Personalized Parsimonious Language Models for User Modeling in Social Bookmaking Systems

Nawal Ould Amer, Philippe Mulhem, Mathias Géry

Personalized Parsimonious Language Models for User Modeling in Social Bookmaking Systems

Nawal Ould Amer[1,2], Philippe Mulhem[1], Mathias Gery[2]

[1] Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble France
Nawal.Ould-Amer@imag.fr, Philippe.Mulhem@imag.fr
[2] Univ Lyon, UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien UMR 5516, F-42023, Saint-Etienne, France
Mathias.Gery@univ-st-etienne.fr

Abstract. This paper focuses on building accurate profiles of users, based on bookmarking systems. To achieve this goal, we define personalized parsimonious language models that employ three main resources: the tags, the documents tagged by the user and word embeddings that handle general knowledge. Experiments completed on *Delicious* data show that our proposal outperforms state-of-the-art approaches and non-personalized parsimonious models.

Keywords: User profile, Parsimonious models, Words Embeddings

1 Introduction and Related Works

Personalized search systems (PSS) define and manipulate users' representations, or *profiles*, to enhance query results quality. We focus here on the use of social bookmarking systems, as they are important textual sources of evidence about users' interests.

Two major sources of user information are investigated by PSS works on social bookmarks: the *tags* assigned by a user to a particular document, and the *content* of the tagged document. This information is then exploited to construct a user profile. For example, [13] models a user over his/her tags, where each tag is weighted using *tf-idf* values. The authors of [4] weight user tags using *tf-iuf*, where [3] proposed a variant of *tf-iuf* (cf. section 3). Exploiting the content of the tagged documents (like web pages) is expected to broaden the profile vocabulary compared to tags. Indeed, previous studies on query log [11] have shown that document content is more useful. Most of works related to document content rely on *tf-idf* term weighting [5], or Latent Dirichlet Allocation (LDA) [6]. The main difficulty in modeling a user using his document content is to accurately filter the terms that come from the documents to keep only the important terms of the users' interests. This problem also occurs in relevance feedback models.

Parsimonious Language Models (PLM) [7] seek to build compact and precise term distributions by eliminating the stop words and nonessential terms. PLM was successfully applied for relevance feedback [8] to capture relevant terms from feedback document to expand a query. In this paper, we propose to adapt PLM to extract relevant terms from tagged document in order to model a user. To extract relevant terms, we use word embedding [1, 9]. This paper introduces *Personalized Tagged Parsimonious Language Models* (PTPLM) that capture an accurate term distribution to model a user using his

bookmarks. Our aim is to answer the following research questions: **RQ1:** Are user's tags effective to estimate important words of a user tagged document, then to model user's interests? **RQ2:** Are Personalized Tagged Parsimonious Language Models (PTPLM) able to improve state-of-the-art approaches? The paper is organized as follows. Section 2 details our PTPLM proposal. Section 3 presents the experiments conducted, and Section 4 is dedicated to the results and discussion. We conclude in Section 5.

2 Approach

2.1 PTPLM Estimation

In order to estimate personalized tagged parsimonious language models (PTPLM), we assume that each document d has a set of related tags which are assigned by a user u: $TG_u(d)$. Then, given d, its terms distribution θ_d and $TG_u(d)$, we re-estimate a new terms distribution for the document, noted θ_{d_u}. Let $d = \{t_1, t_2, ..., t_n\}$ the document tagged by a user u, and $TG_u(d) = \{tg_1, tg_2, ..., tg_P\}$ the set of tags given by u to d. We first estimate a document model as raw probabilistic estimation θ_d (i.e., first iteration in E-M algorithm) using maximum likelihood as follows: $P(t|\theta_d) = \frac{tf(t,d)}{|d|}$, where $tf(t, d)$ is the frequency of term t in d, of length $|d|$.

Now, taking inspiration from [7], we re-estimate the terms distribution by integrating the tags in the $E-Step$, where the terms related to the user tags should be important terms. In other words, if the term t (from the vocabulary V) in the document d is *related* to the tag tg used by a user u for the document d (using $P(t|\theta_{TG_{u_d}})$), then the term t is an important term.

$$E - Step : e_t = tf(t, d) \times P(t|\theta_{TG_{u_d}}) \times \frac{\lambda P(t|\theta_d)}{\lambda P(t|\theta_d) + (1 - \lambda)P(t|\theta_C)} \quad (1)$$

$$M - Step : P(t|\theta_d) = \frac{e_t}{\sum_{t \in V} P(t|\theta_d)} \quad (2)$$

where $P(t|\theta_{TG_{d_u}})$ is estimated as follows:

$$P(t|\theta_{TG_{u_d}}) = \frac{1}{|TG_u(d)|} \sum_{tg \in TG_u(d)} P(t|tg) \quad (3)$$

where $P(t|tg)$ is the probability of term t given a user tag tg, and $P(t|tg)$ is the probability that a term t is related to the tag tg, estimated using the cosine similarity between the two embedded vectors corresponding to term t and tag tg as follow: $P(t|tg) = sim_{cos}(t, tg)$. The iteration is repeated until the estimates do not change significantly anymore. Then we obtain a new term distribution θ_d that we rename θ_{d_u}. This is a personalized document terms distribution.

2.2 Building Users' Profiles

Let $D_u = \{d_1, d_2, ..., d_N\}$ the set of documents tagged by a user u. After PTPLM estimation for each document d in D_u as described above, a document user profile θ_u is defined, as presented in the algorithm 1. This algorithm builds the term user profile by averaging the term probabilities over the documents tagged by u (cf. line 6).

Algorithm 1 Estimation of User Model

Require:

$D_u = \{d_1, d_2, .., d_N\}$: Set of document tagged by a user u.

$TG_u(d) = \{tg_1, tg_2, .., tg_P\}$ Set of tags assigned to document d by a user u.

Ensure:

θ_u: User Model.

1: **for each** $d \in D_u$ **do**
2: $\quad \theta_{d_u} \leftarrow PTPLM(d, TG_u(d))$
3: **end for**
4: **for each** $t \in V$ **do**
5: $\quad$ **for each** $d \in D_u$ **do**
6: $\quad\quad P(t|\theta_u) = \frac{1}{|D_u|} \sum_{d \in D_u} P(t|\theta_{d_u})$
7: $\quad$ **end for**
8: **end for**

2.3 Ranking Model

To rank the documents, we use a query expansion model. we first select the terms related to the query (i.e. terms that are in the same context than the user query) from the user profile using the cosine similarities between q and t. Then, we expand the query using these terms with their weights $P(t|\theta_u)$. The ranking model is as follows: $RSV(q, d, u) = \alpha.RSV(q'_u, d) + (1-\alpha).RSV(q'_u, TG(d))$, where q'_u is the expanded query of a user u, $TG(d)$ is the set of tags assigned to the document d by all users, and α is a parameter in [0, 1].

3 Experiments

Dataset: We evaluate our proposal on the *Delicious* dataset [12]. We first perform a crawl of the English available web pages. For our experiment, we select only users with more than 100 unique tags for more than 100 unique bookmarks The resulting corpus contains 1,238,443 Web pages, 287,969 users and 204,505 unique tags.

Word Embeddings Train: We train a Continuous Bag-of-Word (CBOW) model [9] on Wikipedia corpus consisting of 20,151,102 documents and a vocabulary size of 2,451,307 words. The training parameters are set as follows: the output vectors size is set to 50, the width of the word-context window is set to 8, and the number of negative samples is set to 25.

Evaluation Methodology and Metrics: We use the evaluation framework for personalized search based on social annotation introduced by [2] and used in most of the state-of-the-art works [3, 6, 13]. This framework assumes that "*The users' bookmarking and tagging actions reflect their personal relevance judgment*". Then, the tags are considered as queries. A document is assumed relevant for a tag t considered as a query q issued by a user u, if the document has been tagged by u with the tag t [2]. We split the dataset into training and testing subsets: the last 20% bookmarks (according to the timeline) for each user are for testing, where the first 80% bookmarks are used for learning the profiles. We generate 4,911 queries for 128 users and their relevance judgments. We use the Mean Average Precision (MAP), and P@5 as evaluation metrics.

Parameters Settings: We used the Terrier Information Retrieval framework to compute the matching. We choose to use BM25 [10] weighting model with its default parameters. For the PTPLM approach, we tested the different values of λ in equation (1). The retrieval performances are stable over its different values, we fix here $\lambda = 0.5$. In the M-step of PTPLM, the terms that receive a probability below a fixed threshold (i.e. 0.0001, as in [7]) are removed from the model. In equation 3, for the estimation of $P(t|tg)$ using $sim_{cos}(t, tg)$, we consider only positive values of similarity.

Baselines: We compare our proposal to three personalization state-of-the-art approaches: (**Xu**) where the weights of users' tags are based on TF-IDF values [13]; (**Cai**) where the weights of users' tags are based on user term frequency computed as follow: $w_t = \frac{TF(t)}{D_u}$, where D_u is the number of document tagged by a user [4]; (**Bouadjenek**) where the weights of users' tags are based on user terms frequency computed as follow: $w_t = TF(t) \times \log(\frac{|U|}{|U_t|})$, where U is number of users and $|U_t|$ is the number of users who used t [3]. We also consider classical (non-personalized) (**PLM**), as well as non-expanded queries (**Noexp**).

4 Results

4.1 Impact of User Tags on Parsimonious Language Models

To explore our first research question RQ1, Table 1 shows the estimation of top-5 terms distribution for the same document[3], assuming that the tags assigned by a user to the document are: *casino*, *games*, and *DangerouslyFun*. The distribution estimated using a standard language model is presented in column *Standard LM*. The *PLM* column displays the terms distribution using classical PLM [7], with the final probability for each term averaged over the user's document. The *PTPLM* column presents the distribution estimated as in section 2.

Table 1. Term distribution for one document

Standard LM		PLM		PTPLM	
will	0.0689	online	0.0777	casino	0.1989
online	0.0583	casino	0.0670	players	0.1277
players	0.0477	players	0.0555	games	0.0978
casino	0.0397	games	0.0420	casinos	0.0677
can	0.0371	casinos	0.0346	gaming	0.0347

As seen in Table 1, the PTPLM re-estimate the term probability according to the user tags: the model emphasizes the terms related to the user tags. For example, the probability of the term *casino* is boosted compared to other models. This example shows that PTPLM is able to capture more accurately the personalized view of documents.

4.2 Evaluating User Profile Model: Comparison with baselines

To answer our second research question RQ2, we compare the results of our proposed model PTPLM with those of the baseline and state-of-the-art user model approaches

[3] URL document: http://www.dangerouslyfun.com

described in section 3. We aim to assess the quality of the profiles, then we consider only *single-term expansions*, and we apply several cut-off points for the profiles (100, 200, 300, and 500 terms) according to the term weights. We tested all approaches over $\alpha \in [0, 1]$, and we report the best configuration for each model in Table 2.

We see that PTPLM outperforms all state-of-the-art personalization models in term of MAP and P@5 for all user profile sizes. This shows that PTPLM is able to estimate a better terms distribution to describe user interests. The larger differences (in %) are obtained when keeping the top-100 terms from the profiles: this shows that our proposal is able to bring out important (relevant) terms more accurately than other approaches.

Table 2. One term expansion. Bold value: best query expansion system; **(x%$^{\triangledown}$)**: significant MAP differences w.r.t. PTPLM, bilateral paired Student t-test, p<0.05

Model	MAP				P@5			
Noexp	0.195				0.097			
Profile cutoff	100		200		300		500	
Models	MAP	P@5	MAP	P@5	MAP	P@5	MAP	P@5
PPLM	0.120 **(-34%$^{\triangledown}$)**	0.058	0.157 **(-16%$^{\triangledown}$)**	0.079	0.180 **(-6%$^{\triangledown}$)**	0.090	0.184 **(-5%$^{\triangledown}$)**	0.090
PTPLM	**0.181**	**0.092**	**0.188**	**0.094**	**0.192**	**0.096**	**0.194**	**0.097**
Bouadjenek	0.161 **(-10%$^{\triangledown}$)**	0.081	0.177 **(-6%$^{\triangledown}$)**	0.089	0.188 (-2%)	0.094	0.192 (-1%)	0.096
Cai	0.166 **(-8%$^{\triangledown}$)**	0.084	0.178 **(-6%$^{\triangledown}$)**	0.090	0.188 (-2%)	0.094	0.193 (-1%)	0.096
Xu	0.165 **(-8%$^{\triangledown}$)**	0.083	0.178 **(-6%$^{\triangledown}$)**	0.089	0.187 (-2%)	0.093	0.192 (-1%)	0.096

In Table 2 the *Noexp* runs are presented for the sake of completeness: we did not expect these runs to be outperformed by the naive and limited single-term expansions tested. Although, in a way to provide a fair framework when comparing Noexp and PTPLM results, we need to consider profiles that are potentially able to cover the many facets of the users' profiles. The Figure 1 presents the 68 larger query-by-query AP differences, among the full set of queries, comparing the top-500 terms profiles from PTPLM and the Noexp results.

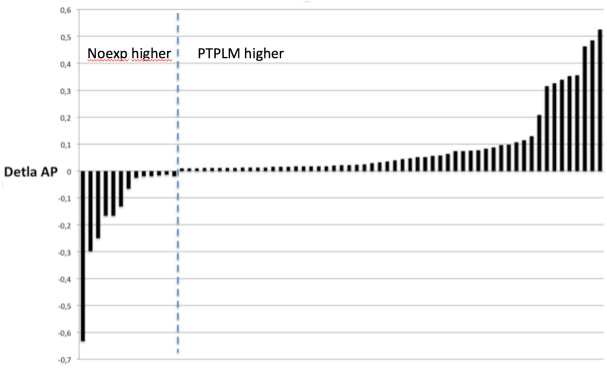

Fig. 1. Delta of AP values PTPLM top-500 profiles w.r.t. Noexp.

Our PTPLM proposal underperforms for 13 queries and outperforms for 55 queries the Noexp approach. So, even limited PTPLM-based expansions are able to play a positive role in many cases. We strongly believe that a more accurate usage of users' profiles will outperform the Noexp runs in the future.

5 Conclusion and Future Works

In this paper, we introduced the PTPLM approach, that exploits user tags to extract relevant terms from user tagged documents, expecting to obtain a better representation of user interests. According to our experiments conducted on *Delicious*, we found that PTPLM outperforms all state-of-the-art user modeling approaches. The PTPLM do not currently take benefit of user tags: we believe to gain effectiveness when using these tags. Our usage of profiles generated by PTPLM underperforms no-expansions runs. However, our analysis conducted query by query indicates that there is a great room for improving our usage of the generated profiles in the future. As future works, we are working on efficiently using our model to improve query expansion, and also on comparing our PTPLM with content based state-of-the-art approaches (e.g.: LDA).

Acknowledgements. This work is supported by the ReSPIr project of the région Auvergne Rhône-Alpes.

References

1. ALMasri, M., Berrut, C., Chevallet, J.P.: A Comparison of Deep Learning Based Query Expansion with Pseudo-Relevance Feedback and Mutual Information, pp. 709–715 (2016)
2. Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing web search using social annotations. In: WWW '07. pp. 501–510 (2007)
3. Bouadjenek, M.R., Hacid, H., Bouzeghoub, M.: Sopra: A new social personalized ranking function for improving web search. In: ACM SIGIR '13. pp. 861–864 (2013)
4. Cai, Y., Li, Q., Xie, H., Yu, L.: Personalized resource search by tag-based user profile and resource profile. In: WISE 2010. pp. 510–523 (2010)
5. Carman, M.J., Baillie, M., Crestani, F.: Tag data and personalized information retrieval. In: Proceedings of the ACM Workshop on Search in Social Media. pp. 27–34. SSM '08 (2008)
6. Harvey, M., Ruthven, I., Carman, M.J.: Improving social bookmark search using personalised latent variable language models. In: ACM WSDM '11. pp. 485–494 (2011)
7. Hiemstra, D., Robertson, S., Zaragoza, H.: Parsimonious language models for information retrieval. In: SIGIR '04. pp. 178–185 (2004)
8. Kaptein, R., Kamps, J., Hiemstra, D.D.: The impact of positive, negative and topical relevance feedback. In: 17th Text REtrieval Conference, TREC 2008 (2008)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
10. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., M.Gatford: Okapi at trec–3. In: Overview of the Third Text REtrieval Conference. p. 109–126 (1995)
11. Shen, X., Tan, B., Zhai, C.: Context-sensitive information retrieval using implicit feedback. In: ACM SIGIR '05. pp. 43–50 (2005)
12. Wetzker, R., Zimmermann, C., Bauckhage, C.: Analyzing Social Bookmarking Systems: A del.icio.us Cookbook. In: Proceedings of the ECAI 2008 Mining Social Data Workshop
13. Xu, S., Bao, S., Fei, B., Su, Z., Yu, Y.: Exploring folksonomy for personalized search. In: ACM SIGIR '08. pp. 155–162 (2008)