I Pinned It. Where Can I Buy One Like It? Automatically Linking Pinterest Pins to Online Webshops

Susana Zoghbi
KU Leuven
Celestijnenlaan 200A
Leuven, Belgium
susana.zoghbi @
cs.kuleuven.be

Ivan Vulić
KU Leuven
Celestijnenlaan 200A
Leuven, Belgium
ivan.vulic @
cs.kuleuven.be

Marie-Francine Moens
KU Leuven
Celestijnenlaan 200A
Leuven, Belgium
sien.moens @
cs.kuleuven.be

ABSTRACT

The information that users of social network sites post often points towards their interests and hobbies. It can be used to recommend relevant products to users. In this paper we implement and evaluate several information retrieval models for linking the texts of pins of Pinterest to webpages of Amazon, and ranking the pages (which we call webshops) according to the personal interest of the pinner. The results show that models that combine latent concepts composed of related terms with single words yield the best performance.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval-Information filtering

Keywords

Topic models, user interest, recommendation systems, personalized linking

1. INTRODUCTION

`Pinterest.com` is a social network that allows users to post and organize (or, simply, to *pin*) items (e.g., images, videos and text) found on the Web. For each post (known as a *pin*), a user often writes some text to describe or express her opinion about the pinned item. Pins often present items or activities users are interested in. They are categorized in *boards*, which may include fashion, travel, cars, food, film, humor, home design, sports, and art, among others.

Recommending relevant items to Pinterest users is interesting for retailers and online webshops. Pinterest itself already performs automatic recommendations of already known pins from other users. For each pin there is a section of "People who pinned this also pinned". In this case, similar known pins are recommended to the user based on the activity of other users that are interested in similar items.

However, for retail applications simply recommending other pins is not enough. While there are pins directly linked to online stores (i.e., Pinterest users sometimes post a link to a retailer or a webshop where the pinned item may be bought), such as `Amazon.com`, `Etsy.com`, `eBay.com`, etc, not all pins provide URL-s that link to online webshops where the pinned items are available for purchase. A Pinterest user might post an item that she would like to buy, but may not know where to buy it. In this case, it is useful to have a system that can automatically recognize the content of the pin and suggest online stores where the item (or similar ones) can be bought. Similarly, an online store might wish to find people interested in products resembling the ones in the store. For instance, if a user has several pins that contain "Louis Vuitton" bags, a related online store might benefit from that information.

The goal of this work is to spark interest in investigating techniques to automatically recommending online webshops to Pinterest users. In this initial stage, we focus on a setting that relies only on the text -disregarding the images and videos- from both pins and product descriptions of online webshops. We investigate whether the textual information available from a single pin is sufficient, and to what extent it helps to find relevant webshops from a variety of possible target webshops. A single pin may already contain modeling information about the user's interests. It provides a small snippet of possible life styles, likes, hobbies, etc. This minimalist approach that deals with unstructured user-generated data allows us to make inferences about the user in the absence of other elements commonly used in recommendation systems, such as known like-minded users.

In the absence of any other information, the task in this setting is naturally framed as an ad-hoc information retrieval (IR) task: Given a single pin (a *query*), the task is to rank a set of items that form a webshop (*document*) according to their relevance to the query. In this paper, we introduce the task along with our data collections acquired from the Web, and report the initial results obtained by a variety of ad-hoc IR models.

2. RELATED WORK

Recommender systems suggest interesting objects to users in a personalized way from a large space of possible options. For example, at Amazon recommendation algorithms personalize the online store for each customer by showing programming titles to a software engineer and baby toys to a new mother [5].

Many of the recommendation systems like Amazon's item-

DUBMOD'13, October 28, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2513577.2513581

to-item collaborative filtering rely on the items in the customer's cart, where an item is a well-defined object. Similar items are found as items that customers often bought together. Amazon then leads customers to an area where they can filter their recommendations by product line and subject area, rate the recommended products, rate their previous purchases, and see why items are recommended. In this work in order to personalize the recommendation, we work with *unstructured* textual data as found on social network sites such as Pinterest, and we completely automatically link a user's post, in our case a pin, to a relevant webshop.

From the early days of the Internet one has dreamed to automatically generate hyperlinks [6], but their automatic creation remains an understudied and difficult problem. Although recommendation techniques recently were inspired by information retrieval models [3, 1], in this paper automated hyperlinking is evaluated as a retrieval problem, where relevant webshops are ranked according to the personal interest of the user.

3. PROBLEM FORMULATION AND DATA

Problem Formulation. Given an information unit (e.g., a single post or pin) from a user, we wish to retrieve relevant online retail shops (webshops) where users could potentially buy items related to their interests. The problem may also be observed as a task of *linking* the online webshops to the items that the user pinned. Formally, let $\mathcal{D} = \{D_1, D_2, ..., D_L\}$ be a target collection of L webshops and Q is a textual content of a pin, that is, a query given by the set of m words in the pin/post $Q = \{q_1, q_2, ..., q_m\}$. The task is to rank the webshops according to their relevance to the pin. To study this, we have collected two datasets: a collection of Pinterest pins (Dataset I), and a collection of Amazon products (Dataset II). The latter serves to emulate online webshops.

Dataset I: Pinterest Data. We implemented a crawler to find Pinterest users, their boards and pins. A board is a collection of pins and a user page contains a set of boards. Boards are often classified by the user into broad categories such as fashion, travel, cars, food, film, sports, art, etc.

Our crawler performed a depth-first search starting from a popular (many followers) Pinterest user. To date we have collected over one million pins, corresponding to over 18,000 boards and 650 users. The number of pins in a board varies from a couple to several thousands. For our sample dataset, the average number of pins per user is 2,476, while the average number of pins per board is 55.6. Although studies indicate that females comprise the majority user population on Pinterest [7], our collection of over a million pins provide a wide variety of items spanning categories in fashion, technology, sports, books, decoration, jewelry, watches, among others.

In this work we do not exploit the users' histories and their overall profile info on Pinterest and leave that for future work, as we rather focus on a task of linking isolated single pins (currently the textual posts of the pin) to relevant webshops.

Dataset II: Amazon Data. We formed webshop documents using Amazon's product categories (or browse nodes). Amazon organizes its items hierarchically. The different levels of the node tree provide an organization principle used to catalog items. The nodes progress from general to specific. For example, a top level browse node might be "Shoes." It's child nodes might be "Men's Shoes," "Women's Shoes," and "Children's Shoes." Navigating down the tree refines the items from the general to the specific. Each node is a collection of related items, i.e., products that belong to the same category. We used leaf nodes to cluster groups of related products. We call these product clusters "webshops".

We implemented an XML parser to download information from over 23,000 products. These products were grouped into 1,171 webshops, where each webshop contains 20 products approximately. We focused on a set of top categories: apparel, beauty, books, groceries, jewelry, shoes, kitchen, music, electronics, sporting goods and watches. We started by querying the top categories and gathered the hierarchy of related child nodes. To represent the product, we used all the text associated to the product's description and editorial review. The idea is to simulate an online retail business that has a set of pages (webshops) containing related products. Our application aims to direct users in a social media site to such target webshops.

We chose Amazon because of the large and varied collection of available products and the ability to automatically download product information through their Product API. Nevertheless, the proposed IR framework may be extended to any other online webshop for which a textual representation is provided.

4. METHODOLOGY: HOW TO LINK IT?

In the task of linking relevant webshops to users' pins, we utilize different *text representations* and *retrieval models*. We investigate the impact of the different representations and models on the quality of linking, and we also explore whether combining different representations can boost the linking performance.

4.1 Bag-of-Words and Cosine Similarity

In this simple model, the webshops/documents and the pins/queries are represented as vectors in a common vector space, where each word represents an axis. A document is represented as a *bag-of-words* (*bow*) as its vector is given by the number of word occurrences (or term frequency, tf), $D_i = [tf_{1i}, tf_{2i}, ..., tf_V]$, where V is the vocabulary size, i.e., the number of distinct word types in the target webshop collection. The query Q is also represented as a V-dimensional vector in the same space. We can compute how similar the query and the document are using the cosine similarity between the two vectors and provide a ranked list of documents according to how similar they are to the query. This model is called *bow+Cosine*.

4.2 Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) computes a low-rank approximation of the term-document matrix (containing the term frequencies) using the top K singular values. The documents that were originally represented as V-dimensional word vectors (see sect. 4.1) may now be represented as vectors in a reduced latent K-dimensional space, where K typically is much smaller than the original rank of the term-document matrix [4]. This model is called *LSI+Cosine*.

4.3 Combining LSI and *bow* Representations

We further combine LSI with the bag-of-word representation by interpolating their similarity scores:

$$sim(Q|D_i) = \lambda sim_{bow}(Q|D_i) + (1 - \lambda) sim_{lsi}(Q|D_i) \quad (1)$$

The similarity $sim(Q|D_i)$ between the query and the document is the linear interpolation between the similarity in the bag-of-word representation $sim_{bow}(Q|D_i)$ and that of LSI, $sim_{lsi}(Q|D_i)$. λ is the interpolating parameter that weighs the contribution of each term. This model is called *bow+LSI+Cosine*.

4.4 Unigram Model

Documents are ranked by the probability $P(Q|D_i)$ that a query Q was generated by a given document model D_i. Each document is again represented as a *bag-of-words*, and a probability that each query word $q_j \in Q$ is sampled from the document model D_i is computed as follows:

$$P_{uni}(q_j|D_i) = C_u P_{mle}(q_j|D_i) + (1 - C_u)P_{mle}(q_j|Coll) \quad (2)$$

where $C_u = \frac{N_d}{N_d+\mu}$, $P_{mle}(q_j|D_i) = \frac{tf_{ji}}{N_d}$. $P_{mle}(q_j|D_i)$ denotes the maximum likelihood estimate of the word q_j in the document D_i, $P_{mle}(q_j|Coll)$ the maximum likelihood estimate in the entire collection, μ is the Dirichlet prior in the Dirichlet smoothing [10], tf_{ji} the frequency of q_j in D_i, and N_d is the length of a document D_i in terms of its number of words. The unigram language model then computes the probability of the entire query as $P_{uni}(Q|D_i) = \prod_{j=1}^{m} P_{uni}(q_j|D_i)$. This model is called *Unigram+PR*.

4.5 Using Latent Dirichlet Allocation (LDA)

Probabilistic topic models such as Latent Dirichlet Allocation (LDA) [2] provide another way of representing documents. Each document is represented as a mixture of K latent dimensions, that is, latent topics. A latent topic is represented as a probability distribution over V vocabulary words. Given a target collection $\mathcal{D}$, the aim of applying LDA is to discover the K main topics that are present in the collection. Effectively, it means computing probability scores $P(w_j|z_k)$, the probability of a word w_j given the topic z_k (these scores constitute *per-topic word distributions*), and $P(z_k|D_i)$, the probability of a topic z_k to be found in document D_i (*per-document topic distributions*). The two sets of distributions allow us to represent each target document as a probability distribution over K latent dimensions/topics. We call it a *probabilistic topical representation* of a document. Furthermore, each query word, as given by the per-topic word distributions, has a certain probability to be generated by a latent topic. The probability of a query word q_j given the target document D_i is then computed as [9]:

$$P_{lda}(q_j|D_i) = \sum_{k=1}^{K} P(q_j|z_k)P(z_k|D_i) \quad (3)$$

The probability of the entire query Q is then computed as in sect. 4.4, and documents are ranked according to their respective scores. This model is called *LDA+PR*.

4.6 Combining LDA and Unigram Representations

Similarly to combining the algebraic models, we may combine the probabilistic retrieval model that relies only on the bag-of-words representation of a document model (sect. 4.4) and the probabilistic retrieval model that relies exclusively on the probabilistic topical representation (sect. 4.5). We adopt a simple linear combination of the two models [9]:

$$P_{lda+uni}(q_j|D_i) = \lambda P_{uni}(q_j|D_i) + (1 - \lambda)P_{lda}(q_j|D_i) \quad (4)$$

where P_{uni} is the simple unigram model given by Eq. (2) and P_{lda} is the LDA model given by Eq. (3). The interpolation parameter λ weighs the importance of each method: $\lambda = 0$ reduces the model to the simple unigram model from sect. 4.4, while $\lambda = 1$ represents the LDA-only model from sect. 4.5. We study the influence of this parameter on the final linking quality for different numbers of topics K. This model is called *Unigram+LDA+PR*.

5. EXPERIMENTAL SETUP

Queries and Ground Truth. We randomly select 50 pins from our collection of one million pins. We use the text from a *single pin in isolation* (i.e., we disregard any information previously posted by the user) as a query and aim to retrieve relevant webshops/documents from the target collection of webshops. As mentioned before, the webshop documents were formed using Amazon product sub-categories known as *browse nodes* in the Amazon documentation (see Table 2). Table 1 shows basic statistics regarding the length of the query set.

Table 1: Query length statistics in terms of number of words

Minimum	Maximum	Mode	Average
1	51	2	8.06

We build the ground truth by manually annotating relevant Amazon webshops for each query. Table 2 shows examples of 10 queries annotated with the Amazon hierarchy path. We assume that a human is able to provide correct links between the pin and the relevant webshop document.

Table 2: Ground truth: Example of pins used as queries and a relevant Amazon category

Sample Pins (used as queries)	Relevant Amazon Category
Pandora New Design Fashion Lively Ladies Bracelet	Jewelry/Bracelets
Best "going home" outfit	Apparel/Baby/
Fashion, Make up, Mouth, Red	Beauty/Makeup/Lips
Sled riding!	Sporting Goods/Snow Sports
David Bromstad Kitchen	Kitchen/Furniture/Kitchen Furniture
blue suede shoes	Shoes/Women/Flats
Hue Layered Net Tights	Apparel/Women/Leggings
Mens Covington Cargo Shorts size 34 NWT	Apparel/Men/Shorts
Rebecca Minkoff 'ILY' Leather Tote	Shoes/Handbags
TIFFANY & CO. Diamond Platinum Pink Spinel 'Blue Book' Ring	Jewelry/Rings

Training Setup. LSI and LDA models are trained on Dataset II. For the LSI model, we retain the top K dimensions ranging from 50 to 600 in steps of 50, but we report only the results in the $K = 150 - 450$ range that produced the best total scores. LDA is trained with the number of topics $K = 100, 200, 500, 800, 1000$ using Gibbs sampling and the standard values for hyperparameters [8]: $\alpha = 50/K$ and $\beta = 0.01$. The Dirichlet parameter μ is also set to a standard value: $\mu = 1000$, according to [9].

6. RESULTS AND DISCUSSION

The *bow+Cosine* representation yields a Mean Average Precision $MAP = 0.3612$. The *LSI+Cosine* model presents its highest $MAP = 0.4111$ for $K = 250$. We improve the

latter results by combining these representations, *bow+LSI+Cosine*, as in Eq. (1). By decreasing λ, we increase the importance of the LSI w.r.t. the bow term, and the MAP scores improve for the range $K = 150 - 350$. Specifically, the highest performance for this combined representation $MAP = 0.4186$ is obtained for $K = 250$ and $\lambda = 0.2$.

The *Unigram+PR* model achieves $MAP = 0.3410$; while the highest score in *LDA+PR* is $MAP = 0.4091$ for $K = 800$. Again, we can improve these results by combining these models as in Eq. 4. We observe that for small values of K (e.g., $K = 100, 200$), the model is not expressive enough for the LDA representation to improve significantly over the simple unigram method. As the model refines the topic representation (i.e., as K increases), the contribution of the LDA representation becomes more helpful, and the MAP score improves w.r.t. the unigram model. The highest performance, i.e., $MAP = 0.4143$, is obtained at $K = 800$ and $\lambda = 0.1$.

Table 3 compares the best results for each method. It shows that the combination of two representations outperforms the individual ones. That is, *LDA+Unigram+PR* outperforms *Unigram+PR* and *LDA+PR* individually. Also, *bow+LSI+Cosine* performs better than *bow+Cosine* or *LSI+Cosine*. The latter combined method provides the highest overall score $MAP = 0.4186$

Table 3: Comparison of best results for each method

Method	MAP
bow+Cosine	0.3616
LSI+Cosine ($K = 250$)	0.4101
bow+LSI+Cosine ($K = 250, \lambda = 0.2$)	**0.4186**
Unigram+PR	0.3410
LDA+PR ($K = 800$)	0.4091
LDA+Unigram+PR ($K = 800, \lambda = 0.1$)	0.4143

These results seem very promising considering that we only use text from a single pin and simple IR models to provide linking to webshops. Surely, there is room for improvement. In the future, we plan to include the users' pin history to better model interests and hobbies. Additionally, we observe that the type of language used on a social media site differs greatly from the one used to describe products in an online store; even when both might be referring to the same concepts. For example, there is a pin with the words "Dark on the bottom". It refers to an eye shade that can be used below the eye. However these words may not be found in the webshop. To overcome this, we may study the use of bilingual topic modeling to learn how to link the different "languages".

Another way to improve results is to incorporate visual information which is often complementary to the accompanying text of a pin. For example, we had a query containing the single word: "browning". It is very difficult for our retrieval models -or a human- to infer what this pin refers to. It turns out the image shows manicure on finger nails. It seems that "browning" refers to a specific shape to apply with nail polish, as we later discovered. Another example is a query containing the word "stack". Before removing stop words, it said "stack it up", and it showed the image of a watch. It suggests the user's message was to stack up on this watch, maybe because it was a good deal.

7. CONCLUSION

The paper proposes a technique to recommend "webshops" (a.k.a. Amazon pages of items) with respect to pins of individuals on Pinterest. The basic premise is that pins represent general interests and hobbies of people. While Pinterest has a mechanism to automatically recommend similar pins, there's no way to leverage this as an advertising opportunity. The paper attempts to do that, by instead recommending item pages from Amazon, which are similar to the pins, and therefore hold potential of future purchase.

We have investigated a particular minimalist setting where we model the user's interests using a single pin without any additional information. We have framed the linking task as an ad-hoc IR task, where users' pins are treated as queries, and webshops as target documents that need to be retrieved/linked. Following that, we have studied the potential and the influence of various IR models on the retrieval performance. The combination of the latent semantic and the bag-of-words representations have yielded the highest mean average precision score, $MAP = 0.4186$. In our future work, we will explore incorporating previous pins to grasp a better user representation. We will also investigate deeper representations (such as the bilingual topic models) and the addition of visual information.

Acknowledgments

This project is part of the SBO Program of the IWT (IWT-SBO-Nr. 110067).

8. REFERENCES

[1] A. Bellogín, J. Wang, and P. Castells. Text retrieval methods for item ranking in collaborative filtering. In *ECIR*, pages 301–306, 2011.

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

[3] A. Costa and F. Roda. Recommender systems by means of information retrieval. In *WIMS*, number 57, 2011.

[4] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[5] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[6] J. Nielsen. *Multimedia and Hypertext: the Internet and Beyond*. Academic Press Professional, Inc., 1995.

[7] R. Ottoni, J. P. Pesce, D. L. Casas, G. F. Jr., W. M. Jr., P. Kumaraguru, and V. Almeida. Ladies first: Analyzing gender roles and behaviors in Pinterest. In *International AAAI Conference on Weblogs and Social Media*, 2013.

[8] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 427(7):424–440, 2007.

[9] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.

[10] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, Apr. 2004.