Are Words Enough? A Study on Text-Based Representations and Retrieval Models for Linking Pins to Online Shops

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

User-generated content offers opportunities to learn about people's interests and hobbies. We can leverage this information to help users find interesting shops and businesses find interested users. However this content is highly noisy and unstructured as posted on social media sites and blogs. In this work we evaluate different textual representations and retrieval models that aim to make sense of social media data for retail applications. Our task is to link the text of pins (from Pinterest.com) to online shops (formed by clustering Amazon.com's products). Our results show that document representations that combine latent concepts 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, unstructured data, user interests, recommendation systems, personalized linking

1. INTRODUCTION

User-generated content is ubiquitous on the web. People freely post information on social media sites, such as Facebook, Twitter and Pinterest. This content provides a potentially rich source of information for business intelligence applications that leverage this content for personalisation, such as recommender systems and on-line marketing. Making sense of the vast amount of information is a challenging problem due to poor data structure and noise (misspelled words, grammatical errors, etc). We wish to study how to leverage the content on such sites for retail applications.

In particular, we use data from `Pinterest.com`. Pinterest 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 "Christian Louboutin" shoes, a related online store might benefit from that information.

In the context of noisy and unstructured data, it becomes important to utilize document representations and retrieval models that allow us to extract relevant semantic information and retrieve relevant documents. The goal of this work is to study the performance of different representations and models in the task of 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

UnstructureNLP '13, October 28 2013, San Francisco, CA, USA
Copyright 2013 ACM 978-1-4503-2415-1/13/10 ...$15.00.
http://dx.doi.org/10.1145/2513549.2513557

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 the text of a single pin (a *query*), the task is to rank a set of webshops (*documents*) 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 several text representations and ad-hoc IR models.

2. RELATED WORK

As previously stated, we aim to both extract useful information about users' interests as posted on a social media sites, and link or recommend online shops relevant to such interests.

Extracting useful information from micro-blogs, such as Twitter, Facebook and Pinterest, calls for textual representations beyond the simple bag-of-words. Some work has been done using topic representations to discover the latent themes: [5] studied how training a topic model on Twitter data can improve performance in document classifications, [7] explored topic models for analyzing disaster-related Twitter data, [13] investigated how to improve topic models given the short and messy texts on tweets. Regarding Pinterest data, some work has been done to perform board recommendations [6], and implementations of topic models to understand users' interests [15]. However, there is little work regarding product recommendations in this setting.

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 [12].

Many of the recommendation systems like Amazon's item-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 [14], 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

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).

4. DATA

4.1 Dataset I: Pinterest.

We implemented a crawler to find Pinterest users, their boards and pins. A user page contains a set of boards. Boards are often categorized by the user and include fashion, travel, cars, food, film, humor, home design, sports, and art. A board is a collection of pins often related to a given category.

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.

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.

4.2 Dataset II: Online Shops.

For this study, we formed webshop documents using Amazon's product categories. 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.

We implemented an XML parser to download information from over 23,000 products. Amazon organizes its items in a hierarchy of browse nodes (or categories). Each node is a collection of related items, i.e., products that belong to the same category. We focused on a set of top categories: Apparel, Beauty, Books, Electronics, Groceries, Jewelry, Kitchen, Music, Shoes, Sporting Goods and Watches. Leaf nodes were used as a natural subgroup of similar items. Thus, we were able to cluster groups of related products. We call these product clusters ”webshops”. We started by querying the top categories and gathered the hierarchy of related child nodes. Each webshop contains approximately 20 products. To represent the product, we used all the text associated to the product's description and editorial review. The idea here is to simulate an online retail business that has a set of webshops containing related products. Our application aims to direct users in a social media site to such target webshops. For our experiments, we acquired 1,171 webshop documents.

Although we use Amazon data in this work, the proposed IR framework may be extended to any other online webshop for which a textual representation is provided.

5. METHODOLOGY

In the task of linking relevant webshops to users' pins, we utilize different *text document representations* and *retrieval models* that are built upon these representations. 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.

5.1 Document Representations

5.1.1 Bag-of-Words (BOW)

In this simple model, the webshops (or documents) and the pins (or queries) are represented as vectors in a common vector space, where each word represents an axis. A document D_i is represented as a *bag-of-words* (*bow*) as its vector is given by the number of word occurrences (or term frequency, tf), $D_i = [tf_{1i}, tf_{2i}, ..., tf_{|V|i}]$, where $|V|$ is the vocabulary size, i.e., the number of distinct word types in the target document collection. The order of words is disregarded. This representation allows us to compute the probability of a term given the document, as we describe in sect. 5.2.1.

5.1.2 Topic Representation (TR)

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 the 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*) (see sect. 5.2.2). The two sets of distributions allow us to represent each target document as a probability distribution over K latent dimensions/topics. One advantage of this representation compared to *BOW* relies on the ability to cluster semantically similar co-occurring terms. The number of latent topics K is smaller than the original size of the vocabulary $|V|$. Thus, we are able to reduce the dimensionality of the document representation. Additionally, it provides a fully generative probabilistic modelling of text. We call it a *topic representation (TR)* of a document.

5.2 Retrieval Models - Linking Pins to Webshops

The task of linking pins to webshops can be tackled with information retrieval techniques. Given a query (i.e., a pin), we rank the webshop documents according to the relevance to the query. Here we describe the retrieval models and sect. 6.1 presents a description of the parameters used.

5.2.1 Probabilistic BOW only

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 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,

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

where

$$C_u = \frac{N_d}{N_d + \mu}, \qquad P_{mle}(q_j|D_i) = \frac{tf_{ji}}{N_d}. \quad (2)$$

$P_{mle}(q_j|D_i)$ 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 [20], tf_{ji} the frequency of q_j in D_i, and N_d is the length (number of words) of a document D_i. The unigram language model then computes the probability of the entire query as

$$P_{bow}(Q|D_i) = \prod_{j=1}^{m} P_{bow}(q_j|D_i) \quad (3)$$

This model is called *BOW-only*.

5.2.2 Probabilistic TR only

We train a topic model with K topics using Gibbs sampling [16] and obtain two conditional probability distributions: The probability $P(w_j|z_k)$ of a word w_j given the topic z_k is given as

$$P(w_j|z_k) = \frac{n_k^{(w_j)} + \beta}{\sum_{t=1}^{|V|} n_k^{(w_t)} + |V|\beta} \quad (4)$$

$n_k^{(w_j)}$ denotes the number of times that the topic z_k was assigned to word w_j occurring at a certain position in the documents. The sum $\sum_{t=1}^{|V|} n_k^{(w_t)}$ is the total number of words assigned to the topic z_k, and |V| is the number of distinct words in the corpus vocabulary.

The probability $P(z_k|D_i)$ of topic z_k given the document D_i is

$$P(z_k|D_i) = \frac{n_i^{(k)} + \alpha}{\sum_{j=1}^{K} n_i^{(j)} + K\alpha} \quad (5)$$

$n_i^{(k)}$ is the number of times a word in document D_i is assigned to the topic z_k.

α and β are the Dirichlet priors and can be interpreted as a prior observation for the number of times a topic is sampled in a document, before having observed any actual words from that document.

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 [18]:

$$P_{tr}(q_j|D_i) = \sum_{k=1}^{K} P(q_j|z_k) P(z_k|D_i) \quad (6)$$

The probability of the entire query Q is then computed analogously to Eq. 3, and documents are ranked according to their respective scores. This model is called *TR-only*.

5.2.3 Probabilistic BOW + TR

We may combine the probabilistic retrieval model that relies only on the bag-of-words representation of a document model (sect. 5.2.1) and the probabilistic retrieval model that relies exclusively on the probabilistic topical representation (sect. 5.2.2). We adopt a simple linear combination of the two models [18]:

$$P_{bow+tr}(q_j|D_i) = \lambda P_{bow}(q_j|D_i) + (1 - \lambda) P_{tr}(q_j|D_i) \quad (7)$$

where P_{bow} is the simple bag-of-words model given by Eq. (1) and P_{tr} is the topic model given by Eq. (6). The interpolation parameter λ weighs the importance of each method: $\lambda = 0$ reduces the model to the simple *bow* model from sect. 5.1.1, while $\lambda = 1$ represents the topic representation model from sect. 5.2.2. We study the influence of this parameter and the final linking quality for different numbers of topics K. This model is called *BOW+TR.*

5.2.4 Probabilistic Relevance-Based Models PRM

Furthermore, it is possible to exploit the aforementioned basic probabilistic retrieval models in a more robust and a more effective framework of probabilistic relevance modeling [11, 10]. Relevance models do not rely on any training data and provide a framework that exhibits state-of-the-art performance in a variety of ad-hoc retrieval tasks across different corpora. For instance, Lavrenko et al. [11, 10] already show that embedding the simple probabilistic unigram BOW only model (see sect. 5.1.1) into the relevance modeling framework leads to a significant increase in overall retrieval quality on TREC data and in the TDT topic tracking task. Additionally, recent work reveals that fusing the shallow semantic knowledge coming from probabilistic topic models with the power of relevance models for retrieval leads to the highest scoring models on TREC and CLEF data for both monolingual [19, 17] and cross-lingual ad-hoc retrieval [17]. Therefore, in this paper, our aim is also to investigate the potential of the relevance modeling framework in our "pins-to-webshops" linking task.

In short, the term *relevance model* addresses a probability distribution that specifies the expectancy that any given word is observed in a set of documents R_Q relevant to the issued query Q. We can again assume that we are given the query $Q = \{q_1, \dots, q_m\}$ consisting of m words, and a target collection of documents/webshops $\mathcal{D}$. The relevance model of the query Q is actually the set of probabilities $P(w_j|R_Q)$ for each word $w_j \in V$, where $P(w_j|R_Q)$ denotes the probability that we will sample exactly the word w_j from the set R_Q of documents relevant to the query Q. Since the relevance modeling framework completely replaces the original query with a distribution over the entire vocabulary, we can observe it as a massive and a robust query expansion technique [11].

In order to estimate all the probability scores $P(w_j|R_Q)$ in the absence of any training data, we follow the approach from Lavrenko et al. [10], also used by [17], that makes the computation of a relevance model tractable:[1]

$$P(w_j|R_Q) \approx P(w_j|Q) = \sum_{D_i \in \mathcal{D}} P(w_j|D_i)P(D_j|q_1, \dots, q_m) \quad (8)$$

The posterior probability $P(D_i|q_1, \dots, q_m)$ is then further expressed as:

$$P(D_i|q_1, \dots, q_m) = \frac{P(D_i)\prod_{j=1}^{m} P(q_j|D_i)}{\sum_{D_l \in \mathcal{D}} P(D_l)\prod_{j=1}^{m} P(q_r|D_l)} \quad (9)$$

$P(D_i)$ denotes a prior probability of a document D_i, and it is usually taken to be a uniform distribution over all $D_i \in \mathcal{D}$. Further, the probability scores $P(w_j|D_i)$ and $P(q_j|D_i)$ from Eq. (8) and Eq. (9) may be computed using any of the previous retrieval models (e.g., the probabilistic BOW-only, TR-only or BOW+TR retrieval models).

Lavrenko et al. [10] notice that the posterior probability $P(D_i|q_1, \dots, q_m)$ from Eq. (9) displays negligible near-zero values for all but a few documents $D_i \in \mathcal{D}$. These target documents are the documents that consitute the relevance set for the query Q, i.e., they obtain the highest scores for the query Q. In the absence of any relevance assessments, in order to speed up the retrieval process, we calculate Eq. (9) over only the top M target documents for the query Q instead of calculating it over the entire target collection. The top M documents are obtained by retrieving the ranked list of documents with a query likelihood model (e.q., any of the BOW-only, TR-only or BOW+TR models). The influence of the parameter M was analyzed previously in the literature; when M is set to a large enough value, it does not influence the qualitative interpretation of the results [9].

Given the target collection $\mathcal{D}$ and a query Q, the final retrieval process follows these key steps:

1. Perform the *first retrieval round* with any basic query-likelihood retrieval model (e.g., probabilistic BOW-only, TR-only or BOW+TR).

2. Retain only M top scoring documents from the previous step as *pseudo-relevant documents.*

3. Estimate the probability scores $P(q_j|D_i)$ and $P(w_j|D_i)$ again using any of the basic models (see Eq. (1), Eq. (6) and Eq. (7)), but only for the M documents.

4. Estimate the relevance model $P(w_i|R_Q)$ for each $w_i \in V$ by calculating Eq. (8) and Eq. (9) over these M documents.

5. Perform the *second retrieval round* over the target collection $\mathcal{D}$ or just rerank a number of top scoring documents retrieved in the first retrieval round. Each document D_i is assigned a score that is the relative entropy (the Kullback-Leibler divergence) between a relevance model R_Q and the exact target document D_i:

$$KL(R_Q||D_i) = \sum_{w_j \in V} P(w_j|R_Q) \log \frac{P(w_j|R_Q)}{P(w_j|D_i)} \quad (10)$$

6. Rank documents in terms of their increasing relative entropy score.

6. EXPERIMENTAL SETUP

6.1 Training Setup

For the *BOW-only* model, we use Dataset II only to build the term-document matrix that contains the term frequencies.

For the *TR-only* model, we explore two ways to learn the latent topics. The first one uses Dataset II only. In this case, only the vocabulary from the webshops is used to learn the topic distributions. We call this **Setup I**.

The second approach combines both the webshop documents and the Pinterest boards, i.e., Dataset II + Dataset I. The idea here is to incorporate the vocabulary employed

[1] The interested reader may find much more information on relevance modeling in the relevant literature [11, 10, 19, 8, 17].

by users on the social media site to learn a richer and more expressive topic representation. We call this **Setup II**.

The TR model is trained with the number of topics $K = 100, 200, 500, 800, 1000$ using Gibbs sampling and the standard values for hyperparameters [16]: $\alpha = 50/K$ and $\beta = 0.01$. The Dirichlet parameter μ is also set to a standard value: $\mu = 1000$, according to [18].

When we combine BOW+TR models as in Eq. 7, we use values of λ from 0 to 1 in 0.1 intervals.

For the PRM model, the first retrieval round uses BOW + TR. The second retrieval round uses two cases: BOW-only and BOW+TR. In this paper, we do not provide results obtained by all possible combinations of the basic models in the relevance modeling framework (see Step 1 and Step 3 of the retrieval process), but rather focus on a subset of models that best illustrate the importance and the robustness of the relevance modeling retrieval framework.

6.2 Queries & 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 (number of words) of the query set.

Table 1: Query length (number of words) statistics

Minimum	Maximum	Mode	Average
1	51	2	8.06

We build the ground truth by manually annotating relevant Amazon webshops for each query. The annotator is presented with both the text and image of the pin to identify all the relevant webshops. Table 2 shows examples of 20 queries annotated with all relevant Amazon hierarchy path available in our dataset. For each path, there are five webshop documents containing 20 products each. We assume that a human is able to provide correct links between the pin and the relevant webshop documents.

6.3 Evaluation

To evaluate our retrieval models, we compute the Mean Average Precision (MAP) for the set $Q = \{Q_1, Q_2, ..., Q_s\}$, where s is the number of queries. Let $\{D_1, ..., D_{c_j}\}$ be the set of c_j relevant documents for an information need $Q_j \in Q$. Let R_{jk} be the set of ranked retrieved results ordered from the highest scored document until the relevant document D_k is reached. The MAP score for the set Q is given as

$$MAP = \frac{1}{s} \sum_{j=1}^{s} \frac{1}{c_j} \sum_{k=1}^{c_j} Precision\, R_{jk}, \qquad (11)$$

where precision is the fraction of the documents retrieved that are relevant to the query. When a relevant document is not retrieved, the precision value in the above equation is zero.

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
Paint first coat then before second coat sets press lines with a ruler diagonally quilted nails	Beauty/Makeup/ Nails/Nail Art
Baby Hat Brown Wig Hat Winter Cap Christmas Gift Ideas by YumBaby, $29.95	Apparel/Baby/
Chair Pose From Three Minute Egg Yoga Pose Weekly	Apparel/Women/Active
Luna Sofa	Kitchen/Furniture/ Living Room Furniture
high-heels-2	Shoes/Women/Pumps
Lace-up Fur Ankle Boots High Heels	Shoes/Women/Pumps
Stripe nail with blue points	Beauty/Makeup/Nails/ Nail Art
Tips To Stay Fit and Healthy	Books/Health, Fitness & Dieting
A Ten Step Guide to Nailing Office Style	Apparel/Men/Suits
beautiful white wedding dress & wedding bouquet - pink rose & little white flowers	Apparel/Women/ Dresses/Wedding Dresses
Gifts Under 50: Coach Stone Stud Earrings	Jewelry/Earrings

7. RESULTS & DISCUSSION

Table 3 presents the results for *BOW-only*, *TR-only* and *BOW + TR* where the latent topics were trained using only the webshop documents (Setup I). When $\lambda = 1$, Eq. 7 reduces to *BOW-only* model, Eq. 1. When $\lambda = 0$, Eq. 7 reduces to *TR* model, Eq. 6.

For the BOW-only model we obtain a score MAP = 0.3410. We observe that for small values of K (e.g., $K = 100, 200$), the model is not expressive enough for the TR representation to improve significantly over the simple BOW method. As the model refines the topic representation (i.e., as K increases), the contribution the TR representation becomes more helpful and the MAP score improves with respect to the BOW model. Fig. 1 visualizes this behaviour. The highest performance $MAP = 0.4143$ is obtained at $K = 800$ and

$\lambda = 0.1$.

Table 3: MAP for Setup I BOW+TR over different values for K and λ

	K				
λ	100	200	500	800	1000
0.0	0.2514	0.3006	0.3993	0.4091	0.3666
0.1	0.2898	0.3271	0.3982	**0.4143**	0.3652
0.2	0.3128	0.3416	0.3981	0.4115	0.3704
0.3	0.3337	0.3460	0.3993	0.4107	0.3712
0.4	0.3495	0.3548	0.4056	0.4053	0.3667
0.5	0.3612	0.3652	0.4037	0.4063	0.3699
0.6	0.3648	0.3708	0.4051	0.4032	0.3694
0.7	0.3694	0.3739	0.4024	0.3985	0.3685
0.8	0.3704	0.3695	0.4004	0.3914	0.3677
0.9	0.3701	0.3645	0.3844	0.3791	0.3638
1.0	0.3410	0.3410	0.3410	0.3410	0.3410

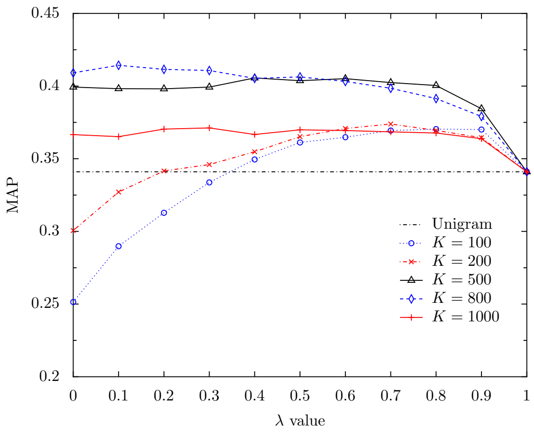

Figure 1: Effect of λ for BOW + TR in Setup I

Table 4 presents the results for *BOW + TR* where the latent topics were trained using both webshop documents and Pinterest boards (Setup II). Fig. 2 shows the effect varying λ. We see that for the range $\lambda = 0.4 - 0.7$, the MAP scores are the highest for $K = 800$. This emphasizes the usefulness of combining two retrieval methods. For this setup, the highest performance $MAP = 0.4071$ is obtained at $K = 800$ and $\lambda = 0.6$.

Table 4: MAP for Setup II BOW+TR over different values of K and λ

	K				
λ	100	200	500	800	1000
0.0	0.2452	0.3133	0.3279	0.3648	0.3541
0.1	0.2887	0.3547	0.3456	0.3816	0.3626
0.2	0.3232	0.3700	0.3515	0.3894	0.3752
0.3	0.3418	0.3843	0.3589	0.3994	0.3785
0.4	0.3659	0.3895	0.3695	0.4056	0.3824
0.5	0.3722	0.3953	0.3746	0.4049	0.3887
0.6	0.3822	0.4009	0.3786	**0.4071**	0.3917
0.7	0.3897	0.4012	0.3810	0.4031	0.3895
0.8	0.3872	0.3923	0.3775	0.3971	0.3843
0.9	0.3797	0.3752	0.3676	0.3955	0.3808
1.0	0.3410	0.3410	0.3410	0.3410	0.3410

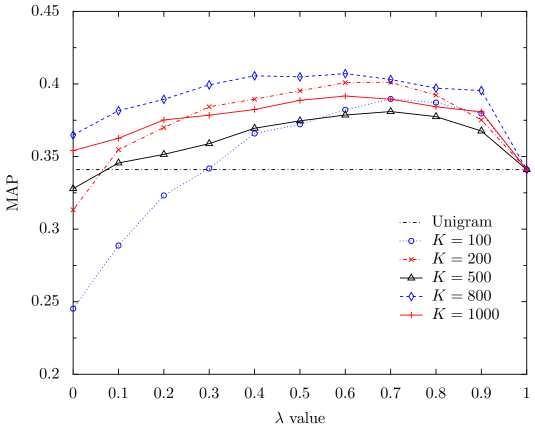

Figure 2: Effect of λ for BOW + TR in Setup II

Table 5: MAP for PRM: First Round: BOW+TR. Second Round: BOW-only for different values of K

	K				
	100	200	500	800	1000
Setup I	0.4332	0.4214	0.4165	0.4421	0.4188
Setup II	0.4408	0.4405	0.4425	**0.4691**	0.4208

Comparing Setup I and Setup II results as shown in Table 3 and Table 4, respectively, it seems that incorporating documents from social media to define the latent topics does not help to improve the MAP scores. This could be because Setup II introduces a larger and more varied vocabulary, and we might need many more documents to have robust and accurate topics. For instance, when we add the Pinterest data -as in Setup II-, we introduce about 100,000 new word types not present in Amazon data. Moreover, adding Pinterest data might actually slightly spoil the topics. That is, the new data might drag the topics away from the actual word distributions that are present in the target document collection. In spite of this, we see that the relevance models actually leverage this information and improve the overall scores.

Table 6: MAP for PRM: First Round: BOW+TR. Second Round: BOW+TR for different values of K

	K				
	100	200	500	800	1000
Setup I	0.4195	0.4143	0.4326	0.4226	0.4333
Setup II	0.4357	0.4366	0.4576	**0.4694**	0.4206

Table 7 compares the best results for each method. It shows that the combination of two representations outperforms the individual ones. That is, *BOW + TR* outperforms *BOW-only* and *TR-only* individually on both Setup I and Setup II. Moreover, the combination of retrieval methods, given by the relevance models, shows higher improvements. Performing two-round (PRM model) retrieval outperforms one-round models in both setups. Interestingly, we obtain the highest results when we use the PRM model in Setup II, i.e., combining documents from both webshops and Pinterest users.

These results seem very promising considering that we only use text from a single pin and topical representation of

Table 7: Comparison of best results for each method

Setup	Method	MAP
I	BOW-only	0.3410
	TR-only K =800	0.4091
	BOW+TR K=800, $\lambda = 0.1$	0.4143
	PRM (Round1: BOW+TR, Round 2:BOW) K=800	0.4421
	PRM (Round 1: BOW+TR, Round 2: BOW+TR) K=800	0.4333
II	TR-only K=800	0.3648
	BOW+TR K=800, λ=0.6	0.4071
	PRM (Round1: BOW+TR, Round 2:BOW-only) K=1000	0.4691
	PRM (Round 1: BOW+TR, Round 2: BOW+TR) K =800	**0.4694**

documents within the IR models. Surely, there is room for improvement. 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 pin with the words ”Be daring, go all out in red! Modern Jessica Rabbit”. It refers to dressing up in red, similar to the style of the cartoon character ”Jessica Rabbit”. The user is talking about a sexy red dress but never mentions these explicit words. Instead, she mentions a cartoon character that wears such style. Another example: a pin with the words ”Dark on the bottom”. It refers to an eye shade that can be used bellow the eye. However these words may not be found in the webshop. To overcome this, we may study the use of bilingual topic modeling [4] 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, if a pin contains the words ”I love it”, it does not provide any information about the item. We only know the user’s sentiment about the item. However, if the picture shows a wedding dress, the visual information can help us refine and disambiguate the pin contents. We also 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. In short, the combination of both textual and visual signals can potentially improve our results. We will look further into this.

8. CONCLUSIONS

In this work we have studied different document representations and retrieval models for the task of automatically linking Pinterest pins to online shops. This task faces the challenges of dealing with highly noisy and unstructured textual data. It is also an interesting task for both users and retailers. 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. We used the bag-of-words (BOW) and the topic representations (TR). The latter always outperformed the former for the different retrieval models. This confirms that the simple word representation of BOW is not enough for dealing with the inherent poor structure and noise of user-generated content. The best results obtained correspond to a two-round retrieval model (PRM), where both webshop documents and Pinterest boards were combined for training. We achieved $MAP = 0.4694$ for $K = 800$ (number of latent topics), as shown in Table 7. We used a collection of 23,000 Amazon products to form webshop documents. We also had a collection of one million pins from which the Pinterest board documents were formed. A natural extension of our work is to further increase the size of the data and study how well our representation and methods scale up. We will also investigate other probabilistic generative models, such as bilingual latent Dirichlet Allocation and incorporate visual information to our system.

Acknowledgments

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

9. 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] W. De Smet and M.-F. Moens. Cross-language linking of news stories on the web using interlingual topic modelling. In *Proceedings of the 2nd ACM workshop on Social web search and mining*, SWSM ’09, pages 57–64, New York, NY, USA, 2009. ACM.

[5] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA ’10, pages 80–88, New York, NY, USA, 2010. ACM.

[6] K. Y. Kamath, A.-M. Popescu, and J. Caverlee. Board recommendation in Pinterest. In *UMAP Workshops*, 2013.

[7] K. Kireyev, L. Palen, and K. Anderson. Applications of Topics Models to Analysis of Disaster-Related Twitter Data. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, Dec. 2009.

[8] V. Lavrenko. *A generative theory of relevance.* Springer, 2009.

[9] V. Lavrenko and J. Allan. Real-time query expansion in relevance models. Technical report, 2008.

[10] V. Lavrenko, M. Choquette, and W. B. Croft. Cross-lingual relevance models. In *SIGIR*, pages 175–182, 2002.

[11] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR*, pages 120–127, 2001.

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

[13] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In *The 36th Annual ACM SIGIR Conference*, page 4, Dublin/Ireland, July 2013.

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

[15] A.-M. Popescu. Pinteresting: towards a better understanding of user interests. In *DUBMMSM*, pages 11–12, 2012.

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

[17] I. Vulić and M.-F. Moens. A unified framework for monolingual and cross-lingual relevance modeling based on probabilistic topic models. In *ECIR*, pages 98–109, 2013.

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

[19] X. Yi and J. Allan. A comparative study of utilizing topic models for information retrieval. In *ECIR*, pages 29–41, 2009.

[20] 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, 2004.