Diversifying Product Review Rankings: Getting the Full Picture

Ralf Krestel
L3S Research Center
Hannover, Germany
Email: krestel@l3s.de

Nima Dokoohaki
Royal Institute of Technology (KTH)
Stockholm, Sweden
Email: nimad@kth.se

***Abstract*—E-commerce Web sites owe much of their popularity to consumer reviews provided together with product descriptions. On-line customers spend hours and hours going through heaps of textual reviews to build confidence in products they are planning to buy. At the same time, popular products have thousands of user-generated reviews. Current approaches to present them to the user or recommend an individual review for a product are based on the helpfulness or usefulness of each review. In this paper we look at the top-k reviews in a ranking to give a good summary to the user with each review complementing the others. To this end we use Latent Dirichlet Allocation to detect latent topics within reviews and make use of the assigned star rating for the product as an indicator of the polarity expressed towards the product and the latent topics within the review. We present a framework to cover different ranking strategies based on the user's need: Summarizing all reviews; focus on a particular latent topic; or focus on positive, negative or neutral aspects. We evaluated the system using manually annotated review data from a commercial review Web site.**

***Keywords*-Ranking, Topic Models, Summarization, Diversification, Review Recommendation.**

I. Introduction

It has become a routine among on-line and off-line consumers to inform themselves on review Web sites before purchasing a certain product. This has given rise to a considerable amount of customer reviews on e-commerce Web sites. To this end potential customers usually browse through a lot of on-line reviews in order to build confidence in a particular item prior to purchasing it. While reviews have become an important factor in helping Web crowds to further assess the quality of products on-line, increasing volume of reviews themselves has led to an information overload [1]. Popular products have thousands of reviews. While excess of reviews is a growing problem, recommending unbiased and helpful reviews is a growing research field. The quality of reviews may vary drastically [2] and might mislead potential buyers. Such humongous amount of information not only distracts the confidence seeker, it might hinder the original goal of users in the first place: They will give up buying a certain product. To deal with these problems, review recommendation techniques are proposed. Review recommendation involves implementing machine-learning techniques for analyzing the product reviews based on their lexico-semantic features in order to classify the reviews and recommend balanced and useful reviews to the readers.

While review recommender systems aim at automatically classifying reviews, some commercial Web sites such as Amazon and TripAdvisor[1] approach this problem by allowing users to rate the reviews using star ratings to improve the rankings (e.g. *this review was helpful vs. not helpful*). There are two inherent problems to these ranking based on user feedback: First, good objective reviews contain quite likely redundant information and ranking them based on the helpfulness score will not cover all aspects. Second, these Web sites do not take into account the personal bias. Not all reviews are helpful to everybody. Due to the fact that different users put different emphasis on different aspects, (e.g. *I don't care about battery life, but really need lots of memory*), helpfulness can only be used to filter out very bad reviews. Therefore, researchers are increasingly distinguishing between the task of review recommendation [3] and review ranking [4]. To improve existing review recommendation techniques and at the same time improve the ranking used for evaluating helpfulness merits of existing reviews, we propose a novel approach to model and rank reviews. The two main components of our system rely on Latent Dirichlet Allocation (LDA) to model the reviews and on Kullback-Leibler divergence to generate an adequate ranking. We make use of the assigned star rating for the product as an indicator of the polarity expressed in the review towards the latent topics. Our framework covers different ranking strategies based on users' needs to adapt to various user scenarios. We currently support three strategies to summarize all reviews, to focus on a particular latent topic, or to focus on positive, negative or neutral aspects. We evaluated the system using manually annotated review data gathered from a popular review Web site.

The main contributions of this paper are: (1) Introducing an algorithm to model reviews using latent topics and star ratings. (2) Ranking of reviews to summarize all reviews for a product within the top-k results. (3) Diversification of review rankings based on star ratings and/or latent topics. The remaining of the paper is organized as follows: We present related work in Section II; Section III gives an overview of our framework. Section IV describes the modeling approach, while Section V describes the ranking approach. We present the evaluation in Section VI and close with conclusions and future work.

[1] http://www.amazon.com and http://www.tripadvisor.com

II. Related Work

We divided related work into two sections: Recommendation and summarization on the one hand, and ranking and diversifying reviews on the other. Since we aim at ranking reviews to summarize opinions we describe in this section related work from the areas of recommender systems and information retrieval.

A. Review Recommendation and Summarization

Generally review recommendation techniques are seen as an explanation [5], [6] or classification problem [7]. O'Mahony et al. [7] give an overview over existing machine learning methods for review recommendation. Kim et al. [8] use SVM regression on structural, lexical, syntactic, semantic, and sentiment features to classify reviews, and stated that helpfulness is very dependent on the length of a review, its unigrams and score. Liu et al. [9] have shown that helpfulness of movie reviews are expertise and time dependent. While the majority of existing work utilize text categorization techniques for recommending reviews Harper et al. [10] train their classifier according to features relating to question categories, text categorization and social networking metrics. Credibility assessment has also been considered by Weekamp et al. [11] to consider features such as timeliness of posts, post length and spelling quality in topical reviews.

Existing work on review summarization falls under subjective classification [12], sentiment analysis [13], or under traditional text summarization. Classic text summarization methods can be categorized into two – template instantiation [14] and passage extraction [15]. Researchers differentiate between review summarization and classic text summarization techniques [16]. Sentiment analysis techniques try to produce a summarized sentiment consisting of sentences from a source document presenting the opinion and idea of it's corresponding author. With respect to length and structure, this summary can be either a single paragraph [17] by careful selection of sentences or the source document, or a structured sentence [18], which is in turn generated by mining features that the author has commented on. To build summaries of sentence list structures, Hu and Liu [18] introduced a method utilizing word attributes such as frequency of occurrence, part-of-speech tagging and WordNet synsets. Following this approach features are extracted, combined with their contextually close words, and finally used to generate a summary by selecting and re-structuring the sentences following the extracted features. Implementations following text sentiment analysis have been proposed such as Opine [19], which uses relaxation labeling to find the lexico-semantic orientation of words, or Pulse [20] which uses bootstrapping to train a sentiment classifier using features extracted by labeling sentence clusters with respect to their key terms.

In comparison with these works we summarize reviews by choosing complementing reviews and ranking them according to different strategies. The product ratings serve as an indicator for the sentiment, and the extracted latent topics ensure topical coverage of relevant aspects.

B. Diversifying Review Rankings

The problem of personalized ordering of results has been subject to research in both classic retrieval of documents as well as increasingly popular recommender systems. Maximum Marginal Relevance (MMR) [21] was used as a ranking metric which balances relevance as the similarity between query and search results with diversity as the dissimilarity among search results. Ziegler et al. [22] take into account a user's full range of interests through diversifying generated recommendation lists and by doing so they minimize redundancy among the recommended items. Reranking methods are mainly used for diversifying search results. Radlinski and Dumais [23] use a log-driven query reformulation with focus on personalized search results. Chen and Karger [24] introduce a Bayesian reranking method to maximize the coverage of various semantics of an issued query among top 10 results visited before. Zhai and Lafferty [25] introduce subtopic retrieval that considers dependencies between search results. They use statistical models to model user preferences as loss functions and the retrieval process as a risk minimization problem. Sanderson et al. [26] consider diversity in image search results and they study the relation between precision and result diversity.

Recent approaches to diversification balance relevance with diversity, though they differ in estimation of relevance and similarity, and choice of diversification objective. Gollapudi and Sharma [27] consider existing approaches to diversification as variants of facility dispersion. They analyze and evaluate various diversification objectives such as MaxSum, MaxMin and MonoObjective. Wang and Zhu [28] introduce an approach for search result diversification adopting the Modern Portfolio Theory of finance. They generalize this well-known principle by maximizing the relevance of top-k as well as minimizing the (co-)variance of the results. A greedy algorithm is used for ranking search results such that relevance is maximized while variance is minimized. Rafiei et al. [29], introduce a similar framework based on Portfolio Theory for reranking Web search results. The problem of result diversification is also investigated in the area of structured data queries. Agarwal et al. [30], classify queries and results to categories of the ODP taxonomy, and diversify results by maximizing sum of categories covered by top-k results, weighted by the probability of categories given the query. Recommending a set of items to a user, as well as returning a query results have been subject to result diversification as well. Vee et. al. [31] propose an algorithm for finding a representative, diverse set of top-k results for a given query. All attributes of an object are ordered according to their priority for diversification by a domain expert.

We propose a greedy algorithm to minimize the Kullback-Leibler divergence (KLD) between the topic models of the top-k results and all reviews for a product. KLD has, e.g., been used as a similarity measure for audio files [32], while we use it to diversify topic models. In addition, we diversify review rankings based on latent topics to get an optimal coverage for all topics within the top-k results.

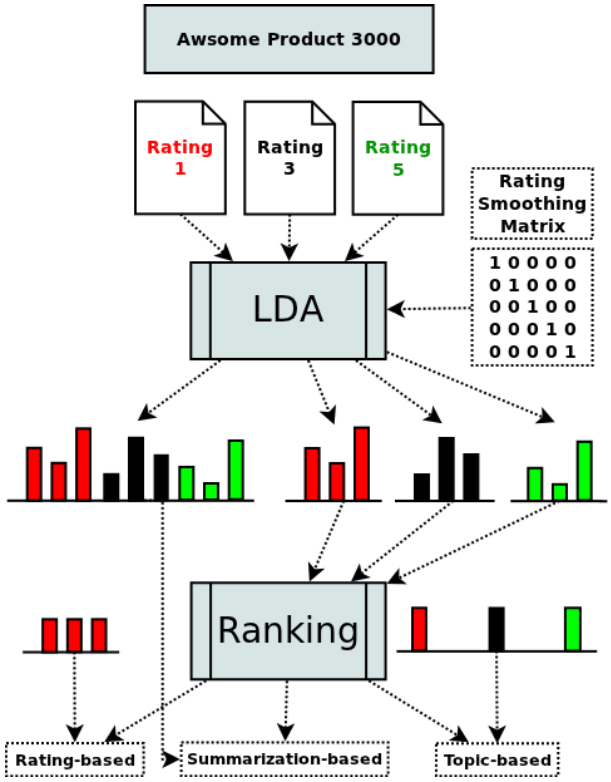


Fig. 1. Overview of the Review Ranking System: Reviews together with ratings are used to extract topic distributions using LDA. Rankings are computed minimizing KL-Divergence with task-specific target distributions.

III. How to Rank Reviews?

In contrast to Web search results, reviews for a product can not be ranked based on relevance since all reviews are equally relevant for the product the review is about. As discussed in Section II review recommendation or classification is a well-studied problem, but they don't optimize a ranking of reviews but evaluate the reviews individually. We try not to find the best or most helpful individual reviews for a product but to find the top-k reviews to provide the user with a good summary of the opinions about a product. To this end we model the reviews using latent topics extracted with Latent Dirichlet Allocation (LDA) and the assigned star ratings for the product. The ranking of the reviews is based on Kullback-Leibler divergence (KLD) to get an optimal summary of all reviews for a product with the largest possible topical diversity. Our framework also allows to set a different goal when computing the optimal ranking, e.g. cover all positive aspects of a product, or cover all sentiments associated with an aspect/feature of the product. In the following section we describe the conceptional architecture of the framework.

A. System Overview

Our framework consists of two main components:

1) The LDA component to model the review data
2) The Ranking component to optimize the ranking based on different strategies

> "...Wish burger - also known as a veggie burger (no meat) Ketchup and Mustard are actually available at In and Out.. just ask, its really easy Double Meat - is la double double with no cheese Flying Dutchman..."

wish.n burger.n also.r know.v veggie.n_burger.n meat.n actually.r available.a ketchup.n mustard.n just.r ask.v really.r easy.a double.r_meat.n double.a_double.a cheese.n flying.n_dutchman.n

Fig. 2. Preprocessed Review Snippet: Original on Top; Segmented and POS-tagged on Bottom

An overview of the system can be seen in Figure 1. The input for the system are all reviews written for a product together with the rating assigned by the user to the product. Our hypothesis is that users who assign 5 stars (on a five point Likert scale) mainly talk about positive experience with the product or it's features. A review accompanied by a 1 star rating indicates a review with rather negative points. To not exclude the possibility that also in a 5 star review a minor negative point could be expressed we use a matrix allowing to smooth the assignment of reviews to rating classes. Especially a 3 star rating can contain negative as well as positive aspects which can be modeled using this matrix. Based on the topic models for each review, we then rank the reviews by minimizing the Kullback-Leibler divergence between the aggregated reviews of the ranking and three other distributions depending on the optimization strategy. After discussing the preprocessing step in the next section, we describe the modeling approach in Section IV and the ranking in Section V.

B. Preprocessing

Since reviews are user-generated they contain more grammatical errors, sloppy language, and spelling errors than more carefully written texts. Therefore, preprocessing the raw data becomes an important task. We used the Stanford POS Tagger [33] for tokenization and part-of-speech tagging. Then WordNet [34] was used to get the lemmas of the terms and remove all terms that are not verbs, adverbs, nouns, or adjectives.

Since uni-grams might not give an accurate picture of what a review is about we extract n-grams of variable lengths in the next step. Especially in the context of product reviews, multi-term phrases are important to model the data, e.g. "Microsoft Windows 7 Professional", "not recommended", or "graphic card". Therefore, we partition our data into meaningful n-grams first. Based on the work of Deligne and Bimbot [35], we compute multigram models for the documents in our corpus the following way: Each sentence is considered as a sequence of n-grams with variable length. The likelihood of a sentence is computed by summing up the individual likelihoods of the n-grams corresponding to each possible segmentation of the sentence. This is done using a Viterbi-like algorithm to find the maximum likelihood segmentation. In an iterative fashion, we re-estimate and update the probabilities until convergence. More details can be found in Bimbot et al. [36].

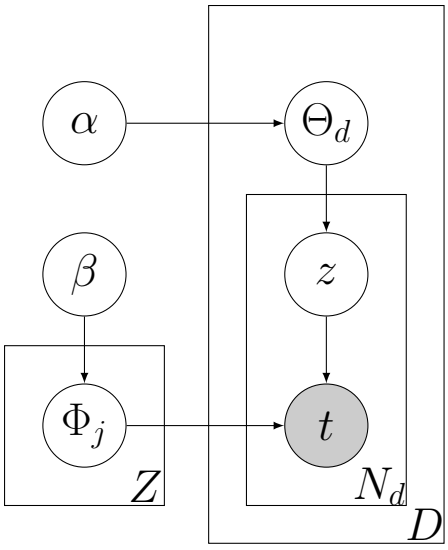


Fig. 3. Plate Notation for Latent Dirichlet Allocation

As a result, all documents in our corpus of product reviews are segmented into variable-length n-grams and the latent topics can now be based on n-grams or phrases instead of fixed sized units or single terms. Figure 2 shows a snippet of an original product review together with the preprocessed version without stop words but with part-of-speech information and some multi-grams.

IV. Modeling Reviews

To model the review data we make use of probabilistic topic models [37] to extract latent topics within the review corpus. We combine this information with the assigned star ratings for the reviews to cover positive and negative statements associated with a particular latent topic. In the following we describe LDA in more detail and in Section IV-B we explain how we combine the star ratings and the information about the extracted latent topics.

A. Finding Latent Topics

A product review usually covers different aspects or features of a product. For example, users have an opinion about the price of a product or the service of a company. Instead of a fine-grained extraction of features and sentiment, as done for instance by Bross and Ehrig [38], we rely on a statistical approach to find features or aspects.

To identify the latent topics we employ Latent Dirichlet Allocation [39], which models each review as a mixture of latent topics[2]. This probabilistic assignment of different topics to a single review allows later to identify topically similar reviews. Figure 3 shows the plate notation for Latent Dirichlet Allocation. LDA identifies a given number of $|Z|$ topics within a corpus of $|D|$ documents. Each term t in a review with N_d terms is associated with a topic z. Being the most important parameter for LDA, the number of latent topics $|Z|$ determines the granularity of the resulting topics, as we will see later. In order to find the latent topics, LDA relies on stochastic modeling.

[2]We use the LDA implementation in the Mallet library [40], which makes use of Gibbs sampling to compute the latent topics.

TABLE I
Top terms composing the latent topics "ticket" and "waiting" for America West Airlines

Term	Prob.	Term	Prob.
ticket.n	0.038	concourse.n	0.015
voucher.n	0.027	miss.v	0.015
clerk.n	0.016	take.v_off.r	0.015
care.v	0.011	hour.n_late.r	0.012
availability.n	0.008	change.n	0.009
complain.v	0.008	delay.n	0.009
look.v	0.008	flight.n_attendant.n	0.009
nightmare.n	0.008	meeting.n	0.009
suggest.v	0.008	not.r	0.009
america.n_worst.r	0.006	reggie.n	0.009

The modeling process of LDA can be described as determining a mixture of topics for each document in the corpus, i.e., $P(z \mid d)$, with each topic described by multigrams following another probability distribution, i.e., $P(w \mid z)$. This can be formalized as:

$$P(w_i \mid d) = \sum_{j=1}^{|Z|} P(w_i \mid z_j) P(z_j \mid d), \tag{1}$$

where $P(w_i \mid d)$ is the probability of the ith multigram for a given document d and z_i is the latent topic. $P(w_i \mid z_j)$ is the probability of w_i within topic z_j. $P(z_j \mid d)$ is the probability of picking a term from topic z_j in the document.

With LDA at hand, we are able to represent latent topics as a list of multigrams with a probability for each multigram indicating the membership degree within the topic. Furthermore, for each document in our corpus (reviews in our case) we can determine to which topics it belongs, also associated with a degree of membership (topic probability $P(z_j \mid d_i)$).

An example for two extracted latent topics represented by the top 10 terms is shown in Table I. Beside the terms also the probability for the terms belonging to the topic are shown. For this example we used $|Z| = 50$ latent topics.

B. Combining Latent Topics and Star Ratings

Each review d can now be modeled as a mixture of latent topics $P(z_i \mid d)$. Together with the rating of each review $r(d)$ we can transform the topic model into a topic-rating model by considering the topics for each rating class $r \in R = \{1, \ldots, 5\}$ separately:

$$P(z'_k \mid d) = \sum_{r \in R} m_{r(d)-1,r-1} * P(z_{k \bmod |Z|} \mid d), \tag{2}$$

where $k = \{0, \ldots, |R| * |Z|\}$ and $m_{i,j}$ an entry in the rating smoothing matrix:

$$M = \begin{pmatrix} 0.6 & 0.3 & 0.1 & 0.0 & 0.0 \\ 0.4 & 0.5 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.6 & 0.2 & 0.0 \\ 0.0 & 0.0 & 0.1 & 0.5 & 0.4 \\ 0.0 & 0.0 & 0.1 & 0.3 & 0.6 \end{pmatrix} \quad (3)$$

The matrix defines how likely it is that, e.g. a negative review contains neutral or positive aspects. This is also dependent on the dataset and the typical user rating behavior.

All latent topics extracted by LDA are now represented individually for each rating class. Each review is modeled as a topic mixture depending on its rating with some overlap according to the rating smoothing matrix. In the next section we describe how to compute the reference topic models to compute the different rankings corresponding to various strategies.

V. Ranking Reviews

Depending on the user's information need we define three ranking strategies:

1) Summary-focused Ranking (Section V-A)
2) Sentiment-focused Ranking (Section V-B)
3) Topic-focused Ranking (Section V-C)

To compute these rankings we take the topic-rating models of the reviews computed in the previous step and try to minimize the distance between the aggregated top-k reviews and a strategy-dependent target distribution. We use a greedy algorithm to find the best review for each position in the ranking.

As a measure for how well the top-k reviews approximate the corresponding target distribution we calculate the Kullback-Leibler divergence between the smoothed topic-rating models for the top-k results and for the target distributions. Kullback-Leibler divergence estimates the number of additional bits needed to encode the distribution U, using an optimal code for Q, and having a combined vocabulary size of $|Z'|$; in our case the number of latent topics $|Z|$ times the number of rating classes $|R|$.

$$D_{KL}(U||Q) = H(U;Q) - H(U) = \sum_{i=1}^{|Z'|} u_i * log_2(\frac{u_i}{q_i}) \quad (4)$$

In our setting, distribution Q is the combined topic-rating model of the top-k reviews and thus $D_{KL}(U||Q)$ can be directly used to measure the similarity with the target distribution.

A. Summary-focused Ranking

In most cases, users reading reviews are interested in getting an overview of the experiences of other users with the product. A ranking which gives a good overview summarizes the views expressed in all reviews. The goal for a review ranking system is therefore to approximate all reviews by the top-k in the ranking. Thus, the top-k reviews *summarize* the opinions about a product present in all reviews.

1	D_{KL}	A	B	C
	$A+B+C$	0.5	0.3	0.7
			Rank 1	
2	D_{KL}	$B+A$	–	$B+C$
	$A+B+C$	0.2	–	0.3
		Rank 2	Rank 1	
3	D_{KL}	–	–	$A+B+C$
	$A+B+C$	–	–	0.0
		Rank 2	Rank 1	Rank 3

Fig. 4. Example of the Greedy Algorithm to find a Ranking Summarizing the Three Reviews A,B, and C

With the topic-rating models computed for each review we try to find a ranking of reviews that approximates the aggregated topic-rating models of all reviews for a product. This means we try to minimize the Kullback-Leibler divergence between the top-k ranked reviews and the aggregation of all reviews.

To clarify the functioning of the greedy algorithm let's consider a product with three reviews represented by A, B, and C. We compute the aggregated topic-rating model $A+B+C$ and measure the Kullback-Leibler divergence D_{KL} for each position in the ranking. The example is shown in Figure 4.

B. Sentiment-focused Ranking

Instead of approximating all reviews, the sentiment-focused ranking tries to summarize only one particular class of ratings, for example negative aspects as represented by the topic-rating model with rating one. It could also be interesting to see which features of a product are discussed mainly in a neutral review or which aspects are only discussed in positive reviews. Depending on the rating smoothing matrix aspects from reviews having a slightly different rating can influence the ranking.

The target distribution that we try to approximate with the review ranking in this case is a (smoothed) uniform distribution over all topics for one rating. That means we get a diverse ranking covering all latent topics associated with a particular rating.

C. Topic-focused Ranking

Corresponding to the previous sentiment-focused ranking, we can focus the review ranking on a particular latent topic. This allows to get all opinions – positive, neutral, and negative – about a certain aspect. This might be useful for users who are interested in a particular feature of a product and the experience other users report in their reviews.

This type of ranking can be achieved by minimizing the Kullback-Leibler divergence of the reviews in the ranking and a (smoothed) uniform target distribution over all ratings for one topic. We refrained from evaluating this strategy due to the lack of large scale user data to test this type of ranking and the difficult mapping of all latent topics to well-defined product features.

TABLE II
DISTRIBUTION OF THE RATINGS FOR THE TEST PRODUCTS

Rating	Number of Posts for	
	"Pokemon"	"America West"
1.0	13	43
2.0	23	29
3.0	22	23
4.0	32	17
5.0	13	5

TABLE III
SAMPLE ANNOTATION FORM FOR "AMERICA WEST AIRLINES" REVIEWS

Feature/Aspect	Positive	Negative
"pricing"	X	
"seating/space"		
"food"		X
"customer service"		
"compliance with timetable"		
"gate changes"		
"luggage condition"		X
"frequent flyer program"	X	
"general aspects"		X

VI. EXPERIMENTS

To evaluate our system and the different rankings we adopt a method from information retrieval to judge rankings based on novelty and diversity. The ideal ranking would cover all different aspects and all different opinions about the aspects. The first review in the ranking should cover many aspects of the product to serve as a good overview. This can be compared to sub-topic retrieval where Web search engines try to find an optimal ranking to cover as many sub-topics as possible (see, for example, TREC 2009 Web Track, Diversity Task [41]). This evaluation approach requires annotated results, namely each review needs to be annotated with the sub-topics discussed in it. In the following we describe our dataset and the annotation of the test data.

A. Dataset

We crawled the Epinions[3] Web site to get around 30,000 reviews for 300 products. For the evaluation we needed to manually annotate the reviews with features of the product covered by the review and the polarity. Out of the 300 products we randomly picked two having not only positive or negative ratings: "America West Airline" and "Pokemon Snap for Nintendo 64". Table II shows the distribution of ratings for these two products in our corpus.

For manually annotating 200 reviews we first identified different features of the products. Table III shows the annotation form to annotate the reviews for "America West Airlines".

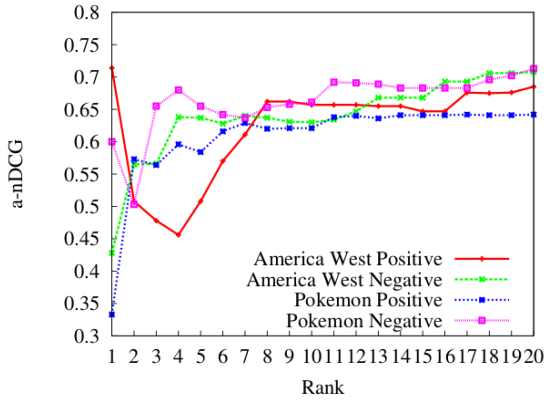


Fig. 5. Results for Sentiment-Focused Ranking Summarizing Only Positive or Negative Aspects Respectively

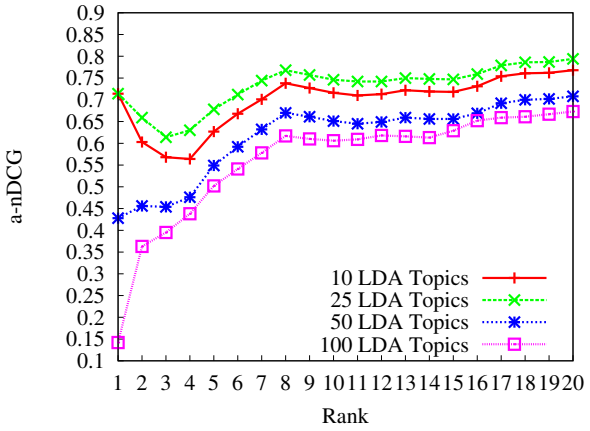
B. Results

The results for sentiment-focused ranking using 10 latent topics and focusing on either positive or negative aspects are shown in Figure 5. To compute the α-nDCG values we only considered the positive, respectively negative, manually annotated aspects to be relevant. As can be seen in the figure, summarizing the negative opinions with the top-k reviews in the ranking for "Pokemon' is easier than the positive opinions. For "America West" the negative aspects are quite well covered after the first four reviews whereas the coverage of positive opinions is at its minimum at this position in the ranking.

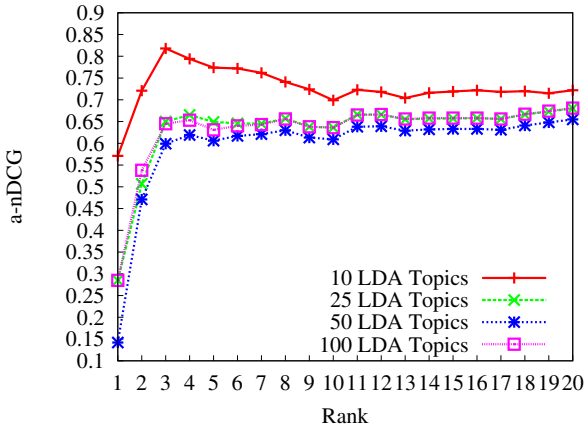
To evaluate the summarization-based ranking we computed α-nDCG [42] for our rankings using the manually annotated reviews to assess relevance and novelty. α-nDCG only accounts for positive and negative features and does not take different degrees of polarity into account in contrast to our optimization approach.

The results for the top-20 reviews using different numbers of latent topics are shown in Figure 6 ($\alpha = 0.5$). The best performance for "America Airline West" is achieved using 25 latent topics whereas for "Pokemon" 10 topics are the best choice. This can be explained by having a closer look at the individual reviews. The "Pokemon" reviews are considerably shorter and have clearly defined features. The airline has more features, longer reviews and are written in a more narrative style, e.g. "I had a 10 day vacation flying out of DC National on December 29, 2000 and spending 3 days in…". The "Pokemon" reviews on the other hand are more to the point: E.g. "This game is boring." or "This game is great in the beginning.". Finding the optimal number of latent topics to extract is an interesting research area by itself [43] and is worth analyzing in the context of product reviews. Figure 6 also shows that for "Pokemon" reading three reviews is already enough to get a good overview of the different opinions. For "America West", the user has to go through the first 8 reviews

[3]www.epinions.com

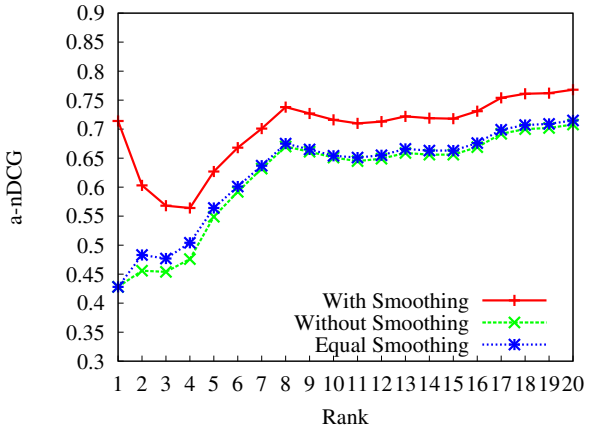


(a) α-nDCG Values for "America West Airlines"

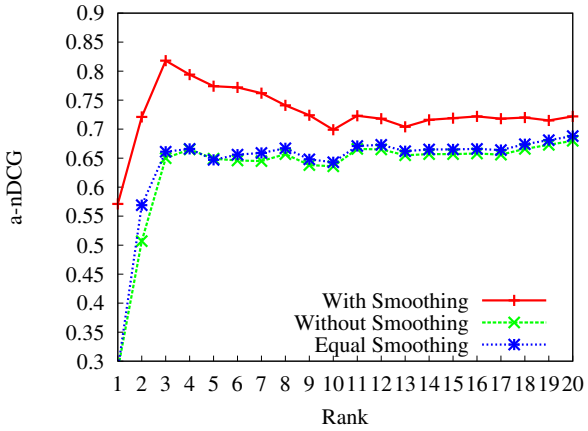


(b) α-nDCG Values for "Pokemon"

Fig. 6. Results for Different Numbers of Latent Topics ($\alpha = 0.5$)



(a) α-nDCG Values for "America West Airlines"



(b) α-nDCG Values for "Pokemon"

Fig. 7. Influence of the Rating Smoothing Matrix: Results for No Smoothing; Equally Weighting of Each Rating Class; Smoothing Between Rating Classes

to cover most of the aspects and opinions.

The influence of smoothing the rating by assigning a fuzzy membership degree for each review to the review classes is shown in Figure 7. "With smoothing" indicates the use of the rating smoothing matrix as depicted in Equation 3. "No smoothing" means that the different rating classes are strictly kept separately with a diagonal rating smoothing matrix $M = (m_{i,j})$ with $m_{i,j} = 1.0$ if $i = j$ and else $m_{i,j} = 0.0$. The results for using a rating smoothing matrix $M = (m_{i,j})$ with $m_{i,j} = 0.2, \forall i, j \in \{1, \ldots, |R|\}$ is labeled "Equal Smoothing". Different variations of the rating smoothing matrix could be necessary for different datasets depending for example on the skewness of the rating distribution over the classes or on individual user preferences.

VII. Conclusions & Future Work

We presented in this paper an approach to rank reviews for products based on latent topics and user-assigned ratings. The main goal was to summarize the opinions expressed in all reviews for a product in the top-k results of a ranking. In contrast to recommending single reviews we aimed at recommending an optimal diverse set of reviews using methods from information retrieval. We showed that diversified rankings of reviews allow users to grasp the overall opinions about a product faster and more reliable, thus unburden the user from having to read many reviews to get an overview. We investigated reviews from two products displaying different characteristics. Manual annotation of the reviews allowed an automatic evaluation of the proposed approach comparing different ranking strategies.

For future work we will investigate the possibility of personalizing the review rankings by taking personal preferences of users into account. For example, a user might be more interested in the battery life of a product than the screen size. Another interesting direction is analyzing and categorizing product reviews on a large scale to identify different types of reviews. As trust is a factor that directly affects the user confidence, we will also investigate the possibility to

incorporate trust between users and authors of reviews.

ACKNOWLEDGMENT

This work is partially supported by the European Union FET project LivingKnowledge (FP7–ICT–231126).

REFERENCES

[1] J. a. Chevalier and D. Mayzlin, "The effect of word of mouth on sales: Online book reviews," *Journal of Marketing Research*, vol. 43, no. 3, pp. 345–354, 2006.

[2] M. P. O'Mahony and B. Smyth, "Learning to recommend helpful hotel reviews," *Proceedings of the third ACM conference on Recommender systems - RecSys '09*, p. 305, 2009.

[3] S. Aciar, D. Zhang, S. Simoff, and J. Debenham, "Informed recommender: Basing recommendations on consumer product reviews," *Online*, no. June, pp. 39–47, 2007.

[4] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: predicting the usefulness and impact of reviews," in *Proceedings of the ninth international conference on Electronic commerce*, ser. ICEC '07. New York, NY, USA: ACM, 2007, pp. 303–310.

[5] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *Data Engineering Workshop*. IEEE, 2007, pp. 801–810.

[6] ——, "The effectiveness of personalized movie explanations: An experiment using commercial meta-data," in *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, 2008, pp. 204–213.

[7] M. O'Mahony, P. Cunningham, and B. Smyth, "An assessment of machine learning techniques for review recommendation," *Science*, 2009.

[8] S. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 423–430.

[9] Y. Liu, X. Huang, A. An, and X. Yu, "Modeling and predicting the helpfulness of online reviews," in *ICDM'08*. IEEE, 2009, pp. 443–452.

[10] F. Harper, D. Moy, and J. Konstan, "Facts or friends?: distinguishing informational and conversational questions in social q&a sites," in *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 759–768.

[11] W. Weerkamp and M. De Rijke, "Credibility improves topical blog post retrieval," *ACL-08: HLT*, pp. 923–931, 2008.

[12] J. Wiebe and E. Riloff, *Creating subjective and objective sentence classifiers from unannotated texts*, ser. Lecture Notes in Computer Science. Springer, 2005, vol. pages, no. 34063406, pp. 486–497.

[13] P. Chaovalit and L. Zhou, "Movie review mining: a comparison between supervised and unsupervised classification approaches," *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, vol. 00, no. C, pp. 112c–112c, 2005.

[14] K. Sparck Jones, "What might be in a summary," *Information Retrieval 93 Von der Modellierung zur Anwendung*, pp. 9–26, 1993.

[15] G. Salton, A. Singhal, C. Buckley, and M. Mitra, "Automatic text decomposition using text segments and text themes," *Proceedings of the the seventh ACM conference on Hypertext HYPERTEXT 96*, pp. 53–65, 1996.

[16] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," *Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06*, p. 43, 2006.

[17] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan, "An exploration of sentiment summarization," in *Proc. of AAAI*. Citeseer, 2003, pp. 12–15.

[18] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[19] A. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 339–346.

[20] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger, "Pulse: Mining customer opinions from free text," *Advances in Intelligent Data Analysis VI*, pp. 121–132, 2005.

[21] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 98*, vol. pp, pp. 335–336, 1998.

[22] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," *Proceedings of the 14th international conference on World Wide Web WWW 05*, p. 22, 2005.

[23] F. Radlinski and S. Dumais, "Improving personalized web search using result diversification," *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 06*, pp. 691–692, 2006.

[24] H. Chen and D. R. Karger, *Less is more: probabilistic models for retrieving fewer relevant documents*, ser. SIGIR '06. ACM, 2006, pp. 429–436.

[25] C. Zhai, W. W. Cohen, and J. Lafferty, *Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval*. ACM, 2003.

[26] M. Sanderson, J. Tang, T. Arni, and P. Clough, "What else is there? search diversity examined," *Advances in Information Retrieval*, vol. 5478, p. 562569, 2009.

[27] S. Gollapudi and A. Sharma, "An axiomatic approach for result diversification," in *Proceedings of the 18th international conference on World wide web WWW 09*, ser. WWW '09. ACM Press, 2009, p. 381.

[28] J. Wang and J. Zhu, "Portfolio theory of information retrieval," *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval SIGIR 09*, p. 115, 2009.

[29] D. Rafiei, K. Bharat, and A. Shukla, *Diversifying web search results*. ACM Press, 2010, p. 781.

[30] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," *Proceedings of the Second ACM International Conference on Web Search and Data Mining WSDM 09*, p. 5, 2009.

[31] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia, "Efficient computation of diverse query results," vol. 00, pp. 228–236, 2008.

[32] D. Schnitzer, A. Flexer, and G. Widmer, "A filter-and-refine indexing method for fast similarity search in millions of music tracks," in *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR09)*, 2009.

[33] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *In Proceedings of HLT-NAACL 2003*, 2003, pp. 252–259.

[34] C. Fellbaum, Ed., *WordNet An Electronic Lexical Database*. Cambridge, MA ; London: The MIT Press, May 1998.

[35] S. Deligne and F. Bimbot, "Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1, pp. 169–172, 1995.

[36] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal, "Variable-length sequence modeling: multigrams," *Signal Processing Letters, IEEE*, vol. 2, no. 6, pp. 111 –113, June 1995.

[37] M. Steyvers and T. Griffiths, *Probabilistic Topic Models*. Lawrence Erlbaum Associates, 2007.

[38] J. Bross and H. Ehrig, "Generating a context-aware sentiment lexicon for aspect-based product review mining," in *WI-IAT '10: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. Washington, DC, USA: IEEE Computer Society, August 31–September 3 2010.

[39] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, January 2003.

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

[41] C. L. A. Clarke, N. Craswell, and I. Soboroff, "Overview of the TREC 2009 web track," in *Proc. of TREC-2009*.

[42] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '08. New York, NY, USA: ACM, 2008, pp. 659–666.

[43] R. Arun, V. Suresh, C. Veni Madhavan, and M. Narasimha Murthy, "On finding the natural number of topics with latent dirichlet allocation: Some observations," in *Advances in Knowledge Discovery and Data Mining: 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings*, ser. Lecture Notes in Computer Science, M. Zaki, J. Yu, B. Ravindran, and V. Pudi, Eds., vol. 6118. Springer Berlin / Heidelberg, 2010, pp. 391–402.