cosine similarity measure and preprocessing steps 1—4 (see subsection II.E of the text); b) when using the query likelihood probability similarity measure and preprocessing steps 1—4; c) when using the cosine similarity measure and preprocessing steps 1—5; d) when using the query likelihood probability similarity measure and preprocessing steps 1—5.

In the presented experiments the value of μ is fixed to 200, while the value of λ is fixed to 0.2. The similarity between two documents is measured using (3).

C. Mallet

The topic modeling algorithm is implemented in the software framework *Mallet* [6]. Mallet is an open source Java-based package for topic modeling, natural language processing, and document clustering. More information regarding Mallet can be found in [6] and [7].

In the presented work, the *ParallelTopicModel* class is primarily used, which is an implementation of the Gibbs Sampling LDA topic modeling algorithm [8]. Also, the *TopicInferencer* class was used for inferring the topic distribution of each question.

D. Experimental Data

Experiments are conducted using question-answer pairs taken from two publically available real-world online collaborative question answering platforms: StackExchange[1] and Yahoo! Answers[2]. These portals enable users to collaborate in the form of asking question and proposing answers. They use a collaborative voting mechanism, which allows members of the community to vote up or down the questions and answers that they think are appropriate or not. Furthermore, a person who asks a question can mark one of the answers as the best answer.

Due to the nature of these sources, each question has multiple answers associated with it. We choose the best answer (marked by the person who asked the question) as the correct answer, resulting in one answer for each question (a question-answer, or Q-A, pair). If no best answer exists then the answer that received the most votes is used instead.

In the StackExchange dataset, 120 question-answer pairs from the health, fitness, and engineering categories are selected at random (the dataset therefore contains 360 Q-A pairs). Various stages of the Topic Modeling approach are optimized using this dataset (preprocessing steps and the similarity measure, see Section III.A), which is referred to in this manuscript as the training set (this is distinct from the training set used to train the topic modeling algorithm, which is formed from the answers that exist in the dataset being evaluated, and therefore the questions form the same dataset form the test set).

The final Topic Modeling approach is then compared to the reference approach (described in subsection F below) using test datasets extracted from the *health* category of the Yahoo! Answers website. A number of test sets are constructed, containing 100, 400, 700, 5 000, 10 000 and 20 000 randomly selected question-answer pairs.

E. Preprocessing

As is common in text based analysis, it is necessary to preprocess the raw data to make it suitable for automatic analysis. The following preprocessing steps are used (in order of application):

1. HTML tag removal;
2. Lowercase transformation;
3. Removing all non alphanumeric characters (smileys, special symbols etc.);
4. Stop word removal;
5. Lemmatization, using the Stanford Lemmatizer [9].

[1] Available from https://archive.org/details/stackexchange.

[2] Available from http://webscope.sandbox.yahoo.com/catalog.php?datatype=l.

Steps 1—4 are mandatory for effective application of the topic modeling algorithm as they remove irrelevant data (Steps 1, 3, and 4), and remove typographic variations that prevent automatic matching of syntactically equivalent text (Steps 2).

It is often not known *a priori* whether lemmatization or stemming should be used in document retrieval tasks [10] and we therefore conducted preliminary experiments to determine the best for this application. As such, the training set was used to determine whether stemming and augmenting documents with word synonyms improves performance. In addition to this, all combinations of stemming, lemmatization and synonym augmentation were tested. The Porter stemmer [11] was used in this part of the experimentation. It was experimentally found that the combination of steps listed above gave the best results.

F. *Reference Model*

The commonly applied Word Count approach is used as the reference model, in which the similarity of two documents is proportional to the number of words that they have in common. The TF-IDF measure [12] is used as the similarity measure and preprocessed answers and questions act as the input. Documents are preprocessed in exactly the same way as in the Topic Modeling approach to enable a fair comparison.

III. Results and Discussion

The first part of this section is dedicated to the choice of preprocessing stages and the selection of parameter values in the Topic Modeling approach, and the second part compares the final Topic Modeling algorithm to the Word Count approach.

A. *Preliminary Experiments*

This section presents the relevant results obtained during preliminary experimentation using the StackExchange training set. In Mallet, the LDA topic modeling algorithm is implemented using Gibbs sampling. Therefore, the performance of the algorithm depends upon the number of iterations used in the Gibbs sampling process.

The average position of the correct answer in the list of answers ranked by similarity is used to evaluate performance. These are presented as functions of the number of topics and iterations when using several variations of the algorithm in Fig. 2. The minimum values obtained in each of these experimental setups are presented in *Table* I.

TABLE I.
PREPROCESSING AND PARAMETER VALUES THAT RESULT IN THE MINIMUM AVERAGE POSITION OF THE CORRECT ANSWER IN EACH OF THE EXPERIMENTAL SETUPS, SEE FIGURE 2.

Min Avg. Position	Preprocessing Stages	Similarity Measure	# Iterations	# Topics
8	1—4	cosine	1200	693
5	1—4	query likelihood	300	51
8	1—5	cosine	1300	481
4	1—5	query likelihood	900	417

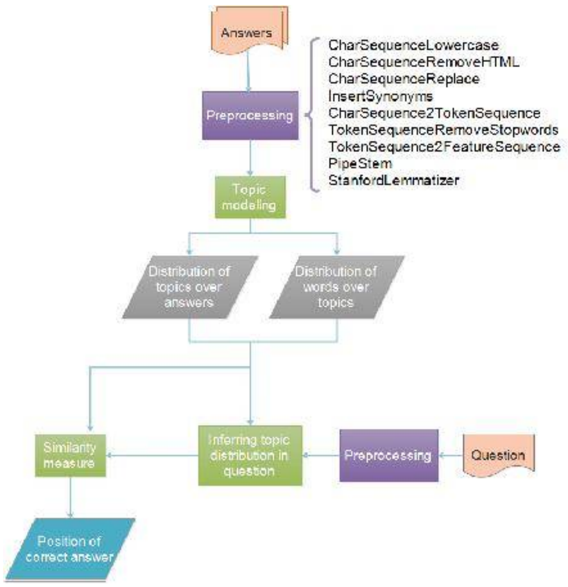
Answers
Preprocessing
CharSequenceLowercase
CharSequenceRemoveHTML
CharSequenceReplace
InsertSynonyms
CharSequence2TokenSequence
TokenSequenceRemoveStopwords
TokenSequence2FeatureSequence
PipeStem
StanfordLemmatizer
Topic modeling
Distribution of topics over answers
Distribution of words over topics
Similarity measure
Inferring topic distribution in question
Preprocessing
Question
Position of correct answer

Figure 3. Detailed diagram of the proposed question-answer matching algorithm.

It can be observed that within each of the two different preprocessing chains (Steps 1—4, and Steps 1—5) the query likelihood similarity measure gives the best results. Moreover, it achieves these results with fewer topics and iterations when compared to the cosine similarity measure. Indeed, it can be observed in Figure 2b that the performance is relatively constant when using this similarity measure in comparison to the cosine similarity measure results presented in Figures 2a and 2c (which performs badly when few iterations or topics are used). This indicates that exploiting information inherent to the topic modeling algrithm in the similarity measure leads to a simpler and more accurate model. It can also be observed that lemmatization (Stage 5) increases the number of topics needed to achieve a similar minimum average position when using the query likelihood measure, this can be explained by the fact that lemmatization reduces the diversity of words in the corpus, and therefore more topics are needed to distinguish between two documents' semantics.

The large peak in Fig. 2d appears when a very large number of topics is used. When the number of topics is greater than some specific number (which depends on the data, preprocessing steps, and similarity measure), then all answers have approximately equal topic distributions and all topics have approximately equal word distributions. As such all answers are equally probable, which causes the average position to be $N/2$, where N is the total number of answers~~.~~ If this peak were to be excluded the landscape would look similar to that in Fig. 2b.

Following this optimization stage, the design of the algorithm is fixed for comparison to the Word Count approach. Figure 3 presents a detailed view of the proposed approach. The preprocessing classes are listed in order of their application during subsequent experimentation. All preprocessing classes, except

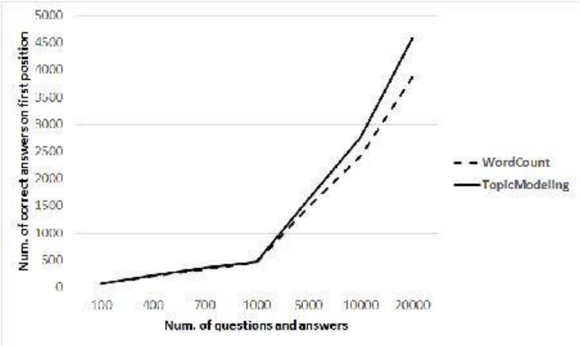

Figure 5. Number of correct answers in the first position for the Topic Modeling and Word Count approaches.

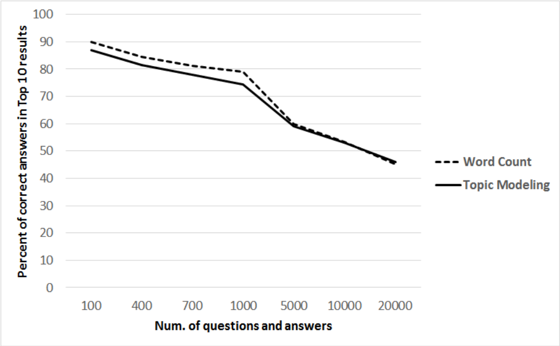

Figure 4. Percentage of correct answers in the top 10 most similar answers for the Topic Modeling and Word Count approaches.

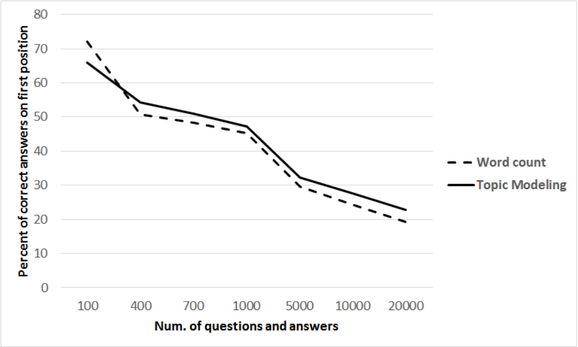

Figure 7. Percentage of correct answers in the first position for the Topic Modeling and Word Count approaches.

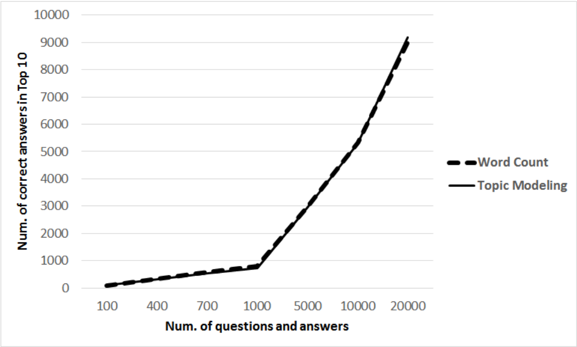

Figure 6. Number of correct answers in the top 10 most similar answers for the Topic Modeling and Word Count approaches.

InsertSynonyms, *PipeStem*, and *StanfordLemmatizer*, were taken from *Mallet*.

B. Main Results

It has been shown that the best results are obtained when using preprocessing steps 1—5 (as described in subsection II.E) and the query likelihood probability similarity measure.

In order to avoid overfitting and to present a fair comparison between the Topic Modeling and Word Count approaches, the algorithms are compared using the unseen Yahoo! Answers test datasets (containing 100, 400, 700, 1 000, 5 000, 10 000, 20 000 question-answer pairs).

The number of correct answers in first position, i.e. those with the highest similarity to the question according to (6) for the Topic Modeling approach and those with the highest TF-IDF measure for the Word Count approach, are presented in Fig. 5. When the number of Q-A pairs is less than 1000, both methods result in approximately equal performance (note that this does not mean that both methods give the same answers for each question). As the number of question-answer pairs increases, however, the difference between the two methods become more pronounced.

An additional important characteristic of each solution is the percent of correct answers in first position. These results are presented in Fig. 7 and it can be observed that as the number of documents increase, performance (according to this measure) decreases. Nevertheless, as the number of documents increases past 400, the Topic Modeling approach results in a slower performance decrease when compared to the Word Count approach.

Besides the number of the correct answers in first position, another important evaluation criteria is the number of times that the correct answer appears in the top 10 most similar results. This evaluation is presented in Fig. 6. Both methods result in almost equal performance. The differences in the results range from ±0.03% to ±0.01%, depending on the number of documents (when using a smaller number of documents, the difference is greater).

The percentage of questions in which the correct answer appears in the top 10 most similar results is presented in Fig. 4. This confirms what was previously discussed: when the number of documents is greater than 5000, the performance of each model is approximately equal.

All of the results presented in this section demonstrate that the Topic Modeling approach gives better performance when evaluating according to stricter criteria, such as the number of correct answers in first position. When the criteria is weaker, such as evaluating the top 10 results, the lexical part of the similarity measure becomes more significant and the model starts to act in a similar way to the Word Count approach. This is because of the model's inability to find semantically similar answers, therefore the contribution from the topic similarity term in (6) becomes almost zero. Nevertheless, it can be concluded that the Topic Modeling approach ranks correct answers higher in the list of results when compared to the Word Count approach.

Finally, the statistical significance of the differences between the first position results of each approach were tested using the Wilcoxon matched pairs test. The results of these statistical tests, using a significance level of 0.05, are presented in Table II.

These results show that the Topic Modeling approach to question answering gives significantly better results than the reference model.

IV. CONCLUSIONS

It has been shown that a question's topic structure, as well as its lexical structure is important for correct answer selection. An interesting question that arises when dealing with online, collaborative question-answer data sources is: What exactly is the correct answer? In this work, the best answer or the best rated answer was assumed to be correct, but there is no guarantee that this is true. Without any expert knowledge to judge which answer is truly correct, any system validated using this data cannot be completely trusted and should instead be used as an aid to find correct answers.

Experimentation has been conducted using two distinct, real-world datasets, giving weight to the generality of the findings presented in this manuscript. Design decisions for the Topic Modeling approach (preprocessing and parameter values) were made using one dataset and applied to another. It was shown to outperform the Word Count approach, however, the differences may be more pronounced when optimizing the parameter values using data derived from the same source.

The most appropriate application of this work would be in interactive systems, which contain a lot of similar and frequently asked questions. In this case the user can be presented with a list of possible answers to their question (say the top 10 most similar), as it has been shown that the discussed approach performs well in this setting. It has been shown that using the Topic Modeling approach, the user would find the correct answer more quickly (as it will be located higher in the ranked list of results) when compared to the Word Count approach.

To further improve the method, future work should be directed towards integrating and encoding expert knowledge within the discussed Topic Modeling approach.

TABLE II.
RESULTS OF THE WILCOXON MATCHED PAIRS TEST FOR STATISTICAL DIFFERENCE BETWEEN THE FIRST POSTION RESULTS FOR THE TOPIC MODELING AND WORD COUNT APPROACHES, STATISTICALLY SIGNIFICANT RESULTS ARE IN BOLD, STATISTICAL SIGNIFICANCE IS TAKEN TO BE 0.05.

# of docs	WordCount	TopicModeling	Significance
100	72	66	0.710
400	**203**	**217**	**0.031**
700	**338**	**357**	**0.021**
1000	**453**	**472**	**0.002**
5000	1484	1614	0.075
10000	**2422**	**2766**	**0.027**
20000	**3866**	**4576**	**0.000**

REFERENCES

[1] David M. Blei, et al., "Latent Dirichlet Allocation," Journal of Machine Learning Research,, vol. 3, pp. 993-1022, 2003.

[2] A. Celikyilma, et al., "LDA Based Similarity Modeling for Question Answering," In Proceedings of the NAACL HLT 2010 Workshop on Semantic Search, pp. 1-9., Los Angeles,2010.

[3] Z. Zolaktaf, et al., "Modeling community question-answering archives," presented at the Second Workshop on Computational Social Science and the Wisdom of Crowds NIPS, 2011.

[4] Guangyou Zhou, et al., "Improving Question Retrieval in Community Question Answering Using World Knowledge," Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2239 - 2245, 2013.

[5] B. Croft, et al., *Search Engines, Information Retrieval in Practice*, Addison-Wesley, 2010.

[6] A. McCallum. (2002)., *MALLET: A Machine Learning for Language Toolkit* [online]. Available http://mallet.cs.umass.edu

[7] D. Mimno, "Machine Learning with MALLET," Department of Information Science, Cornell University.

[8] T. Griffiths and M. Steyvers, "Finding scientific topics," Proceedings of the National Academy of Sciences, vol. 101, pp. 5228-5235, 2004.

[9] C. Manning, et al., "The Stanford CoreNLP natural language processing toolkit," in In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, 2014, pp. 55-60.

[10] V. Hollink et al., "Monolingual document retrieval for European languages," Information Retrieval, vol. 7, no. 1, pp. 33-52, 2004.

[11] M. Porter, "An algorithm for suffix stripping," Program, vol. 14, no. 3, pp. 130-137, 1980.

[12] H. Wu et al., "Interpreting TF-IDF term weights as making relevance decisions," vol. 26, no. 3, pp. 13:1-13:37, 2008.

[13] J. Ramos, "Using TF-IDF to determine word relevance in document queries," in Proceedings of the 1st Instructional Conference on Machine Learning,, Piscataway, 2003.