Improving Social Bookmark Search Using Personalised Latent Variable Language Models

Morgan Harvey and Ian Ruthven
University of Strathclyde
Computer and Information Sciences Department
Glasgow, United Kingdom
{morgan,ir}@cis.strath.ac.uk

Mark J. Carman
University of Lugano
Faculty of Informatics
Lugano, Switzerland
mark.carman@lu.unisi.ch

ABSTRACT

Social tagging systems have recently become very popular as a method of categorising information online and have been used to annotate a wide range of different resources. In such systems users are free to choose whatever keywords or "tags" they wish to annotate each resource, resulting in a highly personalised, unrestricted vocabulary. While this freedom of choice has several notable advantages, it does come at the cost of making searching of these systems more difficult as the vocabulary problem introduced is more pronounced than in a normal information retrieval setting.

In this paper we propose to use hidden topic models as a principled way of reducing the dimensionality of this data to provide more accurate resource rankings with higher recall. We first describe Latent Dirichlet Allocation (LDA), a simple topic model and then introduce 2 extended models which can be used to personalise the results by including information about the user who made each annotation. We test these 3 models and compare them with 3 non-topic model baselines on a large data sample obtained from the Delicious social bookmarking site. Our evaluations show that our methods significantly outperform all of the baselines with the personalised models also improving significantly upon unpersonalised LDA.

Categories and Subject Descriptors

H.3.3 [**Information Storage & Retrieval**]: Information Search & Retrieval

General Terms

Machine Learning, Modelling, Experimentation

Keywords

Personalised Search, Topic Models, Collaborative Tagging, Social Bookmarking

1. INTRODUCTION

Social tagging systems provide a new way for Internet users to organise and share their own digital content and content from other users. Users are able to annotate each resource with any number of free-form tags of their own choosing without having to adhere to an *a-priori* set of keywords. The result of which is a personalised categorisation system defined by its users that can assist in locating resources in the future. Such systems have become extremely popular over the past few years and are used to annotate and categorise a large variety of different resource types [12]. Their simple nature and unrestricted vocabulary is a boon for annotators, however searching for resources of interest in social tagging systems tends to be a frustrating process.

Analyses of tagging systems [3] have shown that term use tends be very inconsistent between different users resulting in a large number of polysemous and synonymous tags. This has a highly detrimental effect on search performance unless the system deals with this inherent variation in some way. Several studies have shown that obtaining high consistency among different taggers is very difficult to achieve and can be affected by many factors including vocabulary use, personal understanding of the resource and language [21, 8].

These highly undesirable characteristics make searching or browsing through the collection difficult and generally less accurate. This problem is not restricted to the domain of social tagging and was identified early in the development of information retrieval systems [20], however due to their unrestricted vocabularies and inherent data sparsity it is a more common issue in social tagging systems. This issue is compounded by the fact that the vast majority of search queries are short (usually less than 3 terms in length) and are frequently ambiguous in nature [6].

In current social tagging systems, search algorithms tend to be rather simplistic in nature, often relying on simple term matching algorithms in order to rank resources given a query and seek to exploit the aggregated annotations across all users, the so called "wisdom of the crowds". This simple approach to the problem fails to deal with the vocabulary problems noted above and can result in quite poor rankings, particularly when users make use of very specific or unusual tags.

One potential method of reducing this ambiguity and thus improving search performance is to use some form of dimensionality reduction so that terms which frequently co-occur and are therefore likely to have a similar meaning, are in some way grouped together or implicitly linked. By doing so we can reduce the requirement on the user to choose exactly the same terms for a query as those used to annotate the relevant resources.

Consider a resource about a laptop computer which has been annotated by a knowledgeable user with the tags "macbook pro" and "core 2 duo". A less knowledgeable user may be searching for this resource and may not know the specific terminology and as a result will use simpler search terms such as "laptop" and "computer". Or, alternatively, the searcher may have a little knowledge of the terminology but misspells some of the query terms, for instance "macbookpro". In a search system with no dimensionality reduction the relevant result will be ranked very low as its annotations do not contain the exact terms of the user's search query. However a reduced dimensionality system does not rank resources based purely on matching terms, but does so by calculating a probability (or distance) of each resource given the query terms over the lower dimensional space. Since there is no requirement for the terms to match exactly and the system will have reduced all of these terms to the same dimension(s), it is highly likely that the relevant resource will be given a high rank for this query, thus allowing the user to fulfil their information need.

Another possible way of dealing with the inherent ambiguity of search queries is to attempt to personalise the search results based on the user's preferences or interest profile. In the case of social tagging data we can build such user profiles implicitly by looking at the resources the user has bookmarked and the tags they have used to annotate these bookmarks; the user's tagging history. Previous studies have suggested that while it can be difficult, if done correctly, personalisation can indeed improve the quality of search results [2].

A classic example where understanding the user's interests is of clear benefit is when the user enters a vague and highly ambiguous query. For example a user interested in astrology may want to find articles about the star sign *Cancer* and may simply choose to enter the query "cancer". It seems a reasonable assumption that such a query would provide good results, however the word *cancer* has another very different meaning. At the time of writing, entering such a query on the Google search engine returns absolutely no results pertaining to the astrological meaning of the word within the first page of results. However in a personalised system the user's preference for astrology would cause results relating to this topic to be pushed up the rankings, making it much more likely that the user will easily find a relevant result.

In this paper we investigate both of these possibilities by utilising techniques based on topic modelling to rank resources from a social bookmarking system given simple search queries. We first investigate related work in both social tagging and general information retrieval fields. Next we introduce topic modelling by describing the well-known Latent Dirichlet Allocation model [1, 4] and then go on to explain a logical extension to this model, allowing it to capture the notion of user interests [5] and propose an alternative to this model which has a more appealing generative story. We describe algorithms for ranking resources using these models and evaluate their performance based on a large sample of data from the social bookmarking site delicious and compare them with a number of competitive non-topic model baselines. Finally we conclude with a discussion of the results of the research and some suggestions for future work.

2. RELATED WORK

Previous attempts have been made to improve search performance in tagging systems, such as work by Hotho et. al. [9] which utilised graph theory techniques based on the famous PageRank algorithm to rank documents. The authors conclude that enhanced search facilities are vital to support emergent semantics in tagging systems and found that their algorithm was good at identifying latent communities of interest. The algorithm is therefore useful for recommending documents to users based on their topical interests but is perhaps not so suited for use as a search system.

[13] investigate the use of tags from Delicious as additional source of data to assist in automatic clustering of web pages. Their results show that principled inclusion of tagging data can improve model quality and aid in the clustering process. They use both k-means and topic modelling based approaches and find that the latter significantly improves on the former indicating that such models are a good fit for tagging data. This work provides an interesting insight into how our own models may perform however it differs significantly from this work as it does not attempt to rank resources solely on tagging data and does not attempt to personalise the results.

In more recent but similar work [17] the authors describe methods of deriving user profiles based on data obtained from social bookmarking systems to personalise search results on the Yahoo! search engine. However, again they do not attempt to apply this model to rank resources in the bookmarking system itself, they use it to *re-rank* the top URLs returned by the Yahoo! Boss API based on the user preferences obtained from delicious data. Their results and methods are therefore not comparable with those described in this paper.

Closer to the work described in this paper is [18] where the authors also attempt to provide personalised rankings using social tagging data. We discuss their models later on and use the best performing one (when applied to our data) as a highly competitive baseline. In this case the authors use Language Modelling techniques to estimate probabilities of resources given tags and tags given users. They use the resulting parameters to rank resources given single term queries and compare various smoothing methods for obtaining these estimates.

Other uses for personalisation in social tagging systems have been investigated and several papers have looked at providing personalised tag suggestions to users when annotating resources. This includes work by Sigurbjörnsson et. al. [14] and more recent work by the authors [5] in which we make use of hidden topic models to provide the suggested tags. We use this work as a starting point to build a new model presented later and also derive a personalised resource ranking algorithm for this model and use it in our experiments. Work by Krestel et. al. [11] also explored the use of topic models for tag recommendation and by extension to improve search results, however they did not make any attempt to personalise the recommendations.

Outside of social tagging, there have been a number of studies on the possibility of personalising search systems. For example Dou et al. [2] investigated a number of methods for creating user profiles and generating personalised rankings using query logs. Their approach was to use a set of pre-defined interest categories and a K-nearest neighbour approach for clustering similar users. In this work we take a similar view that by reducing the dimensionality of the data we can get better results, however we use more principled techniques that do not rely on predefined categories but derive these from the data as part of the estimation process. Other work [10] has used Singular Value Decomposition techniques to factorise data for personalisation of movie recommendations in the Netflix competition.

Teevan et. al. [15] investigated for what kinds of queries personalisation techniques most improved ranking performance. They found that how ambiguous a query is provides a good indication of how much benefit will be gained from personalisation. However for queries of low ambiguity (where all users tend to find the same results relevant) the personalisation can have a negative impact on performance. This work indicates that we must be careful when designing such systems to ensure that too much weight is not given to prior user preferences in deference to the unpersonalised document score.

We now describe topic models, explain why we have chosen them as a manner of factorising social tagging data and show how they can be used to rank resources. We go on to show how these models can be used to subtly include a user's preferences into the rankings. It is worth noting that while we making some reuse of the model presented at ECIR in this work we are using it for a completely different purpose and that significant modification was required to obtain improvements over the LDA baseline. Furthermore in this work we derive a second, more sound, generative model which gives significantly improved performance.

3. TOPIC MODELS

Topic models attempt to probabilistically uncover the underlying semantic structure of a collection of resources based on analysis of only the vocabulary words present in each resource, this latent structure is modelled over a number of topics which are assumed to be present in the collection.

In order to use these techniques we need to construct representations of documents made up of terms from a shared vocabulary. In this case our "documents" are the URLs (or in social tagging parlance *resources*) users have chosen to bookmark and we construct the documents representations by conflating the tags used by all users to annotate each bookmarked URL. Therefore each URL is now represented by the complete set of tags used to describe it by users of the social tagging system. Ideally this approach should allow us to: (1) generalise vocabulary terms to deal with synonymy and polysemy and (2) generalise the resource representations based on the similarity to other resources in the data set. These models operate using Bayesian inference which is useful when reasoning from noisy data, this is particularly appealing in this context as we expect the distributions of tags over resources to be both sparse and noisy.

In this section we briefly discuss Latent Dirichlet Allocation (LDA) [1, 4] which is a simple latent topic model that we use as a basis for ranking resources in the social tagging system. We then describe the Tagging Topic Model (TTM), a personalised model based on LDA.

3.1 Latent Dirichlet Allocation (LDA)

Figure 1 shows a graphical model diagram for LDA on the left. Notice that we are not using the "standard" diagram in which a second plate is drawn to represent all of the samples (words and topics) from the same document. Instead, and equivalently in terms of the generative process, we introduce an observed variable d_i denoting the corresponding document ID for each word w_i in the corpus. We use this notation to facilitate easier comparison between the LDA model and our new model introduced in the next section.

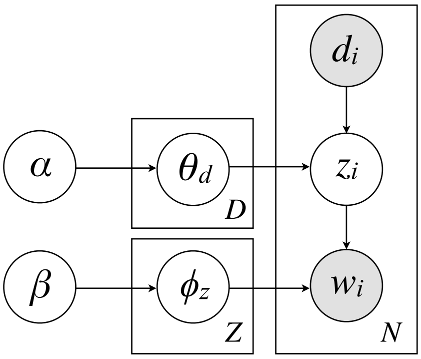

Figure 1: An alternate graphical model for Latent Dirichlet Allocation (LDA). N **is the number of word positions,** D **is the number of resources and** Z **is the number of topics.**

LDA represents documents as random mixtures over latent topics which are random mixtures over observed words in the vocabulary. The model possesses a number of advantageous attributes; it is fully generative meaning that it is easy to make inferences on new documents or terms and overcomes the overfitting problem present in models such as Probabilistic Latent Semantic Indexing (pLSI) [7]. Also since in LDA each document is a mixture over latent topics it is far more flexible than models that assume each document is only drawn from a single topic.

The parameters estimated in LDA are two matrices Φ and Θ containing estimates for the probability of a word given a topic $P(w|z)$ and a topic given a document $P(z|d)$. Thus each column of the respective matrices contains (estimates for) a probability distribution over words for a particular topic and over topics for a particular document, denoted ϕ_z and θ_d respectively. In order to prevent overfitting the data, LDA places a symmetric Dirichlet prior on both these distributions, resulting in the following expectations for the parameter values under the respective posterior distributions $P(\phi_z|\mathbf{w}, \mathbf{z})$ and $P(\theta_d|\mathbf{z}, \mathbf{d})$, where $\mathbf{w}$ is the vector of words occurrences w_i in the corpus, $\mathbf{z}$ is an assignment of topics to each word position z_i and $\mathbf{d}$ is the vector of documents d_i associated with each word position:

$$\hat{\phi}_{w|z} = \frac{N_{w,z} + \beta \frac{1}{W}}{N_z + \beta}$$

$$\hat{\theta}_{z|d} = \frac{N_{z,d} + \alpha \frac{1}{Z}}{N_d + \alpha}$$

Here $N_{w,z}$, $N_{z,d}$ and N_z are counts denoting the number of times the topic z appears (in $\mathbf{z}$) together with the word w, with the document d and in total. W is the vocabulary size and Z is the number of topics. Symmetric Dirichlet priors with hyperparameters α and β are placed over the distributions θ_d and ϕ_w and essentially act as a pseudo count indicating a relation to smoothing in language models. This allows the model to fall back on the priors in the event of sparse data.

Exact inference of the LDA model is intractable, however a number of methods of approximating the posterior distribution have been proposed including mean field variational inference [1] and Gibbs sampling [4]. Gibbs sampling is a Markov chain Monte Carlo method where a Markov chain is constructed that slowly converges to the target distribution of interest over a number of iterations. Each state of the Markov chain is (in this case) an assignment of a discrete topic (from 1 to Z) to each z_i, i.e. to each observed word in the corpus. In Gibbs sampling the next state in the chain is reached by sampling all variables from their distribution when conditioned on the current values of all the other variables.

The Gibbs sampling procedure for LDA involves iteratively updating the assignment of each topic z_i in the topic vector $\mathbf{z}$ by sampling a value from the distribution $P(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})$, which is conditioned on the current assignment to all topic variables except z_i. (The vector $\mathbf{z}_{-i}$ denotes all topic assignments except z_i.) In LDA the word assignment is conditionally independent of the document given the topic assignment:

$$P(z_i|w_i, d_i) \quad = \quad \frac{P(z_i, w_i|d_i)}{P(w_i|d_i)} \propto P(w_i|z_i)P(z_i|d_i)$$

Thus the expected value for the conditional distribution is simply:

$$\begin{aligned} \mathbf{E}[P(z|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})] \quad & \propto \quad \hat{\phi}_{w_i|z_i}\hat{\theta}_{z_i|d_i} \\ & \propto \quad \frac{N^{-i}_{w_i,z} + \beta\frac{1}{W}}{N^{-i}_{z} + \beta} \frac{N^{-i}_{d_i,z} + \alpha\frac{1}{Z}}{N^{-i}_{d_i} + \alpha} \end{aligned}$$

The estimates $\hat{\phi}_{w|z}$ and $\hat{\theta}_{z|d}$ are calculated over $\mathbf{z}_{-i}$ rather than $\mathbf{z}$. So $\mathbf{z}_{-i}$ denotes the assignment of topics to all word positions (except the current topic z_i). W is the vocabulary size. In the full derivation $N^{-i}_{w_i,z}$ is the number of times word w_i is assigned to topic z and N^{-i}_{z} is the total number of words assigned to topic z (both excluding the current position, z_i). $N^{-i}_{d_i,z}$ is the number of times topic z occurs in resource d_i (excluding z_i) and $N^{-i}_{d_i}$ is the total number of words in resource d_i (less 1).

After sufficient iterations of the sampler, the Markov chain converges and the parameters of the LDA model can then be estimated from $\mathbf{z}$. We can assume that the chain has converged when we observe minimal change in the model likelihood over successive samples, in the case of LDA the likelihood is:

$$P(\mathbf{w}, \mathbf{z}|\Phi, \Theta) = \prod_i \sum_z \hat{\phi}_{w_i|z}\hat{\theta}_{z|d_i}$$

For increased accuracy, we average parameter estimates over consecutive samples from the Markov chain. We can now use our estimated parameters Φ and Θ to compute a variety of useful distributions such as which documents are similar to each other, which words are similar to each other and by sampling over new data we can easily incorporate new documents into our model without having to re-run the entire algorithm.

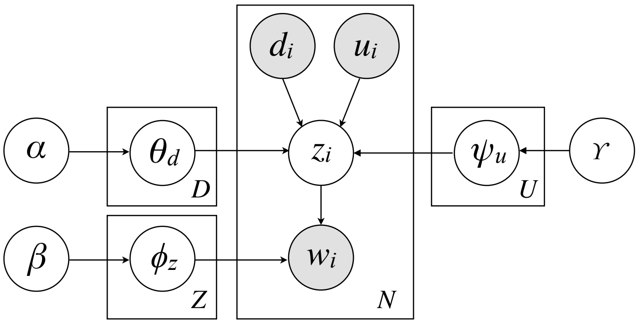

Figure 2: Tagging Topic Model 1 (TTM1) has 3 observed variables and one latent variable per word position. The difference with LDA being the addition of an observed user variable u_i, which like the word w_i is dependent on the topic z_i.

Having introduced standard LDA we now describe a model, influenced by LDA, which can capture the user's interests over topics therefore allowing for personalised modelling of the corpus. We also propose an alternative model (TTM2) which benefits from a much clearer and more appealing generative story and a simpler and more efficient ranking algorithm.

3.2 Tagging Topic Model 1 (TTM1)

In the LDA model we construct distributions of topics over resources, $\theta_{z|d} = P(z|d)$, which are easily interpretable as describing the topics that resource most likely pertains to. In social tagging systems we can also consider distributions over users, where the distribution indicates a user's topical preferences. Here we explain how tagging systems are structured, how that structure is normally modelled and then describe a different model which is suitable for capturing these user topical distributions.

Social systems typically consist of 3 distinct entities: the resource being tagged, the user who tagged the resource and the tag itself. In the literature, this is typically modelled as a tripartite graph [9] with 3 disjoint sets of nodes: resources $\mathcal{D} = \{\mathtt{d}_1, \ldots, \mathtt{d}_D\}$, users $\mathcal{U} = \{\mathtt{u}_1, \ldots, \mathtt{u}_U\}$ and tags $\mathcal{W} = \{\mathtt{w}_1, \ldots, \mathtt{w}_V\}$ [1]. In this graph the edges between these nodes represent the individual annotations; a user $\mathtt{u}$ annotating resource $\mathtt{d}$ with tag $\mathtt{t}$. Each assignment of a tag to a resource by a user - each edge - is denoted as the relation $\mathcal{Y}$ and is typically called a tag assignment (*tas* for short). Therefore the complete folksonomy is actually a quadruple $\mathcal{F} := (\mathcal{U}, \mathcal{W}, \mathcal{D}, \mathcal{Y})$. The resources are typically identifiers linking each unique resource id to a single web resource such as an image - as on Flickr - or a URL - as on social bookmarking sites such as delicious.

In attempting to modify LDA to include user preferences the first, most natural step to take is to change the Θ matrix from being the $P(z|d)$ to the $P(z|d, u)$; i.e. the probability of topic z given both resource d and user u. This new represen-

[1] Note that in order to remaining in keeping with the notation from topic modelling literature we use the character d to denote resources and that for all intents and purposes the words *documents* and *resources* are interchangeable.

tation of users and resources over topics is a large, extremely sparse, 3D tensor $\in \mathbb{N}^{\mathcal{D}\times\mathcal{U}\times\mathcal{Z}}$. The sheer size and inherent sparsity of this distribution presents significant problems, particularly in terms of memory capacity required to work with it, the increased danger of overfitting and the considerable amount of time required to fully sample the conditional distribution. Consider that for most combinations of users, resources and topics we will have no information to go on from the corpus and as such in the majority of cases the estimate will be reduced to the prior over the distribution.

A solution to this problem is to take the naive Bayes assumption that the probability of a user and a resource are independent given a topic allocation. The tensor is therefore split into a pair of 2 dimensional matrices Θ, representing the $P(z|d)$ - as in LDA - and Ψ, the $P(z|u)$. The new probability of a topic given a user u and resource d is now derived as:

$$\begin{aligned}
P(z|\theta_d, \psi_u) &= \frac{P(z)P(\theta_d, \psi_u|z)}{P(\theta_d, \psi_u)} = \frac{P(z)P(\theta_d|z)P(\psi_u|z)}{P(\theta_d, \psi_u)} \\
&= \frac{P(z)[\frac{P(\theta_d)P(z|\theta_d)}{P(z)}][\frac{P(\psi_u)P(z|\psi_u)}{P(z)}]}{P(\theta_d, \psi_u)} \\
&\propto \frac{P(z|\theta_d)P(z|\psi_u)}{P(z)}
\end{aligned}$$

In order to keep the model fully Bayesian we place a symmetric Dirichlet prior γ over the user-topic distributions ψ_u. This results in the following parameter estimation under the posterior distribution $P(\psi_u|\mathbf{z}, \mathbf{u})$:

$$\hat{\psi}_{z|u} = \frac{N_{z,u} + \gamma\frac{1}{Z}}{N_u + \gamma}$$

Where $N_{z,u}$ and N_u are counts of the number of times the topic assignment z appears in annotations made by user u and N_u is the total number of annotations made by u.
For the Gibbs sampling procedure, the probability of a topic assignment z at position i in this model is factorised as:

$$P(z_i|w_i, d_i) = \frac{P(z_i, w_i, u_i|d_i)}{P(w_i, u_i|d_i)} \propto P(w_i|z_i)\frac{P(z_i|d_i)P(z_i|u_i)}{P(z_i)}$$

Thus the expected value for the conditional distribution is now:

$$\mathbf{E}[P(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})] \propto \hat{\phi}_{w_i|z_i}\frac{\hat{\theta}_{z_i|d_i}\hat{\psi}_{z_i|u_i}}{\hat{P(z)}}$$

Where $\hat{P(z)}$ is simply estimated as N_z/N (less the current topic allocation z_i). As with LDA we iterate this sampling routine until the Markov chain converges and then the parameters of the model can be estimated from the topic assignments $\mathbf{z}$, averaged over consecutive samples. The resulting model of the complete folksonomy, shown in Figure 2, can then be used to uncover relationships between users, tags and resources and therefore make useful inferences about new data.

3.3 Tagging Topic Model 2 (TTM2)

In the previous section we described TTM1; a complete model of the tripartite data found in social tagging systems and suggested that it may be used to provide a personalised ranking of resources in the tagging system. However this model's generative story (i.e. how we imagine that the data were originally generated) is a little unclear and does not intuitively fit in with how we expect social annotations to be generated. In both LDA and TTM1 it is assumed that each document in the collection "chooses" its own topical distribution θ_d, leading to an assignment of word positions in the document to topics based on this distribution. In the case of TTM1 this is somehow also related to the user's topical distribution, however it is not clear exactly what this relationship may be.

This generative story fits in well in a normal information retrieval setting where we are indexing the actual content of documents. However with social tagging data we are not using the content of the documents as features but rather the words (tags) chosen to describe resources by users of the social bookmarking system. Therefore we propose an alternative model, shown in Figure 3, where the resource is chosen by the topic rather than the other way round. This model describes the following "generative story":

1. For each word position i, a topic allocation z_i is randomly chosen from user u's topical distribution $P(z|u)$

2. A relevant resource is drawn randomly from topic z_i's document distribution $P(d|z)$

3. finally, a tag w_i to describe the resource is drawn from topic z_i's tags distribution $P(w|z)$

The generative story for this model seems to be a better fit for annotations as the user initially chooses a topic (or topics) she is interested in and then based on those topics will find resources to bookmark and annotate. In this model Θ now contains probability estimates of the form $P(d|z)$ and each column θ_z is a probability distribution over resources (documents) for a particular topic. The expected value of these parameters under the posterior are calculated as follows:

$$\hat{\theta}_{d|z} = \frac{N_{z,d} + \alpha\frac{1}{D}}{N_z + \alpha}$$

Given our new parameterisation, the probability of a topic assignment z at position i in this model can be factorised much more cleanly as:

$$P(z_i|w_i, d_i) \quad = \quad \frac{P(z_i, w_i, u_i|d_i)}{P(w_i, u_i|d_i)} \propto P(w_i|z_i)P(d_i|z_i)P(z_i|u_i)$$

Finally the expected value for the conditional distribution is:

$$\mathbf{E}[P(z_i|\mathbf{w}, \mathbf{z}_{-i}, \mathbf{d})] \propto \hat{\phi}_{w_i|z_i}\hat{\theta}_{d_i|z_i}\hat{\psi}_{z_i|u_i}$$

For both of these models, the resulting reduced-dimensionality distributions over the complete folksonomy can then be used to uncover relationships between users, tags and resources and therefore make useful inferences about new data. In the next section we describe ranking algorithms for both of these tagging models and also for LDA.

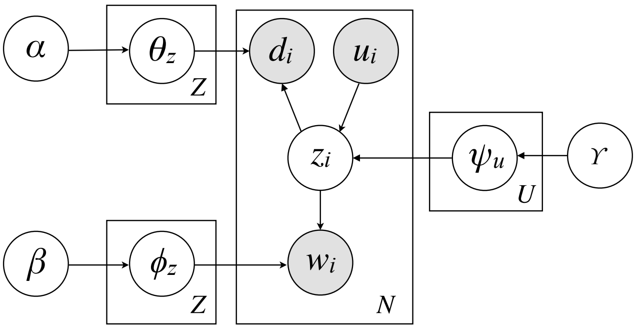

Figure 3: Tagging Topic Model 2 (TTM2) again has the addition of an extra observed user variable u_i. It differs from TTM1 by having a clearer generative process where the user would select the topic(s) they are interested in any then find resources based on those interests.

4. RANKING RESOURCES

We now describe formulas for ranking resources using the parameters that we have estimated in the topic models described above. Given a query q we wish to return to the user a ranked set of resources ($d \in D$) according to their likelihood given the query under the model, which in the case of LDA can be estimated as follows:

$$\begin{aligned} P(d|q) \propto P(d)P(q|d) &= P(d) \prod_{w \in q} P(w|d) \\ &= P(d) \prod_{w \in q} \sum_z P(w|z)P(z|d) \\ \textit{where} \quad & P(d) = N_d/N \end{aligned}$$

Notice that the ranking formula consists of the product of 2 distinct parts; a prior on the probability of the resource, $P(d)$, and the probability of the query given the resource, $P(q|d)$.

In the case of the TTM models we also know which user has issued the query and can therefore include that user's preferences into the ranking. We now rank documents according to their likelihood given both the query *and the user*, the ranking formula for **TTM1** is:

$$\begin{aligned} P(d|q,u) &\propto P(d|u)P(q|d,u) = P(d|u) \prod_{w \in q} P(w|d,u) \\ \textit{where} \quad P(d|u) &= P(d) \sum_z \frac{P(z|d)P(z|u)^{\pi_u}}{P(z)} \\ \textit{and} \quad P(w|d,u) &= \frac{\sum_z P(w|z)P(z|d)P(z|u)^{\pi_u} P(z)^{-1}}{\sum_z P(z|d)P(z|u)^{\pi_u} P(z)^{-1}} \end{aligned}$$

In the case of **TTM2** the $P(d|u)$ and $P(q|d,u)$ are as follows:

$$\begin{aligned} P(d|u) &= \sum_z P(d|z)P(z|u)^{\pi_u} \\ \textit{and} \quad P(w|d,u) &= \frac{\sum_z P(w|z)P(d|z)P(z|u)^{\pi_u}}{\sum_z P(d|z)P(z|u)^{\pi_u}} \end{aligned}$$

Again we can see that the formulas are the product of 2 parts: a *user-specific* document prior, $P(d|u)$, and the probability of the query given the resource and the user, $P(q|d,u)$. Notice also that we have introduced a weighting parameter, π_u in the range zero to one, on $P(z|u)$ so that we can vary the influence of the user's topical interests on the rankings. The intuition behind this being that resources likely tell us more about their own topic distribution than the users who annotated them.

In the next section we use a large sample of data obtained from the popular social bookmarking site delicious to evaluate the performance of these models.

5. EXPERIMENTS

We now discuss the experiments we performed on social bookmarking data comparing the LDA baseline model with our adapted Tagging Topic Models.

5.1 Preparing the datasets

In order to evaluate the relative performance of our models on real-world data we performed a crawl of the popular social bookmarking site delicious. To ensure a random sample of recent data we began by downloading the 100 most recent URLs submitted to delicious and recorded the usernames of the users who bookmarked them. We continued this process until we had collected a sample of 60,663 unique usernames. Then for each of these usernames we downloaded the respective user's 100 most recent bookmarks (as this is the largest number of recent bookmarks the delicious API will allow access to). Note that as 100 is the maximum number of bookmarks available via this crawling method per user not all users had this many bookmarks available resulting in 31% of the users having less than 100 bookmarks.

Each "document" (URL) is uniquely identified by computing a 32 bit MD5 hash of the complete URL, each URL and user in the data set was assigned a unique and anonymous ID number. To clean the resulting data set, we selected only the URLs which had been bookmarked by more than 2 unique users to ensure that all resources will always exist at least once in the training data. In order to give our systems reasonably complete user profiles to work from we selected only the users who had bookmarked more than 60 unique URLs from the remaining data after the first pass. Each remaining bookmark is a triple consisting of a URL identifier, a user identifier and a set of tags. We parsed the set of tags for each bookmark and finally removed all tags that appeared less than 2 times in the data set.

The original data set and the resulting reduced set is described in more detail in Table 5.1.

5.2 Evaluation methodology

We separated the dataset into training and testing subsets by retaining the last 10% of bookmarks by each user for testing. In doing so we ensure that the test data is distributed over users in the same way as the training data. In order to generate queries to input into our ranking algorithms we use the set of tags from each test set bookmark as a pseudo query. We now need some form of relevance judgement for each pseudo-query and since we know what resource was chosen for each bookmark we can classify a ranked resource as being relevant if it is the same resource the user actually bookmarked.

We have chosen to use this method as we are interested in

Metric	Original	Reduced
users	60,663	9,587
URLs	476,248	111,232
vocab count	113,428	14,023
bookmarks	3,235,299	569,117
word occurrences	12,294,136	2,473,738
avg bookmarks/user	53.3	59.4
avg bookmarks/URL	6.79	5.1
avg annotations/URL	25.8	22.2
avg annotations/bookmark	3.8	4.3

Table 1: Counts and statistics for the original dataset created from the delicious crawl (Original) and after reduction (Reduced).

personalised results, therefore only the user(s) who originally tagged the resource can really say whether it is truly relevant to them or not. We believe this will accurately reflect the performance of a live system and is likely to actually give a slight under-estimate of the true performance.

In order to evaluate ranking performance we calculated the success at rank k (S@k)[2] and the mean reciprocal rank (MRR). These 2 measures are briefly described below:

S@k - "success at rank k" the ratio of times where there was at least 1 relevant document (resource) in the first k returned.
$S@k = \frac{1}{|q|} \sum_{i}^{|q|} I(rank(d_i, q_i) \leq k)$

MRR = "mean reciprocal rank" the multiplicative inverse of the rank of the first relevant suggested resource, averaged over test resources.
$MRR = \frac{1}{|q|} \sum_{i}^{|q|} \frac{1}{rank(d_i, q_i)}$

Since we are primarily interested in how well these models rank URLs we report the S@k and MMR up to rank 10 as they are the most commonly reported in other literature since people tend to only pay attention to the first page of results in a ranked list.

5.3 Parameter settings and sampling

We experimented with a large range of parameter settings for both the number of topics in each model, (discussed further below), and the hyperparameter settings for each of the prior distributions. We set the concentration parameters α and β to be 25.0 and $0.1W$ respectively, which means the α setting is slightly lower than is common in the literature [4]. We found that a slightly smaller value provided better results, perhaps because the average length of a "document" (resource) in these systems is much less than in a more standard IR corpus. For both personalised models we also set γ to 25 and in the TTM2 model we set α to $0.1D$. None of the topic models were particularly sensitive to parameter values, provided we did not choose excessively low or high values, where we are applying almost no smoothing or in the other extreme; smoothing out the information from the data completely.

For sampling we use the Rao-Blackwellised Gibbs sampler [4]. For all models we sampled the chain for 300 iterations in total, as this appeared to consistently give good convergence in terms of model likelihood, and discarded the first 200 samples as chain "burn-in". The remaining 100 samples from the end of the chain were averaged over to obtain the final parameter values.

5.4 Baselines

In order to usefully evaluate the performance of the topic models we chose 3 different baselines; SMatch - which emulates the kind of simple matching formulas currently used when searching social tagging sites, Okapi BM25 - a popular and quite robust probabilistic retrieval framework and BayesLM - a competitive baseline Language Model with Bayesian smoothing. For each of the baselines we optimised any free parameters to ensure a fair and unbiased comparison with the topic models. Here we briefly describe the formulas for these models:

SMatch $score(d, q) = \sum_{w \in q} N_{w,d}$

BM25 $score(d, q) = \sum_{w \in q} IDF(w). \frac{N_{w,d}(k_1+1)}{N_{w,d}+k1(1-b+b\frac{|d|}{avgdl})}$
where $IDF(w) = \frac{N-N_w+0.5}{N_w+0.5}$, $|d|$ is the length of resource d and $avgdl$ is the average length of a resource over the whole training corpus. k_1 and b are free parameters which we optimised to 2.0 and 0.1 respectively.

BayesLM $P(d|q) = P(d) \prod_{w \in q} \frac{N_{w,d}+\mu(N_d/N)}{N_d+\mu}$
where μ is the Bayesian smoothing parameter which we optimised to 0.75.

Note that BayesLM is the same as the non-personalised model used by Wang et. al. [18] except that we have adapted it to deal with queries of lengths greater than one. We tried using their full personalised model as a baseline, but found that it performed extremely poorly. This is perhaps because we are using a much larger data set with a vocabulary 14 times larger than theirs. In this case their choice to use raw tags as user profiles (rather than reduced dimensionality features as in this paper) may have resulted in significant overfitting and poor generalisation. We therefore do not report results from their personalised model.

5.5 Sampling using the weighted user-topic distribution

As noted in the *Ranking Resources* section above our intuition is that while giving equal weight to both the resource and user distributions within the models may work well for tag suggestion, this approach may not work quite so well for ranking resources. In this case we would expect the resource to convey more information about itself than the users who are annotating it, therefore in our ranking formulas we introduced a weight, π_u, on the user distribution. However we are still making the assumption that both the resource and user distributions are equally important in the sampling. Unfortunately incorporating such a weight into the sampling by simply raising the user distribution to a power will not have the same effect as it does in the ranking formula. This is because, in our experiments, the Gibbs sampling routine still eventually tended towards the non-weighted full conditional distribution over successive iterations. Since we are always averaging over multiple samples from the full distribution it simply took slightly longer to converge.

[2] We note that since we only have one bookmarked URL per set of tags, precision at rank k (P@k) is equal to S@k/k and thus we do not report it separately.

	S@1	S@5	S@10	MRR@10
SMatch	0.0555	0.1372	0.1860	0.0900
BM25	0.1701	0.2975	0.3376	0.2238
BayesLM	0.1819	0.3299	0.3772	0.2440
LDA	0.1994	0.3397	0.3936	0.2579
TTM1	0.2030	0.3556∗	0.4158∗	0.2675∗
TTM2	**0.2137†**	**0.3559∗**	**0.4202∗**	**0.2743†**

Table 2: Ranking performance of all models on the test data set. The highest score for each metric is highlighted. ∗ indicates the result is significantly better than LDA ($p < 0.05$), † indicates the result is significantly better than TTM1 and LDA ($p < 0.05$).

Our solution to this problem is to only sample using the user distribution, ψ_u, on every *k'th* sample. By averaging over a large number of samples from the end of the chain this approximates a weight of $\frac{1}{k}$. In all of our experiments we set the parameter k to 5, resulting in an effective weighting of 0.2. We found this had very little impact on the convergence time of the chain and has the added benefit of slightly reducing the average computational complexity of the sampling.

6. RESULTS

Table 2 shows the results of the ranking experiments for all of the models, for all of the topic models we set the number of topics at 250. Between the more "conventional" ranking methods we see that the language model with Bayesian smoothing has the best overall performance and considering its relative simplicity, it performs very well. BM25 is clearly less suited to this kind of data than it is to more normal documents and the SMatch algorithm - unsurprisingly - returns particularly poor results.

Comparing the "conventional" models with the topic models results show that over all metrics the topic models perform significantly better than the baselines. This is in contrast to results from previous work into ranking using topic models [19] and perhaps highlights the difference between the "documents" constructed from social tagging data and much longer real-world documents more commonly discussed in IR literature. In the case of social tagging data, the topic model's generalisation of the data and ability to deal with some of the vocabulary problems noted earlier are much more beneficial than perhaps they are with more normal corpora.

In comparing the 3 topic models we see that both personalised models are able to outperform the unpersonalised LDA baseline. TTM1 outperforms LDA by a statistically significant margin on all but one of the metrics whereas TTM2 outperforms it significantly over all measures. Between the 2 personalised models we see that TTM2, with its clearer and more straightforward modelling assumptions and ranking formula, is able to outperform TTM1 over all measures (and as a result also significantly outperforms LDA). TTM2 is able to outperform TTM1 by a significant margin on both S@1 and MRR@10 which considering the task at hand (ranking of resources) are arguably the most important metrics. This is because a better Mean Reciprocal Rank indicates that the model is able to rank the relevant resources higher more often where the user is most likely to see and therefore click on them. This is confirmed by the significant improvement in S@1 score where TTM2 is more able to identify the relevant resource as being most likely given the user and query on the first attempt.

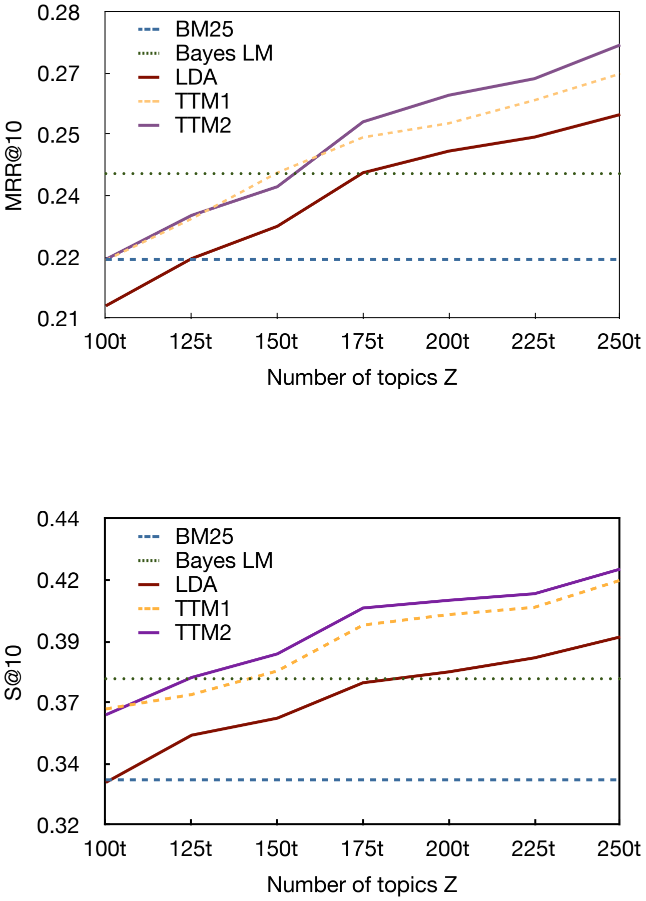

Figure 4: MRR@10 and S@10 over varying numbers of topics.

6.1 Varying the number of topics

When using hidden topic models an important consideration is how complex a model we should use in terms of the number of latent topics. We can in fact view each model (in this case we have 3; LDA, TTM1 and TTM2) as being a class of an infinite number of different models, where the complexity in number of topics is in the range $\{1, \ldots, \infty\}$. There has been a considerable amount of work published on so called non-parametric processes where the best model is inferred automatically based on the training data, the most appropriate for this work being Dirichlet Processes [16]. However these processes add significant further complexity and as such it is generally acceptable to use empirical methods to choose the most optimal parameterisation.

In this work we are not trying to optimise in terms of held-out likelihood but in terms of retrieval performance where these techniques may not be as appropriate. We would ex-

	S@10		MRR@10	
	0-60	60-80	0-60	60-80
SMatch	0.1707	0.1667	0.0815	0.0811
BM25	0.3232	0.3271	0.2098	0.2180
BayesLM	0.3624	0.3776	0.2344	0.2291
LDA	0.3694	0.3941	0.2212	0.2534*
TTM1	0.3705	0.4175*	0.2361	0.2700*
TTM2	0.3719	0.4454*	0.2394	0.2804*

Table 3: Ranking performance over user profile size. * indicates 60-80 bin significantly different from 0-60 bin ($p < 0.05$).

pect improvements in the held-out likelihood to taper off before improvements in retrieval performance do. Therefore we estimated parameters for the 3 topics models over different numbers of topics to see how retrieval performance was effected. Figure 4 shows the results for the metrics Success@10 and MRR@10 for the 3 topic models over the range of topics from 100 to 250 with increments of 25. We also show the results from the 2 most competitive non-topic model baselines to allow direct comparison, we omit SMatch from the figure as its performance is considerably worse than all the other models.

One can see quite clearly from the figure that as we increase the number of topics, the performance also increases. There appears to be a slight tailing off of performance improvement as the number of topics increases, however it is apparent that we could achieve even better ranking performance if we were to increase the number of topics even further. We chose to stop increasing the topic count at 250 due to time constraints and because by this point it was clear that the topic models were outperforming all of the baselines. There is no reason why in principle we couldn't keep increasing the topic count, however we would expect that at some point performance would peak and we would then be in danger of overfitting the model. Furthermore when using such systems a balance should be made between model complexity in terms of topics and ranking performance, since the amount of time required to rank resources using the models is linear in the number of topics.

Comparing between models, the data indicates that LDA needs approximately 175 topics before it begins to outperform BayesLM whereas the 2 personalised models only need somewhere between 125 and 150 topics, showing the advantage of incorporating the extra user data. The 2 personalised models have similar performance profiles over topics, however it appears that TTM2 begins to generally outperform TTM1 once it has enough topics to work with. This trend is particularly clear in the MRR figure where we can see that the 2 models only begin to diverge at around 175 topics and are fairly similar before this point.

6.2 Do we have enough data?

As we noted earlier in the paper, due to restrictions imposed by the delicious public API we were only able to collect a maximum of 100 bookmarks per user. Once we had removed all singleton resources and tags from the data set this left us with a fairly small profile for each user on which to build interest profiles over the topic space (an average of 59.4 bookmarks per user). We investigated how performance was impacted by the size of the user profiles by "binning" users based on the number of resources they had bookmarked in the training data into 2 bins. Table 3 shows the results of this analysis.

Clearly we can see that the non-topic model baselines do not benefit from having more information about the user, as you would expect. There is no significant difference in results between the 2 bins for SMatch, SM25 or BayesLM. In contrast, all of the topic models appear to show better performance when ranking resources for users with longer profiles. For LDA, the difference between the S@10 values for the 2 bins is not significant, however for the MRR@10 metric it is significantly different.

This effect is more pronounced in the personalised models, particularly TTM2 where the increase in both measures is very large when it has more information about the user. In fact the difference in performance between the 2 bins over both metrics for both personalised models is significant. This indicates that our models would perform even better if we had more information about our users, which would be the case were these techniques to be utilised on a live system. Note that we do not report results from the 80-100 resources bin as it only covers a very small percentage of the total users (103 out of 9587).

7. CONCLUSIONS

In this work we have discussed the problems facing ranking algorithms when dealing with social tagging data and proposed the use of hidden topic models to deal with its inherent sparsity and vocabulary ambiguity. We highlighted the 2 most prominent issues resulting from this kind of data and indicated how such models might be able to at least partially overcome these obstacles. Reference to related work shows that topic modelling has been successfully used in this area in the past, however it has not been used to provide personalised search results based purely on tagging data.

We first described the Latent Dirichlet Allocation model which serves as a starting point for the other topic models explored. We went on to discuss an extension designed to include user information into the model and suggested an entirely novel new model which we argued was cleaner and more parsimonious. We developed new resource ranking algorithms based on the parameters from these 3 models where the 2 personalised models also incorporate the user information into the rankings. Furthermore, we discuss a number of important subtle changes required to obtain useful improvements in ranking performance when using these models which would be necessary for successful implementation on a live system.

In order to test the relative performance of the models we proposed an evaluation framework utilising real data obtained by crawling the popular social bookmarking website Delicious and briefly described 3 non-topic model baselines including one previously used to research ranking in a social annotation setting. Finally we described and analysed the results of our experiments on the social tagging data and showed that our intuition of using topic modelling to overcome the vocabulary problems in tagging systems was appropriate.

The results showed that for social tagging data, the topic modelling approaches provided better resource rankings than even the most competitive baselines and outperformed them all by a statistically significant margin. They also demonstrated that our personalised models were able to effectively leverage the extra user information to present better rankings than the unpersonalised LDA model. Over all measures our proposed TTM2 model was able to significantly outperform LDA and was able to significantly outperform the less parsimonious TTM1 model on 2 key metrics. Further analysis of the results indicated that the performance of our personalised topic models could be improved further, relative to the other systems, if we had more data for each user.

In future work we would like to explore more complex models, perhaps where the resource-topic distributions are separate from user interest distributions or models where we incorporate more information including perhaps the actual content of the resources or by including temporal information in the model. We also wish to explore sampling and ranking methods that do not assume all queries are the same and instead adapt the rankings algorithms to better suit each individual query. For example if it was possible to identify how ambiguous a query was prior to ranking we may be able to determine how much weight to give to the user interest distribution.

8. REFERENCES

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, (3):993–1022, 2003.

[2] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 581–590, New York, NY, USA, 2007. ACM.

[3] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. In *Journal of Information Science*, volume 32, pages 198–208, 2005.

[4] T. Griffiths and M. Steyvers. Finding scientific topics. *National Academy of Science*, 2004.

[5] M. Harvey, M. Baillie, M. Carman, and I. Ruthven. Tripartite hidden topic models for personalised tag suggestion. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010*, 2010.

[6] B. He and I. Ounis. Query performance prediction. In *Information Systems*, volume 31, 7, pages 585 – 594, 2006.

[7] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[8] R. Hooper. Indexer consistency tests—origin, measurements, results and utilization. Technical report, IBM, Bethesda, 1965.

[9] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *Lecture Notes in Information Science*, 4011:411–426, 2006.

[10] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

[11] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 61–68, New York, NY, USA, 2009. ACM.

[12] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. 2004.

[13] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 54–63, New York, NY, USA, 2009. ACM.

[14] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, 2008.

[15] J. Teevan, S. T. Dumais, and D. J. Liebling. To personalize or not to personalize: Modeling queries with variation in user intent. In *SIGIR '08: Proceedings of the 30th international ACM SIGIR conference on Research and development in information retrieval*, 2008.

[16] Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical dirichlet processes. In *JASA 101(476)*, pages 1566–1581, 2006.

[17] D. Vallet, I. Cantador, and J. M. Jose. Personalizing web search with folksonomy-based user and document profiles. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010*, 2010.

[18] J. Wang, M. Clements, J. Yang, A. P. de Vries, and M. J. T. Reinders. Personalization of tagging systems. In *Information Processing and Management: an International Journal*, volume 460-1, pages 58–70, 2010.

[19] X. Wei and W. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval (SIGIR 2006)*, 2006.

[20] V. H. Yngve. The feasibility of machine searching of english texts. In *Proceedings of the International Conference on Scientific Information*, 1959.

[21] P. Zunde and M. E. Dexter. Indexing consistency and quality. *American Documentation*, 20(3):259–267, April 1969.