A Topic Model Scoring Approach for Personalized QA Systems

Hamidreza Chinaei[1], Luc Lamontagne[1], François Laviolette[1], and Richard Khoury[2]

[1] Department of Computer Science and Software Engineering, Université Laval
[2] Department of Software Engineering, Lakehead University

Abstract. To support the personalization of Question Answering (QA) systems, we propose a new probabilistic scoring approach based on the topics of the question and candidate answers. First, a set of topics of interest to the user is learned based on a topic modeling approach such as Latent Dirichlet Allocation. Then, the similarity of questions asked by the user to the candidate answers, returned by the search engine, is estimated by calculating the probability of the candidate answer given the question. This similarity is used to re-rank the answers returned by the search engine. Our preliminary experiments show that the reranking highly increases the performance of the QA system estimated based on accuracy and MRR (mean reciprocal rank).

Keywords: Personalized QA, User Modeling, Topic Modeling

1 Introduction

In QA systems, the system automatically finds answers to questions asked by the user. For instance, the user may ask *How old is the oldest complete genome?* The question is processed linguistically and search phrases or keywords are extracted, which are then used to retrieve documents, snippets (the passages returned by the search engine), or sentences [3]. Then a short answer is constructed from the search results and returned to the users. For instance, for the above question, the answer would be *700,000 years old*.

In personalized QA systems, the user may ask several questions either to complete a task or to learn several facts about a topic of interest. The user may need to find as much information as possible say about the *oldest complete genome*. Each time, the system returns an answer and may give extra information or suggestions for other questions of potential interest to the user.

Through its interaction with the user, the personalized QA system can learn incrementally user models based on the user browsed contents and from questions submitted to the system. We assume that a personalized QA system supports some user tasks. While accomplishing their tasks, the users should consult some documents (e.g., news articles). The learned models are then reused in the topic model function that we propose in this paper to estimate the probability of a candidate answer given a question. In this way, the system results can be assessed and prioritized using the learned user model.

Schlaefer [8] introduced the Ephyra question answering engine, a modular and extensible framework that allows to integrate multiple approaches to question answering in one system. We use an open version of the Ephyra engine, i.e., *openEphyra*[3], and build our personalized QA system on top of it. Our QA system includes for instance a logger that collects the documents read by user, a topic modeler that models the topics of interest to the user, etc.

In particular, we propose a reranking function to increase the evaluation of the passages, returned by the search engine, that may contain the answer. This is because the QA systems have a bigger chance to extract the correct answers if the top retrieved page./nippets contain the answers [6].

To this end, we propose a new probabilistic scoring approach based on the topics of the questions and the candidate answers. First, the set of interesting topics to the user is learned by applying a topic modeling approach, such as Latent Dirichlet Allocation (LDA) [1], on passages of interest to the user. Then, the similarity between questions asked by the user and candidate answers returned by the system is estimated by calculating the probability of the answer given the question. The probabilities are then estimated using the models learned by LDA. This topic model function is added to the set of QA functions (so called filters) that contribute to extracting and ranking the candidate answers to the question.

In the rest of this paper, we describe topic modeling. A brief introduction to the LDA method is presented in Section 2. We then explain in Section 4 our proposed method of scoring the candidate answers based on the topics of the question and those of the candidate answers. In Section 5, we briefly describe the architecture of our personalized QA system. We then explain in Section 6 our experiments and results. Finally, we conclude this work in Section 7.

2 Topic Modeling

Topic modeling techniques learn a set of possible topics from text documents of a corpus. The Latent Dirichlet Allocation (LDA) model is the pioneer topic modeling approach with interesting results. LDA is a latent Bayesian topic model which is used for discovering the hidden topics of documents [1]. In this model, a document can be represented as a mixture of the hidden topics, where each hidden topic is represented by a distribution over words occurred in the document.

Suppose we have the sentences shown in Table 1. Using the LDA method, we can automatically discovers the topics that are contained in the given collection (data). For example, given n asked topics, the LDA model learns the topics and the assignment of each topic to each sentence in the given data. The learned topics are represented using the words and their probabilities of occurring within each topic. The topics are presented in Table 2 with their top-10 words. The topic representation for topic A illustrates that this topic is about horse, dna, genome, etc. The topic representation for topic B illustrates that this topic B is about film, festivals, stars, etc. For any given snippet, then the algorithm produces in output a probability distribution over the set of all topics. This distribution is interpreted as the probability that the snippet belong to each of the learned topic, as presented in Table 3. Note that the examples shown in this section are from our collection as well as the results of the experiments of this work.

[3] `https://mu.lti.cs.cmu.edu/trac/Ephyra/wiki/OpenEphyra`

Table 1. The sample passage used to learn topic models with LDA.

Snippet 1:	Scientists have sequenced the oldest genome to date —and shaken up the horse family tree in the process. Ancient DNA derived from a horse fossil that's between 560,000 and 780,000 years old suggests that all living equids—members of the family that includes horses, donkeys, and zebras—shared a common ancestor that lived at least 4 million years ago, approximately 2 million years earlier than most previous estimates. …
Snippet 2:	Cameron Diaz is ready to make it a hard-knock life for Annie. The actress has just secured the role of Miss Hannigan in the long-simmering Annie remake.Prior to nabbing the part, Sandra Bullock had been the rumored frontrunner for the mean-spirited, often drunken matron who runs the orphanage in the story. …
Snippet 3:	Riyaj Shamsudeen offers an in-depth look at Oracle Database 12c , which he calls a 'true cloud database', bringing a new level of efficiency and ease to database consolidation.…
…	…

Table 2. Topics learned by LDA from the passage presented in Table 1.

Topic A:	dna years horse drone genome ago year time previous million
Topic B:	film jolie festival angelina brad vice year baalbek season trip
Topic C:	food pizza place top made water sugar high set side
Topic D:	space earth vision satellite yankees closer stereo images bridgeman alex
…	…

Table 3. The LDA topic assignments to the passages presented in Table 1.

Topics	A	B	C	D	…
Snippet 1:	**44%**	3%	25%	2%	…
Snippet 2:	4%	**65%**	11%	5%	…
Snippet 3:	4%	5%	2%	5%	…
…	…	…	…	…	…

Formally, given a document in the form of $\boldsymbol{d} = (w_1, \ldots, w_M)$ in a document corpus $\boldsymbol{D}$, and given a parameter N that represents the total number of different topics to be found by the algorithm; the LDA model learns two parameters:

1. The parameter $\boldsymbol{\theta}$ which is generated from the Dirichlet prior $\boldsymbol{\alpha}$.
2. The parameter $\boldsymbol{\beta}$ which is generated from Dirichlet prior $\boldsymbol{\eta}$.

The first parameter, $\boldsymbol{\theta}$, is a vector of size N for the distribution of the hidden topics z. The second one, $\boldsymbol{\beta}$, is a matrix of size $M \times N$ in which column j stores the probability of each word given the topic z_j.

Figure 1 shows the LDA model in the plate notation in which the boxes are plates that represent replicates. The shaded nodes are the observation nodes, i.e., the words w. The unshaded nodes z represent hidden topics. To build a generative model, LDA performs as follows:

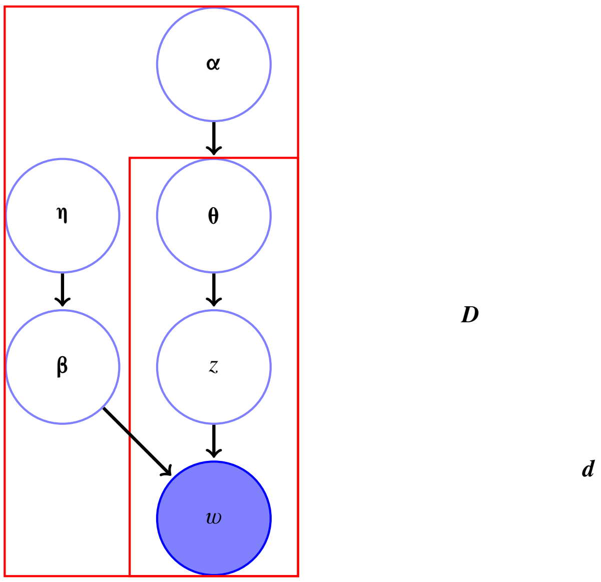

Fig. 1. Latent Dirichlet allocation. The boxes are plates that represent replicates. The shaded nodes represent observations and the unshaded nodes represent hidden topics.

1. For each document $\boldsymbol{d}$, a parameter $\boldsymbol{\theta}$, is drawn for the distribution of hidden topics based on multinomial distribution with the Dirichlet parameters $\boldsymbol{\alpha}$.
2. For each document set $\boldsymbol{D}$, parameter $\boldsymbol{\beta}$ is learned for the distribution of words given topics. Given each topic z, the vector $\boldsymbol{\beta}_z$ is drawn based on multinomial distribution with the Dirichlet parameters $\boldsymbol{\eta}$.
3. Generate $w_{i,j}$, the jth word in the document i, as follows:
   (a) Draw a topic $z_{i,j}$ based on the multinomial distribution with the parameter θ_i.
   (b) Draw the word $w_{i,j}$ based on the multinomial distribution with the parameter $\phi_{z_{i,j}}$.

3 Related Work

Topic modeling has been used in few QA systems [3,2,4]. The closest to our research effort is the work of [3]. They proposed an LDA-based similarity measure for QA. Their proposed degree of similarity DES(q, a), for a question q and a candidate answer a, based on the similarity of the topic distribution of the question and that of the candidate answers. The degree of similarity is also based on the distributions of question's words, given each topic, and the candidate answers' words given each topic. The similarity is then calculated using an information-theoretic similarity metric, called transformed information radius (IR), which is based on Kullback-Liebler (KL) divergence. Note that in IR measurement, the divergence is transformed into a distance measure [5].

4 A Topic Model Scoring Approach for QA

To account for user information needs in a personalized QA, we propose a new probabilistic scoring approach based on the topics of questions and candidate answers. In a probabilistic approach, the score of a candidate answer, the snippet returned by a search engine, can be estimated as *the probability of the answer given the question*. Formally, the score can be calculated as:

$$\text{score}_a = p(a|q) \tag{1}$$

where q is the question submitted by the user and a is the candidate answer returned by the search engine. Using Bayes rule, Equation 1 becomes $\text{score}_a = \frac{p(a,q)}{p(q)}$. Since the denominator is the same for all candidate answers, we can remove the denominator. Thus, the topic model score for a candidate answer can be approximated as $p(a, q)$:

$$\text{score}_a \propto p(a, q) \tag{2}$$

To learn the expression in Equation 2 from data, we should consider the information need of the user (which can be learned through topic modeling). To do so, using the law of total probabilities, we reformulate the topic model score:

$$p(q, a) = \sum_z p(a, q, z)$$

where z is the topic learned from training data. Then, using conditional rule, we can insert the user topic models in the estimation function:

$$\begin{aligned} &= \sum_z p(a, q|z)\ p(z) && (3) \\ &= \sum_z p(a|z)\ p(q|z)\ p(z) \end{aligned}$$

In the last equality, we assume that given the topic the answer is independent from the question.

We estimate three probability distributions in Equation 3 from topic models learned by LDA. Note that LDA learns a set of topics from the passages of interest to the user. Given a topic z, and question q, then $p(q = (w_1, \ldots, w_{|q|})|z)$ is calculated based on the bag of word assumption that is considered in LDA. Thus, we assume that $p(q = (w_1, \ldots, w_{|q|})|z) = p(w_1|z) \ldots p(w_{|q|})$. Similarly, we calculate $p(a = (w_1, \ldots, w_{|a|})|z)$, where a is a candidate answer. Note however that longer answers will result to smaller probabilities; thus, we normalize Equation 3 by taking it to the power of $1/|a|$ similar to [7].

Each $p(w_i|z)$ is estimated based on the learned model by LDA stored in β (where w_i is a word in the question or the answer). If some words do not occur in the training data, no probabilities are considered for them. In addition, $p(z)$, the learned distributions over topics are learned by LDA, stored in the vector θ. For the passages that do not occur in training data, the distribution over topics can be inferred using LDA inference method [1], as we have done in our experiments.

Our work is similar to that of [3] in using LDA in QA systems. However, [3] used their degree of similarity (that is an information-theoretic similarity metric) particularly as a feature of an SVM classifier. In contrast, we proposed probabilistic scoring approach using topic modeling that is used as a ranking function. This ranking function is added to the set of QA answer extractio./election functions. The topic model function contributes to ranking the candidate answers to the question. For each extracted answer, its score is calculated from Equation 3. In particular, we use the topic distributions, $p(z)$, as a prior probability over the user's topic of interest, whereas [3] use the topic distributions as a factor in their measure of degree of similarity.

5 QA Architecture

Figure 2 shows the architecture of our personalized QA system. Our personalized QA system supports user tasks. While accomplishing theirs tasks, the users should consult some documents (e.g., news articles). The document, read by the user, are logged to be used as input to learn the user model. The learned models are then reused in our proposed topic model function to estimate the probability of a candidate answer given a question. Our personalized QA system is built using open source resources. We developed a graphical user interface (GUI) allowing the users to submit their queries through. The GUI also displays the topics learned for the users based on their questions and browsed snippets/documents. We use RSSOwl for the news feed reader, which is a free news reader software[4]. The core of the QA system is built on the top of the OpenEphyra QA system, an open source QA system developed at CMU. In addition, we use Lucene as our search engine, which is an open source text search engine[5]. The user model is learned using Mallet, an open source machine learning software[6].

6 Experiments

For our experiments, our users collected 1,872 news articles from their topics of interests. These articles were used as our collection in the QA system. Our users then asked 100 factoid questions. The answers for the questions have to occur in the documents read by them (the 1,872 logged news articles). For testing, we manually extracted the answers from the collection (one to five answers based on the question). We then made patterns of each answer (as regular expressions), to be used in the automated evaluation module. This module verifies if each answer returned by the QA system includes the answer's pattern.

For each question, we collected the snippets returned by the Lucene search engine. We then regarded the question and all the returned snippets as a document to be used in LDA, so called *the question document*. In this case, the prior probability of topic

[4] `http://www.rssowl.org/`
[5] `https://lucene.apache.org/core/`
[6] `http://mallet.cs.umass.edu/`

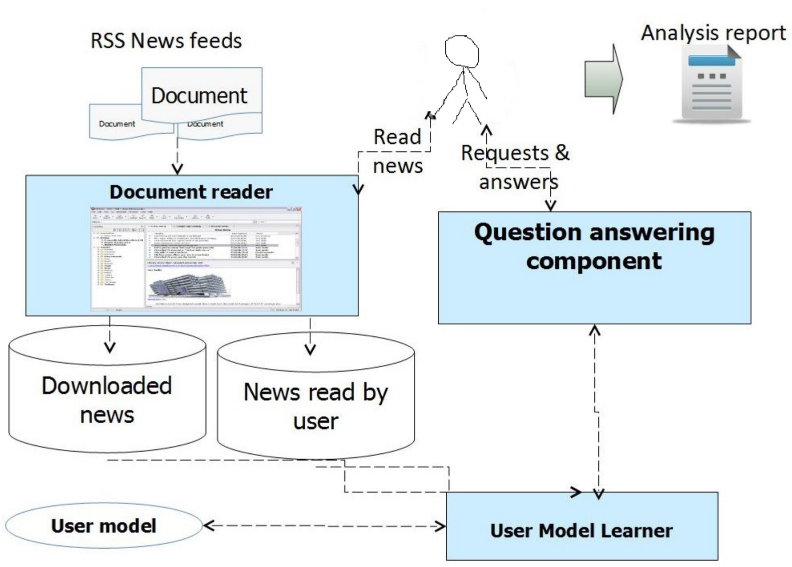

Fig. 2. The personalized QA System architecture.

distributions, $p(z)$, for each question is the learned topic distribution of the question document. Specifically, we used LDA to learn 10, 20, 40, 60, 80, and 100 topics using parameters $\alpha = 50$ and $\beta = 1$. For each learned model, the probability of each word given each topic was learned by Mallet for the total of 34,669 distinct words.

We then implemented the topic model function for our QA system. In particular, the topic model function takes the top 10 results, returned by Lucene. For each result, if it includes the question's target, the topic model function scores the result based on the topic model scoring in Equation 3. Then, the topic model score is normalized over all results that got a topic model score (the top 10 results returned by Lucene that contained the question's target). The score for all other results remains minus infinity (the default value in openEphyra). Then, all results returned by Lucene are passed to the result of default functions (filters) in openEphyra, so that the answer extraction and selection be completed.

The topic model performance is evaluated based on the QA system performance. That is, the system performance is calculated in the presence and absence of topic model function. We calculated the following measures: MRR, top1-accuracy, top5-accuracy, and top10-accuracy. The MRR is the mean reciprocal rank. The top1-accuracy, top5-accuracy, and top10-accuracy are respectively the accuracy for the top1, top5, and top10 results [9].

The results are shown in Table 4. The results demonstrate that the topic model function significantly improves the QA system's performance with any number of topics. The table shows consistent results for MRR, top1-accuracy, top5-accuracy, and

Table 4. The QA system's performance on the news data.

	default functions	10 topics	20 topics	40 topics	60 topics	80 topics	100 topics
MRR	0.20	0.28	**0.30**	0.27	0.29	0.25	0.28
top1-acc	0.16	0.23	**0.24**	0.22	0.24	0.20	0.23
top5-acc	0.25	0.34	**0.37**	0.34	0.35	0.31	0.34
top10-acc	0.27	0.38	**0.40**	0.37	0.38	0.35	0.36
all-acc	0.27	0.38	**0.40**	0.37	0.38	0.35	0.36

Table 5. The QA system's performance on the news data, using two-fold cross validation.

	default functions	20 topics
MRR	0.22	**0.29**
top1-acc	0.17	**0.24**
top5-acc	0.27	**0.38**
top10-acc	0.29	**0.38**
all-acc	0.29	**0.38**

top10-accuracy. The best performance is achieved when 20 topics are used where MRR drastically increases from 0.20 to 0.30 and top-10-accuracy from 0.27 to 0.40, i.e., 48% improvement.

We also performed a two-fold cross validation on the same data set using 20 topics (the rest of parameters remain the same). We used half of the data for training, i.e., learning the LDA model, particularly the vector β. The other half was used for testing. In particular, the topic distributions were inferred for the testing data, i.e., the vector θ used as the prior in Equation 3.

Specifically, we randomly took half of the questions and made their question document, similarly we made the question documents for the second half to make two folds. We then used one of the folds for training, and the second half for testing. The prior topic distributions, $p(z)$, were inferred from the question document of the tested questions (using the inference approach introduced in LDA [1]). We then calculated the MRR and top1-accuracy, top5-accuracy, top10-accuracy, and accuracy. We calculated the same measures on the first half while we used the second half for training. The results summarized in Table 5 are averaged over the two folds. These experiments show consistent results with the results shown in Table 4. That is, the topic model filter highly increases the QA system performance on the news dataset.

7 Conclusions

In this paper, we proposed a new probabilistic scoring approach based on the topics of the question and the candidate answers particularly useful for personalized QA systems. In particular, the score of a candidate answer is estimated by calculating the probability of the candidate answer given the question. Our preliminary experiments show that the reranking based on the user topics highly increases the performance of the QA system. In the future work, we will run our experiments on the TREC QA dataset. In addition, we are going to use hierarchical topic modeling approach in the place of flat LDA.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
2. Cai, L., Zhou, G., Liu, K., Zhao, J.: Learning the latent topics for question retrieval in community qa. In: Proceedings of 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand (2011)
3. Celikyilmaz, A., Hakkani-Tur, D., Tur, G.: LDA based similarity modeling for question answering. In: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT'10), Workshop on Semantic Search (SS'10). pp. 1–9. Association for Computational Linguistics, Los Angeles, California, USA (2010), http://dl.acm.org/citation.cfm?id=1867767.1867768
4. Ji, Z., Xu, F., Wang, B., He, B.: Question-answer topic model for question retrieval in community question answering. In: Proceedings of the 21st ACM international conference on Information and knowledge management, Maui, Hawaii, USA (2012)
5. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, USA (1999)
6. Pala Er, N., Cicekli, I.: A factoid question answering system using answer pattern matching. In: Proceedings of the 6th International Joint Conference on Natural Language Processing (IJNLP '13). Nagoya, Japan (October 2013), http://www.aclweb.org/anthology/I13-1106
7. Qu, M., Qiu, G., He, X., Zhang, C., Wu, H., Bu, J., Chen, C.: Probabilistic question recommendation for question answering communities. In: Proceedings of the 18th International Conference on World Wide Web (WWW '09). Madrid, Spain (2009), http://doi.acm.org/10.1145/1526709.1526942
8. Schlaefer, N., Gieselmann, P., Schaaf, T., Waibel, A.: A pattern learning approach to question answering within the ephyra framework. In: Petr Sojka et al. (Eds.) Proceedings of the 9th International Conference on Text, Speech and Dialogue (TSD '06). Brno, Czech Republic (2006), http://dx.doi.org/10.1007/11846406_86
9. Voorhees, E.M.: Overview of the TREC 2004 Question Answering Track. In: Text REtrieval Conference (TREC). vol. Special Publication 500-261. National Institute of Standards and Technology (NIST) (2004)