Ranking Answers by Hierarchical Topic Models

Zengchang Qin[1,⋆], Marcus Thint[2], and Zhiheng Huang[1]

[1] BISC Group, EECS Department, University of California Berkeley, USA
[2] Computational Intelligence Group, Intelligent Systems Lab, BT Group, UK
zcqin@berkeley.edu, zhiheng@cs.berkeley.edu, marcus.2.thint@bt.com

Abstract. Topic models are hierarchical probabilistic models for the statistical analysis of document collections. It assumes that each document comprises a mixture of latent topics and each topic can be represented by a distribution over vocabulary. Dimensionality for a large corpus of unstructured documents can be reduced by modeling with these exchangeable topics. In previous work, we designed a multi-pipe structure for question answering (QA) systems by nesting keyword search, classical Natural Language Processing (NLP) techniques and prototype detections. In this research, we use those technologies to select a set of sentences as candidate answers. We then use topic models to rank these candidate answers by calculating the semantic distances between these sentences and the given query. In our experiments, we found that the new model of using topic models improves the answer ranking so that the better answers can returned for the given query.

1 Introduction

Question answering (QA) is an important area in information retrieval. It involves query analysis, recognition of relevancy and search. In our previous work, we designed a deduction engine which supports a "multi-pipe" process flow to handle keyword search as well as some special prototypes [12]. By reasoning based on these prototypes, our systems can improve over classical keyword matching approach. However, the fundamental problem for learning from text and natural language processing is how to learn the 'meaning' and 'usage' of words in data-driven fashion. How to model polysemy and synonymy of words become the first step towards semantic understanding.

Latent semantic indexing (LSI) [5] a is well-known technique which partially addresses this problem. LSI makes three claims: semantic information can be derived from a word-document co-occurence matrix; that dimensionality reduction is an essential part of this derivation; and the words and documents can be represented as points in Euclidean space. The key idea is to map high-dimensional vocabulary count vectors to a lower dimensional representation by using Singular Value Decomposition and selected largest eigenvalues. Due to the unsatisfactory theoretical foundation, Hofmann developed probabilistic latent semantic indexing (pLSI) [9] based on a mixture decomposition derived from a latent

⋆ School of Automation Science, Beihang University, Beijing, China.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 103–112, 2009.

class model. pLSI models each word in document as a sample from a mixture of topics. Each word is generated from a single topic, and different words in a document may be generated from different topics. However, pLSI does not provide a probabilistic model at the document level. Blei *et al.* [4] later proposed a three-level hierarchical Bayesian model called latent Dirichlet allocation (LDA). In LDA, document level is modeled by a Dirichlet distribution. LDA has been heavily cited in machine learning community for its effectiveness and theoretical soundness.

Griffiths and Steyvers [7] proposed similar models for learning natural language by using these latent topics, we call these models "topic models". In this paper, we discuss an application using topic models proposed in [13] for answer ranking in a question answering system. This paper is organized as follows: technical details of topic models are introduced in section 2. In the section 3, after a brief introduction on our previous work on question answering system, we discuss how to rank answers by using topic models. Some experimental results on a small corpus are presented in section 4. The conclusions and discussions are given in the final section.

2 Topic Model

The study of latent topics is popularized by Hofmann's work [9] on probabilistic latent semantic indexing. In this model, a document label d ($d = 1, \cdots, D$) and a word w_i ($i = 1 \cdots W$) are conditionally independent given a latent topic z:

$$p(d, w_i) = p(d) \sum_z p(w_i|z)P(z) \tag{1}$$

The model learns the topic mixture $p(z|d)$ only for those training documents and the size of parameters grows linearly with the corpus size M. LDA overcomes these problems by treating the topic mixture weights as a k-dimensional random variable θ with Dirichlet distribution [4].

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1} \tag{2}$$

Then, the probability of a document $\mathbf{w}$ becomes the conditional probability given hyper-parameters α and γ [4]:

$$p(\mathbf{w}|\alpha, \gamma) = \int p(\theta|\alpha) \prod_{i=1}^{W} \sum_{z_i} p(z_i|\theta) p(w_i|z_i, \gamma) d\theta \tag{3}$$

where γ is a $k \times W$ matrix with $\gamma_{jl} = P(w_i = l|z_i = j)$ and $p(w_i|z_i, \gamma)$ is a multinomial probability conditioned on the topic z_i. More details about LDA can be found in [4].

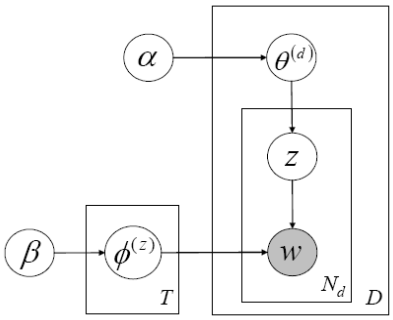

Fig. 1. A general structure of a topic model. The shaded node is the observable variable and others are latent variable. Relationships between latent topic z, random variable θ, ϕ and hyperparameters α and β are discussed in Section 2 below.

Griffiths and Steyvers [7,13] explored a variant of LDA by introducing another multinomial random variable ϕ to smooth the word distribution in every topic. ϕ is also a Dirichlet distribution governed by the hyperparameter β.

$$p(\phi|\beta) = \frac{\Gamma(\sum_{i=1}^{k} \beta_i)}{\prod_{i=1}^{k} \Gamma(\beta_i)} \phi_1^{\beta_1 - 1} \cdots \phi_k^{\beta_k - 1} \tag{4}$$

Figure 1 shows the graphical relations between these variables. The topic model is a generative model [10] for documents. Documents are assumed to be generated following a simple probabilistic procedure. Each document is a mixture of topics and each topic is a probability distribution over words [1,13]. There are set of parameters governed by some prior distributions and these priors are defined by Dirichlet distributions. The variables ϕ, θ and z (the assignment of word tokens to topics) are latent variables and hyperparameters α and β are constants in the model. The inner plate over z and w illustrates the repeated sampling of topics and words until N_d words have been generated for document d. The plate surrounding $\theta^{(d)}$ illustrates the sampling of a distribution over topics for each document d for a total D documents (the whole corpus). The plate for $\phi^{(z)}$ illustrates the sampling of word distributions for each topic z until T topics have been generated [13]. Figure 2 shows the geometric interpretation of the relations between document-topic and topic-word. Suppose we only have 3 words in our vocabulary, then a topic can be represented as a probability distribution on these 3 words. Therefore, a topic must lie on the simplex of these 3 words. Similarly, if we only have 3 topics, a document can be represented as a point on the simplex of topics.

There are a few methods to estimate the parameters for graphical models, such as variational methods [4] and Expectation Maximization (EM) algorithm [14]. In this paper, we use Gibbs sampling algorithm to consider each word token in the text in turn, and estimate the probability of assigning the current word token to each topic, conditioned on the topic assignments to all other word

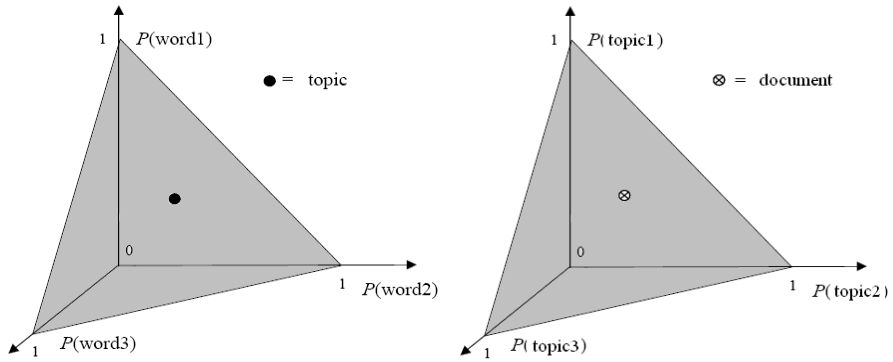

Fig. 2. A topic is a distribution over observable words and a document is a distribution of latent topics. In this simple case, there are 3 words and 3 topics. A document can be represented as a point on the surface of a simplex of topics. A topic can be regarded as a point of on the simplex of words.

tokens. The probability $z_i = j$ (assign token i to topic j) is conditioned on w_i (current word token), $\mathbf{z}_{-i}$ (topic assignments of all other word tokens), and d_i (current document).

$$P(z_i = j|\mathbf{z}_{-i}, w_i, d_i, I) \propto \left(\frac{C^{WT}_{w_i,j} + \beta}{\sum_{w=1}^{W} C^{WT}_{wj} + W\beta}\right)\left(\frac{C^{DT}_{d_i,j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d_i,t} + T\alpha}\right) \tag{5}$$

where I refers to all other observed information such as all word and document indices $\mathbf{w}_{-i}$ $\mathbf{d}_{-i}$ and hyperparameters α and β. $\mathbf{C}^{WT}$ and $\mathbf{C}^{DT}$ are the matrices of counts with dimensions $W \times T$ and $D \times T$, respectively; C^{WT}_{wj} contains the number of times word w is assigned topic j, not including the current instance i and C^{DT}_{dj} contains the number of times topic j is assigned to some word token in document d, not including the current instance i. The estimate of posterior probability distributions for θ (topic-document distribution) and ϕ (word-topic distribution) directly obtained by [13]:

$$\hat{\phi}^j_i = \frac{C^{WT}_{ij} + \beta}{\sum_{k=1}^{W} C^{WT}_{kj} + W\beta} \tag{6}$$

$$\hat{\theta}^d_j = \frac{C^{DT}_{dj} + \alpha}{\sum_{k=1}^{T} C^{DT}_{dk} + T\alpha} \tag{7}$$

The detailed sampling algorithm and justification of above equations are available at [7,13].

Given a new document $\mathbf{w}$ which is not contained in the training corpus D, how can we represent the document by a distribution of topics? For a particular topic t_j, according to Bayes' rule:

$$p(t_j|\mathbf{w}, D) \propto p(\mathbf{w}|t_j, D)p(t_j|D) \tag{8}$$

where the 'prior' probability of topic t_j (conditional on the training corpus) can be calculated by:

$$p(t_j|D) = \frac{\sum_{i=1}^{W} C_{w_i,j}^{WT}}{\sum_{i=1}^{W}\sum_{j=1}^{T} C_{w_i,j}^{WT}} \tag{9}$$

and the likelihood is:

$$p(\mathbf{w}|t_j, D) = \sum_{d}^{|D|} \prod_{i} p(w_i|t_j)p(t_j|d)p(d) \tag{10}$$

For training the topic model in Figure 1, we have to predefine the number of topics. Automatic determination of the number of clusters has been a persisting challenge in machine learning though some work has tried to address this problem [1]. In this work, we simply decide the topic number based on the size of corpus. For example, we set 20% of the corpus size as the topic number. For a new document, we can then calculated the topic distribution according to equation 8. Some recent work are looking at more general topic models by considering the correlation between topics [3] and the time evolution of topics [2] in large corpus. In this paper, we assume the topics are conditionally independent without evolutionary properties because the corpus we are testing only contains some simple texts from user manuals. In the following section, we discuss the use of this model for ranking answers given a query.

3 Ranking Answers

In our previous work [11,12], we described a hybrid reasoning engine which supports a "multi-pipe" process flow to handle Precisiated Natural Language (PNL) based deduction as well as other natural language phrases that do not match PNL protoforms. The resulting process flows in a nested form, from the inner to the outer layers: (a) PNL-based reasoning where all important concepts are pre-defined by fuzzy sets, (b) deduction-based reasoning which enables responses drawn from generated/new knowledge, and (c) key phrase based search when (a) and (b) are not possible. The design allows for two levels of response accuracy improvement over standard search, while retaining a minimum performance level of standard search capabilities.

In this research, we add the topic model to the end of our pipeline design. We use topic models to measure the 'semantic' distance between each candidate answers and the query. Closer the semantic distance implies closer semantic relation between the answers and the query. This information is used to re-rank the candidate answers selected by other NLP tools.

One challenge in this application is that, we usually have a short list of keywords for the query and each candidate answer sentence. Hence, the topic distribution calculated on these words may not be very accurate when there is some semantic meaning behind the words we cannot capture. For example, in a query "Where can I buy a Ford near Berkeley", a human can understand that

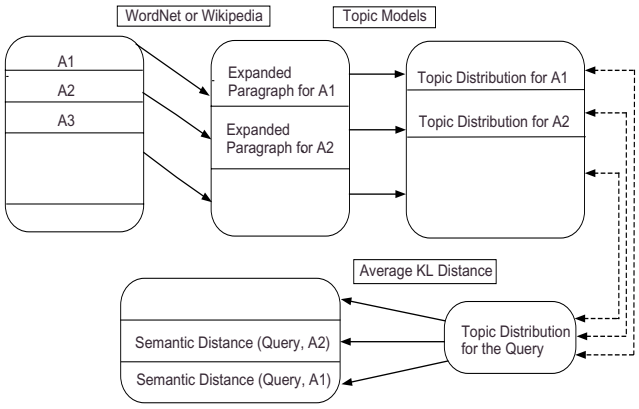

Fig. 3. Topic Models for answer ranking: we selected a subset of candidate answers by using keyword search or other NLP techniques. Each keyword in the answer is expanded by WordNet or Wikipedia. We calculate the topic distributions for answers and the query and compare the distance between them. The average Kullback-Leibler divergence is used to measure the semantic distances between candidate answers and the query.

someone is trying to buy a car of brand Ford. Even if there are some sentences about auto dealer in San Francisco Bay Area, this useful information cannot be found by key-word search and other classical NLP tools. We propose a sentence augmentation approach to expand each short sentence (including query) into a paragraph by using WordNet [8] or Wikipedia. We use the descriptions on those keywords (mainly nouns) to formulate a new paragraph which later on will be used to calculate topic distribution. By using Wikipedia, we would select the first two sentences. For example, word 'Ford' becomes

> Ford Motor Company is an American multinational corporation and the world's third largest **auto** maker based on worldwide vehicle sales. Based in Dearborn, Michigan, a suburb of Detroit, the automaker was founded by Henry Ford and incorporated in June 16, 1903.

and 'Berkeley' becomes

> Berkeley is a city on the east shore of **San Francisco Bay** in Northern California, in the United States. Its neighbors to the south are the cities of Oakland and Emeryville.

Although the expansion injects noise to the query and answer sentences, we can see that addition of valuable keywords 'auto' and 'San Francisco Bay' helps to semantically link 'Ford' and 'Berkeley', respectively. The basic process is illustrated in Figure 3. The semantic distance between answer A and query Q is measured by the average Kullback-Leibler (AKL) Distance (or divergence):

$$AKL(A||Q) = \frac{KL(A||Q) + KL(Q||A)}{2} \tag{11}$$

TOPIC_2		TOPIC_5		ToPIC_9	
rule	0.02464	software	0.05364	email	0.16295
government	0.02113	process	0.04024	send	0.07976
famili	0.01761	person	0.03689	forward	0.03816
Tang	0.01410	softwar	0.03019	spam	0.03123
central	0.01410	computer	0.02684	SPAM	0.02430
century	0.01410	develop	0.02684	nformation	0.02430
court	0.01410	personal	0.02684	messag	0.02430
dynasty	0.01410	intranet	0.02348	Forward	0.01737
earli	0.01410	microsoft	0.02013	creat	0.01737
early	0.01410	packag	0.02013	open	0.01737
empir	0.01410	comput	0.01678	Business	0.01390
greatest	0.01410	costs	0.01678	simpli	0.01390
canal	0.01058	document	0.01678	arriv	0.01043
collaps	0.01058	separate	0.01678	attach	0.01043
dynasti	0.01058	advers	0.01343	attachment	0.01043
great	0.01058	anti-virus	0.01343	inbox	0.01043
han	0.01058	applications	0.01343	virus	0.01043
li	0.01058	close	0.01343	Attachment	0.00697
militari	0.01058	control	0.01343	Express	0.00697
octob	0.01058	development	0.01343	PowerPoint	0.00697
periods	0.01058	signific	0.01343	Ridnour	0.00697
pow	0.01058	1:1	0.01008	Subject	0.00697
prosper	0.01058	Personal	0.01008	activate	0.00697
stabil	0.01058	Switch	0.01008	department	0.00697
9th	0.00707	databas	0.01008	devic	0.00697
Bai	0.00707	edit	0.01008	easi	0.00697
Dynasties	0.00707	form	0.01008	ent	0.00697
Dynasty	0.00707	remove	0.01008	express	0.00697
Emperor	0.00707	restor	0.01008	help	0.00697
Empress	0.00707	variant	0.01008	outgoing	0.00697

Fig. 4. The first 30 words with their distributions in the samples of topic 2, 5 and 9

The score of a set of candidate answers A_i $(i = 1, \cdots, |\mathbf{A}|)$ is calculated by:

$$S(A_i) = \frac{AKL(A_i||Q)^{-1}}{\sum_{i=1}^{|\mathbf{A}|} AKL(A_i||Q)^{-1}} \tag{12}$$

This normalization will assign higher scores to the answers which are more semantically close the query.

4 Experimental Studies

In order to test the effectiveness of the topic model, we compare the keyword based search engine and the new system of combining keyword search and the topic model. We use Lucene [6] as the standard keyword search engine. We applied the topic model to the results from Lucene and the overall score is unweighted sum of the Lucene score and the topic model score. We tested these two systems on the test corpus which contains 110 documents. Over half of them are about telecommunications, the other half documents are about sports, travel, history and other random documents from the Internet. We proposed 90 questions whose correct answers are available in the corpus.

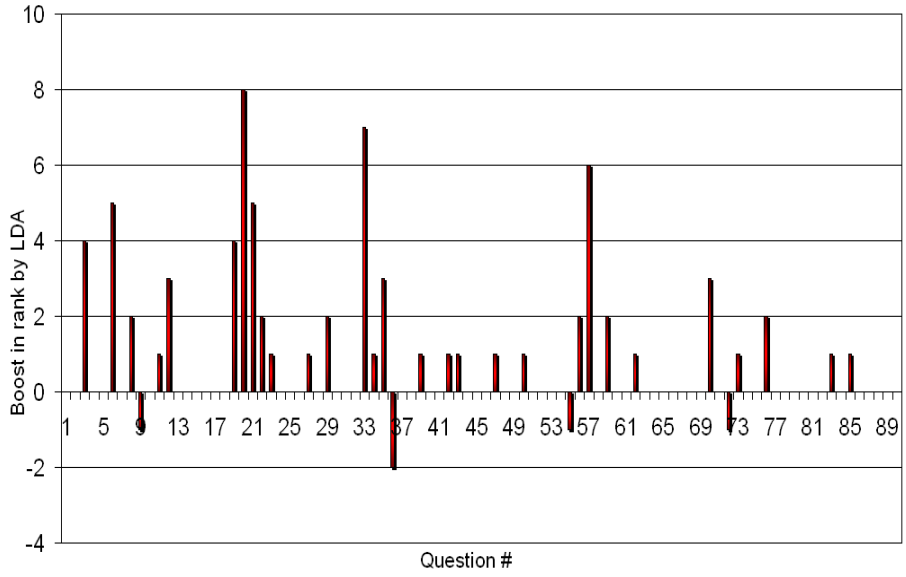

Fig. 5. The rank difference for Lucene and Lucene+LDA on the BT corpus. The horizontal axis is the question number and the vertical axis is the boosted rank of the best answer for that question.

In our offline training, we use 20 topics (about 20% of the corpus size) and the first 30 most frequent words in topic 2, 5 and 9 are shown in Figure 4. For each topic, the numbers after words are the probabilities of these words in this topic. Although there is no explicit name for each topic, we still can see that topic 2 is about history (Tang dynasty of China), topic 5 is about computer and internet, and topic 9 is about spam emails. Since these words occur together more frequently, we may consider that they are semantically close. By using the topic model, we can cluster these words into one semantic topic.

Figure 5 illustrates the results of the two systems (Lucene and Lucene+LDA) on a test BT corpus. Each bar represents the difference between two ranks given by Lucene and Lucene+LDA on the same question. For example, given Question 6, the best answer is ranked as 10th by Lucene and ranked 5th by Lucene+LDA. We obtained the difference 5 by using:

$$D = Rank_{(Lucene)} - Rank_{(Lucene+LDA)} \tag{13}$$

where $Rank_{Lucene}$ is the cardinal number of the best answer rank given by Lucene. Therefore: if the difference is positive, it means that the new system boosts the best answer with high ranks; if negative, it means that the new system actually performs worse ranking than Lucene due to the extra augmentation noise; if zero, it means that there is no significant differences for these two systems. In the 90 questions, the new system improve ranking of the basic keyword matching approach for 29 questions. There are also 4 questions that the new

system obtained worse ranking. For the remaining 57 questions, these two systems give the same rankings of the best answers. The bad rankings for questions #9, #50 #83 and #88 are likely due to the noise injected by keyword augmentation. Further research is still needed to reduce such noise in terms of an optimal augmentation strategy.

The following output is an example of better ranking by using topic models. Given the question: **What is the oil price per barrel in October?** The Lucene results give the best answer ranked as number 10: The new system (Lucene+LDA) boosted the same answer to 5th rank. The scores of Lucene and LDA are shown at below:

> **5: doc: QAtestDocs/senDocLucene/QAtestDocs-BT-finance2&5.txt with lucene: 0.147376507520675 and LDA: 0.23581354260208 (Final Score: 0.38319005012276147)**
> **Content: Light sweet crude for October delivery rose 32 cents to settle at $70.03 a barrel on the New York Mercantile Exchange.**

The benefit of such rank boosting is that when the QA system selects the top N-ranked sentences as candidate answers the best answer is more likely to be included and shown the the final answer list.

5 Conclusions

In this paper, we investigated a methodology for using hierarchical probabilistic models for question answering. Topic models are used to re-rank the candidate answers from a standard keyword based search engine. Each candidate answer and the query is represented by the distributions over latent topics from offline training of topic models. Because there is only a short list of keywords for candidate answer sentences and query, their keywords were expanded by WordNet before the topic distribution calculation.

We compared the new system and the basic keyword search engine on a small test corpus. The ranking results of 90 sample questions are presented. The new systems performs better than the basic keyword matching for about 31% of the questions, for 66% of questions, the new system performs as good as the basic keyword matching. There is also about 3% of questions that the new system performs worse due to the noise of augmentation. By considering the semantic information, topic models can improve the ranking made by classical search engine. It also provides a new research direction of how to efficiently use probabilistic models to improve the answer ranking in question answering.

Acknowledgements

Qin and Huang are funded by the British Telecom (BT)/BISC Research Fellowship.

References

1. Blei, D.M., Griffiths, T., Jordan, M.I., Tenenbaum, J.: Hierarchical Topic Models and the Nested Chinese Restaurant Process. In: Thrun, S., Saul, L., Schoelkopf, B. (eds.) Advances in Neural Information Processing Systems (2004)
2. Blei, D.M., Lafferty, J.D.: Dynamic Topic Model. In: Proceedings of the 23rd ICML, Pittsburgh, USA (2006)
3. Blei, D.M., Lafferty, J.D.: Correlated Topic Models. In: Advances in Neural Information Processing Systems 18. MIT Press, Cambridge (2006)
4. Blei, D.M., Ng, A., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Leanring Research 3, 993–1022 (2003)
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science 41 (1990)
6. Gospodnetic, O., Hatcher, E.: Lucene in Action, Manning (2004)
7. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. Proceedings of the National Academy of Science 101, 5228–5235 (2004)
8. Miller, G.: WordNet: a lexical database. Communications of the ACM 38(11), 39–41 (1995)
9. Hofmann, T.: Probabilistic Latent Semantic Anlalysis. In: Proceedings of UAI 1999, Stockholm (1999)
10. Jordan, M.I.: Learning in Graphical Models. MIT Press, Cambridge (1999)
11. Beg, M.M.S., Thint, M., Qin, Z.: PNL-enhanced Restricted Domain Question Answering System. In: The Proceedings of IEEE-FUZZ, London, pp. 1277–1283. IEEE Press, Los Alamitos (1999)
12. Qin, Z., Thint, M., Beg, M.M.S.: Deduction Engine Designs for PNL-based Question Answering Systems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 253–262. Springer, Heidelberg (2007)
13. Steyvers, M., Griffiths, T.: Probabilistic topic models. In: Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.) Latent Semantic Analysis' A Road to Meaning (2007)
14. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: SIGIR 2006, Seattle, WA (2006)