LDA for On-the-Fly Auto Tagging

Ernesto Diaz-Aviles, Mihai Georgescu, Avaré Stewart, Wolfgang Nejdl
L3S Research Center / Leibniz Universität , Germany.
{diaz, georgescu, stewart, nejdl}@L3S.de

ABSTRACT

In this paper, we propose a method for automatic tagging sparse and short textual resources. In the presence of a new resource, our method creates an ad hoc corpus of related resources, then applies Latent Dirichlet Allocation (LDA) to elicit latent topics for the resource and the associated corpus. This is done in order to automatically tag the resource based on the most likely tags derived from the latent topics identified. We evaluate our method, using an offline analysis on publicly available BibSonomy dataset and an online study, showing its effectiveness.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval–*Information Filtering*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing–*Language Models*

General Terms: Algorithms, Performance, Experimentation

Keywords: Automatic Annotation, Social Tagging, LDA, Recommender Systems, Web 2.0

1. INTRODUCTION

Tagging has proven to be an intuitive and flexible Web 2.0 mechanism to enhance the users's online experience. Tags are capable of facilitating search, easing navigation (e.g., tag clouds), improving personalization in collaborative tag recommendations and across disparate media types.

An important prerequisite for realizing the benefit of tags, is that a resource actually has at least one tag associated with it. When a resource has no associated tags or users, a collaborative tagging recommender can not provide a recommendation – the cold start problem.

One of the methods used to address the cold start problem is automatic tagging. State of the art work in this area relies upon latent data models to make explicit, some hidden, underlying "context".

We propose $\alpha-TaggingLDA$: an approach in which an untagged item is annotated by exploiting the content from similar resources found outside the boundaries of a single site.

RecSys2010, September 26–30, 2010, Barcelona, Spain.

2. PROBLEM DEFINITION

Consider a *folksonomy* as a four-tuple, $\mathbb{F} := (U, T, R, Y)$ [7], where:

- U, T and R are finite sets, whose elements are called users, tags and resources, respectively, and
- Y a ternary relation between them, i.e. $Y \subseteq U \times T \times R$, whose elements are called tag assignments.

The set of all tags that user $u \in U$ has assigned to resource $r \in R$ is defined as $T(u, r) := \{t \in T \mid (u, t, r) \in Y\}$ and the set of all *posts* of the folksonomy as $P := \{(u, T(u, r), r) \mid u \in U, r \in R, T(u, r) \neq \emptyset\}$.

The goal of *automatic tagging* consists of automatically annotating a given resource $r \in R$, with a set of tags $\widetilde{T}(SYS, r) \subseteq T$, where $SYS \in U$ is a special user representing the system.

3. THE PROPOSED LDA-BASED METHOD

3.1 Motivational Use-case

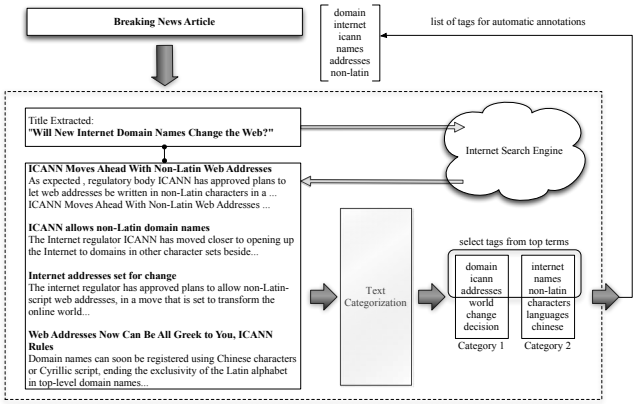


Figure 1: Web 2.0 Social Information System

Consider the scenario of a Web 2.0 Social Information System that analyses news headlines from RSS feeds and from micro blogging data streams. The system automatically annotates resources to improve navigation, searching, and serendipitous discovery of related resources.

As we see in Figure 1, the system is provided whith just the title of some breaking news article, or an entity(e.g, persons, product names, places, etc.) identified in a microblogging post or *tweet*[1]: *Will New Internet Domain Names Change*

[1]http://twitter.com/

the Web?. This is a *novel* resource, not present in the system's database and without tag annotations. Using an Internet search engine we retrieve the titles and snippets of similar resources, and together with the title of the resource, we build a corpus from which a LDA-based method will extract a list of tags to recommend.

3.2 Description of the Method

Latent Dirichlet Allocation (LDA) [2] is a generative probabilistic model for collections of discrete data such as text corpora. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over terms.

Each document is generated by picking a distribution over topics, and given this distribution, picking the topic of each specific term. Then, terms are generated given their topics. We consider that the documents correspond to resources $r \in R$. When a new resource without tags, needs to be annotated automatically by the system, i.e., by user $SYS \in U$, we can exploit r's textual content information or metadata to first, associate it to a collection of 'similar' resources, i.e., a specific corpus for the resource, and then discover the latent topics that generate them.

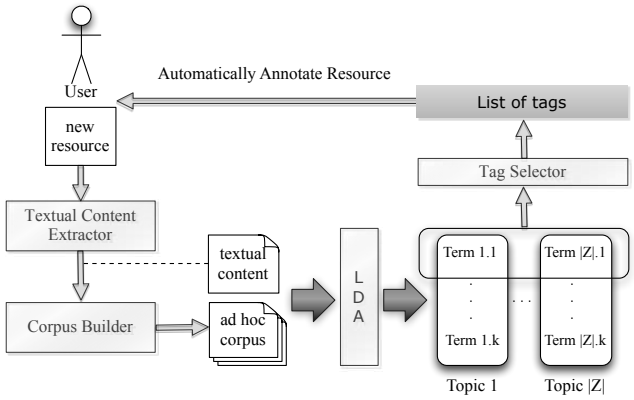


Figure 2: α-*TaggingLDA* general method

We use the probability distribution assigned by LDA on to the latent topics identified, which indicates the contribution of each topic to the overall collection, to guide our selection of tags from the most relevant topics. Furthermore, LDA also assigns a probability distribution over the terms within a latent topic. We exploit this distribution to rank the terms in a particular topic and use the top ones as candidate tags for annotations.

A LDA model is created *on-the-fly* for the resource and the associated similar documents, and it is discarded after the list of tag annotations is inferred.

3.3 Concrete Realization

The concrete realization of α-*TaggingLDA* used in our experiments is implemented in Java. The corpus builder (Figure 2) is based on the search results obtained by querying Yahoo!'s open search web services platform (BOSS)[2]. The titles and short text summaries (snippets) of the top-10 results returned are used to create ten different textual documents. The final *ad hoc* corpus for the resource consists of these and the textual content of the resource. Then, by applying LDA on this corpus we extract the desired number of latent topics, and from them, the needed tags are inferred.

We use the LDA with Gibbs sampling implementation provided by the Machine Learning for Language Toolkit (MALLET) [10].

[2]http://developer.yahoo.com/search/boss/

4. EVALUATION

To evaluate the efectiveness of our approach, the problem of automatic tagging is cast as a recommender system task.

4.1 Off-line evaluation

4.1.1 Datasets

We evaluate our α-*TaggingLDA* method on a BibSonomy dataset from [3]. This dataset is almost a complete dump of BibSonomy, i.e., all users, resources (publication references and bookmarks) and tags publicly available until December 31^{st}, 2008. All tags are lowercased and a cleansing process was applied to the data.

The characteristics of the dataset are:

$\lvert U\rvert$	$\lvert T\rvert$	$\lvert R\rvert$	$\lvert Y\rvert$	$\lvert P\rvert$
3,617	93,756	378,378	1,401,104	421,928

There are two kind of resources: *bookmarks* and *bibtex records*. To use as textual resources we extract the url and description available from bookmarks and the following fields from bibtex entries: author, editor, title, abstract, journal, booktitle, notes and description.

4.1.2 Baselines

We compare the performance of our method against two baselines. The first one ($baselineMP$), relies on the most specific tags of a resource. For a given user $u \in U$, a given resource $r \in R$, and some $n \in \mathbb{N}$ the top-n *most popular tags by resource* are given by:

$$\widetilde{T}(u,r) := \underset{t \in T}{\overset{n}{\operatorname{argmax}}}(|Y_{t,r}|)$$

where, $Y_{t,r} := Y \cap (U \times \{t\} \times \{r\})$, for $t \in T$ and $r \in R$, which corresponds to the frequency of tag assignments on r having the tag t. When resources have just a few number or zero tag assignments, we complement the set with the *most popular tags of the folksonomy*:

$$\widetilde{T}(u,r) := \underset{t \in T}{\overset{n}{\operatorname{argmax}}}(|Y_t|)$$

where, Y_t is the set of all tag assigments having tag $t \in T$, and is defined as $Y_t := Y \cap (U \times \{t\} \times R)$.

The second baseline corresponds to a LDA-based tag recommender introduced in [9] and evaluated on the same datasets and splits as ours in [8], the values of their evaluation on the corresponding test splits are reported here as baseline and are identified as $baselineLDA$.

4.1.3 Measures

For the evaluation, we used the test data splits provided also by [3]. For a given user $u \in U$ and a given resource $r \in R$, the test data consists of a set of posts without tag assignments, i.e., $P^{test} := \{(u,S,r) \mid u \in U, r \in R, S = \emptyset\}$. The system has to compute the set of tags for this posts $S = \widetilde{T}(u,r)$ to complete the tag assignments. Some statistics about the test data splits are presented as follows:

$\lvert U^{test}\rvert$	$\lvert R^{test}\rvert$	$\lvert R^{test} \setminus R\rvert$	$\lvert P^{test}\rvert$
1,591	43,002	39,070	43,002

Table 1: Precision, recall and f1m against baselines

#tags	baselineMP			baselineLDA			α − tagging		
	recall	precision	f1m	recall	precision	f1m	recall	precision	f1m
1	0.01075	0.03995	0.01694	0.04113	0.14797	0.06436	0.0654	0.2218	**0.1010**
2	0.01961	0.03674	0.02557	0.06876	0.12523	0.08878	0.1058	0.1837	**0.1342**
3	0.02756	0.03780	0.03188	0.08723	0.10738	0.09626	0.1395	0.1675	**0.1522**
4	0.03257	0.03616	0.03427	0.10196	0.09518	0.09845	0.1578	0.1459	**0.1516**
5	0.03528	0.03408	0.03467	0.11358	0.08630	0.09808	0.1768	0.1369	**0.1543**

In the evaluation, the list of recommedations consist of five different tags, i.e., $|\widetilde{T}(u,r)| = 5$. The number of iterations of the underlying LDA algorithm is set to 100.

As performance measures we use precision and recall and f1-measure (f1m) which are standard in such scenarios [4] for each post $(u, T(u,r), r)$ as defined above. We then average these values over all posts in the given set and compute the f1-measure:

$$precision(\widetilde{T}(u,r)) = |T(u,r) \cap \widetilde{T}(u,r)|/|\widetilde{T}(u,r)|$$

$$recall(\widetilde{T}(u,r)) = |T(u,r) \cap \widetilde{T}(u,r)|/|T(u,r)|$$

$$f1m = 2 * precision * recall/(precision + recall)$$

4.2 On-line Evaluation

We deployed our implementation as a recommender system on the BibSonomy recommedation framework according the guidelines described in [3, 6].

The online evaluation took place from July 27^{th}, 2009, until September 1^{st}, 2009. More than 200 users received recommendations. The recommendations consisted of a list of 5 tags. The number of posts for we delivered tag recommendations is 11,102. For the online evaluation, we set the LDA parameter to produce two *general* topics and fixed the number of iterations to 50.

5. RESULTS AND DISCUSSION

5.1 How many topics?

The behavior of our method varying the number of topics(|Z| = 2, 4, 8, 16, 32) is shown in Figure 3a. As can be seen in the figure, performance decreases with the LDA topic size. A solution with few topics typically will generally result in broad topics whereas a solution with too many topics will result in uninterpretable topics that pick out idiosyncratic tags. The results, on the datasets explored, suggest that such broad topics have higher chance to produce tags general enough to explain the limited document collection, leading to a higher recall and precision.

5.2 Offline Evaluation

The prediction quality of α-*TaggingLDA* with two general topics is clearly superior to the one of the baselines (Table 1, Figure 3b) achieving a f1m@5= 15.43% (i.e., f1m evaluating 5 tags) . Given the high number of unseen resources in this dataset, a solution based on relational information only, such as the most popular tags by resource, is expected not to perform well, in this case, the *baselineMP* just achieves a of f1m@5= 3.5%. Surprisingly, the LDA baseline method just achieves a f1m@5 of only 9.8%. This can be explained on how this method represents the resources of the system. Each resource is considered as a *bag of tags*, without exploiting any content feature. A LDA model is built using the whole corpus of available resources, i.e., bag of tags, in the training set, the model is then applied on test resources to infer the tag recommendations. For unseen resources, i.e., without tags, the method fails to produce a suitable representation and performs suboptimal in this, more realistic, sparse dataset.

5.3 Online Evaluation

In the online setting, the average time period the recommender needs for delivering a list of tag recommendations is 1630.58 milliseconds[3] The results obtained during the online evaluation are shown in Table 2:

Table 2: $\boldsymbol{\alpha}$-***TaggingLDA*** online performance

# tags	recall	precision	f1m
1	0.06875	0.22500	0.10532
2	0.10625	0.19000	0.13629
3	0.13125	0.18000	0.15181
4	0.15000	0.16500	0.15714
5	0.15469	0.15000	0.15231

6. CONCLUSION

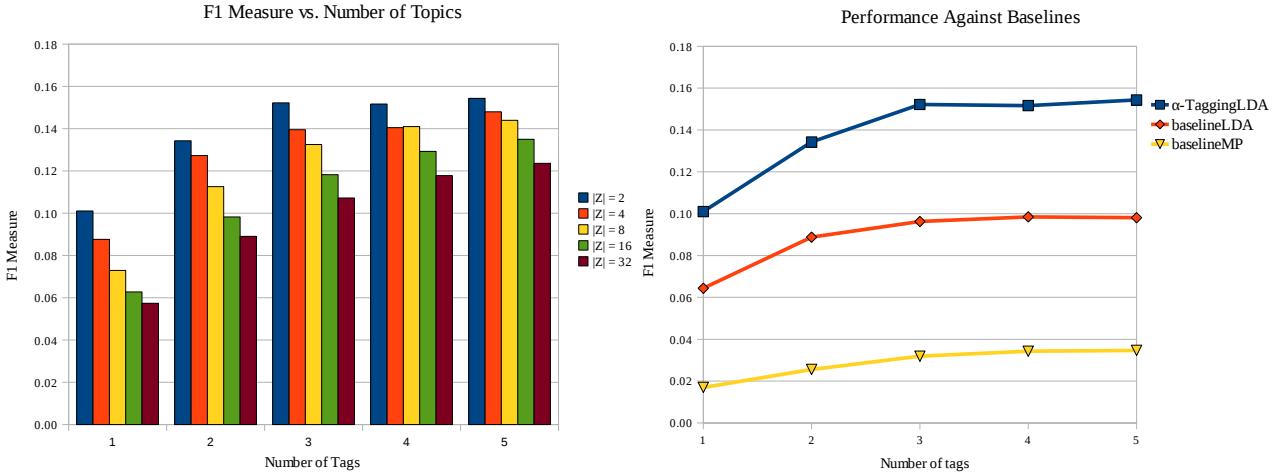
In this work, we present an approach to addressing the dynamics associated with online environments, where novel items appear rapidly, using probabilistic topic models, in specific, Latent Dirichlet Allocation. We show the ability of our method, α-*TaggingLDA*, to enrich sparse and limited textual information by means of exploiting the resource redundancy and latent topic overlap between similar resources found in an auxiliary domain.

We empirically evaluate, both offline and online, the effectiveness of our approach addressing the cold start problem on a collaborative tagging recommender scenario.

The online deployment of our method demostrates its efficiency and scalability delivering high quality recommendations in real time, without requiring any expensive offline model computation or updates. We believe that our approach would be ideally suited as part of a complementary solution for bootstrapping Web 2.0 social information systems.

As future work we intend to evaluate how the tags we suggest can help with regard to recommending resources to the users. We plan to evaluate our approach on items that are not textual like photos, video, music and other multimedia resources using the metadata. Early results in this direction suggest this to be challenging not only given the sparseness

[3]The results presented in this work ignore any possible timeouts in the process.



(a) F1 measure for different number of topics and tags

(b) F1 measure against baselines($|Z|$=2)

Figure 3: F1 measure

of such metadata, but the difficulty with which topics can be found, even after the enrichment from an auxillary domain.

7. RELATED WORK

Latent data models have been used to expose some hidden stucture or "context" to suggest tags for enhanced information access and collaborative tag recommendations. By context we refer to some meaningful aggregation of resources such as: association rules [5] or user/system defined clusters[1, 12]. In each of these cases, properties of aggregated resources increase overlap, which can be exploited to derive tag information about the resource on the Web.

Latent approaches treat the automatic suggestion of tags by relying upon dimensionality reduction: such as Latent Dirichet Allocation. In [9, 8] resources annotated by many users and thus having a relatively stable and complete tag set are exploited to overcome the cold start problem. They build an LDA model from tags which have been previously assigned by users. In this way, a resource in the system is represented with tags from topics discovered by LDA. For a new resource with few or no annotations, they expand the latent topic representation with the top tags of each latent topic. The work of [11] external knowledge from a large, so called "Universal Dataset" is used to address textual sparseness in the classification of short segments of text such as chat messages, or news feeds. They learn a LDA topic model from both a small set of labeled training data and the universal dataset. The model is then exploited to discover a set of latent topics which are subsequently used as the target in a multi-class classifier for the original sparse text.

In contrast to model based systems, instance based approaches do the association between users and annotations on-the-fly. For example, in Cross-Tagging [13], information accesses is enhanced for a non-folksonomy user, such as a music blogger, by exploiting the tag assertions made by (similar) users of folksonomies. The overlap between the mention of tracks in music a blog and the tracks in LastFM is determined. The user-resource-tag triples are modeled with a tensor; exploiting the underlying latent semantic structure in the tensor to form multi-way correlations between users, tags, and resources.

Acknowledgments: This work was funded in part by the Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, scholarship no. (E07D400591SV) and the the EU Project FP7 - 248984.

8. REFERENCES

[1] F. Abel, M. Frank, N. Henze, D. Krause, D. Plappert, and P. Siehndel. Groupme! - where semantic web meets web 2.0. In *ISWC/ASWC*, 2007.

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

[3] F. Eisterlehner, A. Hotho, and R. Jäschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09) http://www.kde.cs.uni-kassel.de/ws/dc09*, volume 497 of *CEUR-WS.org*, Sept. 2009.

[4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 2004.

[5] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *31st Annual International ACM SIGIR Conference (SIGIR'08)*, July 2008.

[6] R. Jäschke, F. Eisterlehner, A. Hotho, and G. Stumme. Testing and evaluating tag recommenders in a live system. In *RecSys '09*. ACM, 2009.

[7] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag Recommendations in Social Bookmarking Systems. *AI Commun.*, 21(4), 2008.

[8] R. Krestel and P. Fankhauser. Tag recommendation using probabilistic topic models. In *ECML/PKDD Discovery Challenge (DC'09)*, 2009.

[9] R. Krestel, P. Fankhauser, and W. Nejdl. Latent Dirichlet Allocation for Tag Recommendation. In *RecSys '09*. ACM, 2009.

[10] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[11] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW '08*. ACM, 2008.

[12] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles. Real-time automatic tag recommendation. In *SIGIR '08*. ACM, 2008.

[13] A. Stewart, E. Diaz-Aviles, W. Nejdl, L. B. Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Cross-tagging for personalized open social networking. In *HT '09*. ACM, 2009.