Exploiting Social Debates for Opinion Ranking

Youssef Meguebli[1], Mouna Kacimi[2], Bich-liên Doan[1] and Fabrice Popineau[1]
[1]*SUPELEC Systems Sciences (E3S), Gif sur Yvette, France*
[2]*Free University of Bozen-Bolzano, Bozen-Bolzano, Italy*

Keywords: Opinion Ranking, Opinion Mining, Topic Aspects Extraction.

Abstract: The number of opinions in news media platforms is increasing dramatically with daily news hits, and people spending more and more time to discuss topics and share experiences. Such user generated content represents a promising source for improving the effectiveness of news articles recommendation and retrieval. However, the corpus of opinions is often large and noisy making it hard to find prominent content. In this paper, we tackle this problem by proposing a novel scoring model that ranks opinions based on their relevance and prominence. We define the prominence of an opinion using its relationships with other opinions. To this end, we (1) create a directed graph of opinions where each link represents the sentiment an opinion expresses about another opinion (2) propose a new variation of the PageRank algorithm that boosts the scores of opinions along links with positive sentiments and decreases them along links with negative sentiments. We have tested the effectiveness of our model through extensive experiments using three datasets crawled from `CNN`, `Independent`, and `The Telegraph` Web sites . The experiments show that our scoring model achieves high quality results.

1 INTRODUCTION

1.1 Motivation

Media platforms, like CNN[1] and ALjazeera[2], deliver the latest breaking news on various topics about everyday events. Moreover, they provide the possibility to write opinions about any published article and engage in discussions with other users. Typically, these opinions are unstructured making it hard to catch the flow of debates and to understand their main points of agreements and disagreements. Thus, there is a need for organizing users' opinions to (1) have a better understanding of the main issues related to each topic and (2) facilitate the participation to debates and thus increase the chance of acquiring new opinions. Figure 1 shows an example of how users' opinions could be organized. The opinions are organized based on their aspects, meaning the main points of the discussions around "Boston Bombing", such as *Immigration Reform*, *Border Control*, and *Employers Persecution*. The idea is that if a user cliques on an aspect (or searches for a new aspect), he gets all related opinions. Opinions are ranked based on their prominence

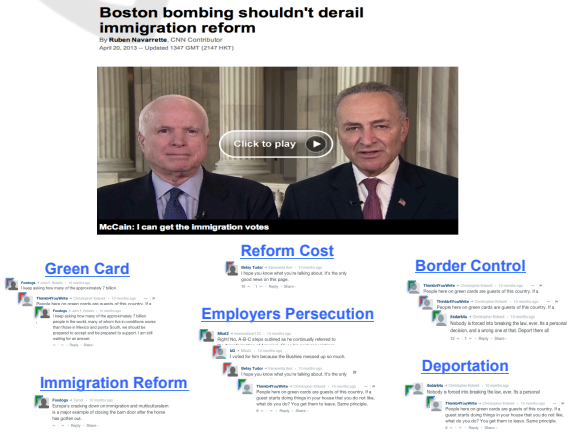

Figure 1: Illustration of users' opinions organization.

and are tagged with the color red if they have a negative sentiment, otherwise they are tagged with a green color, as shown in Figure 1. In this way we can have a better idea about the aspects and the sentiments expressed in users' debates.

1.2 Contribution

Our work aims at organizing user's opinions in news media platforms to facilitate their access, understand their trends, and provide a valuable source for en-

[1]http://www.cnn.com
[2]http://www.aljazeera.com/

Meguebli Y., Kacimi M., Doan B. and Popineau F..
Exploiting Social Debates for Opinion Ranking.
DOI: 10.5220/0005081702500260
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2014), pages 250-260
ISBN: 978-989-758-048-2

riching articles content. The result of this work can be useful for many applications including news recommendation, and the assessment of public opinion polls. However, this task can be very challenging since user generated content is a free source of information which can be subject to a lot of noise. Therefore, we focus on how to select high quality opinions about the different aspects of a given topic. The novel contribution by this paper has the following salient properties:

1. We propose a novel scoring model for opinions based on their relevance to a given topic aspect and their prominence. We define the prominence of an opinion based on how subject it is to replies and discussions, and the expertise of users reacting to it.
2. We model users' debates as a directed graph of opinions where links can be either positive or negative representing agreements and disagreements between opinions
3. We propose a new variation of the PageRank algorithm which handles both positive and negative links between graph nodes. The idea is to boost opinions scores along positive links and decrease them along negative links
4. We test our approach by running experiments on three datasets crawled from, `CNN`, `Independent`, and `The Telegraph` Web sites. The results show that our model achieves high quality results, particularly for highly popular and highly controversial topics having a large amount of user debates

Our proposed approach goes beyond existing opinion ranking techniques in several ways. First, determining prominent opinions about daily life topics is much more complex than identifying helpful product reviews as suggested in prior work (Hong et al., 2012; Kim et al., 2006; Liu et al., 2008; Tsur and Rappoport, 2009; Danescu-Niculescu-Mizil et al., 2009). Second, unlike existing approaches, we define user expertise not only based on explicit ratings, but also on implicit ratings the user gets for his actions. This is due to the fact that explicit ratings suffer different kind of bias (Liu et al., 2007) such as the winner circle bias, where opinions with many votes get more attention therefore accumulate votes disproportionately, and the early bird bias where the first opinion to be published tends to get more votes. Third, none of the existing approaches takes into account implicit ratings provided by users' debates and exchange of opinions. In our work, we take into account the relationships between opinions and their replies, which we call nested opinions, propagating the sentiments along those relations to compute the final score of an opinion.

2 RELATED WORK

Ranking opinions has received attention, in the past few years, driven by the need of automatic annotation of product reviews. The proposed approaches focus on how to find helpful product reviews (Hong et al., 2012; Kim et al., 2006; Liu et al., 2008; Tsur and Rappoport, 2009; Danescu-Niculescu-Mizil et al., 2009). These approaches assign a helpfulness score to each review, based on past interactions in the system, and return to the user a ranked list of reviews. Different parameters have been exploited to rank reviews. Kim et. al., (Kim et al., 2006) exploit the multitude of user-rated reviews on Amazon.com, and train an SVM regression system to learn a helpfulness function. This helpfulness function is then applied to rank unlabeled reviews. Danescu et. al., (Danescu-Niculescu-Mizil et al., 2009) show, through extensive experiments, that social effect is a significant factor for measuring helpfulness. The social effect is based on the relationship of one user's opinion to the opinions expressed by others in the same setting. More precisely, the relationship of a review's star rating to the star ratings of other reviews for the same product. Tsur et. al., (Tsur and Rappoport, 2009) identify a lexicon of dominant terms that constitutes the core of a virtual optimal review. This lexicon defines a feature vector representation. Reviews are then converted to this representation and ranked according to their distance from a "virtual core" review vector. Liu et. al., (Liu et al., 2008) show that the helpfulness of a review depends on three factors: the reviewer's expertise, the writing style of the review, and the timeliness of the review. Based on those features, they propose a nonlinear regression model for helpfulness prediction. Hong et. al., (Hong et al., 2012) start from the assumption that user preferences are more explicit clues to infer the opinions of users on the review helpfulness. Thus, they employ user-preferences based features including information need, credibility of the review, and mainstream opinions.

The approaches described above use different features to define the helpfulness of a review ranging from its content and the expertise of its author to the preferences of users. However none of them takes into account the relationships between the reviews, meaning the debates that users engage into to discuss a given product. In our work, we take into account the relations between opinions and all the reactions they got from users including implicit and explicit feedbacks. Then, we propagate the sentiments along those relations to compute the final score of an opinion. Additionally, unlike the approaches described above, we define user expertise from the implicit ratings he gets

for his actions.

Another research problem that is directly related to our work is *opinion mining*, also called *sentiment analysis*. This problem has been studied in the past few years (Dave et al., 2003; Wilson et al., 2004; Ding et al., 2008) exploiting two main directions: (1) finding product features commented on by reviewers and (2) deciding whether the comments are positive or negative.

As stated in (Ding et al., 2008), most classification methods follow two approaches: (1) *corpus-based* approaches, and (2) *lexicon-based* approaches. Corpus-based approaches find co-occurrence patterns of words to determine the sentiments of words or phrases, e.g., the works in (Turney, 2002). Lexicon-based approaches use synonyms and antonyms in WordNet to determine word sentiments based on a set of seed opinion words (Ding et al., 2008; Dragut et al., 2010). Some of these approaches perform sentiment classification at *document level* (Dave et al., 2003; Lin and He, 2009; He, 2010; Bespalov et al., 2011; Gao and Li, 2011; Lin et al., 2011; Amiri and Chua, 2012) where a sentiment orientation is assigned to the whole document. In contrast, other approaches address sentiment classification at *sentence level* (Ding et al., 2008; Jia et al., 2009) so various sentiments can appear within the same document.

The opinion mining techniques described above focus on how to classify opinions depending on their sentiments and most of them use product reviews as test cases. In our work, we adopt a document-level classification using a recent open source api and apply it on opinions about general topics.

3 DEBATE-BASED SCORING MODEL

We consider a query $Q(u, q_1...q_n)$, issued by a query initiator u, as a set of keywords $q_1...q_n$ that describe one or several aspects related to a given news article. The goal is to retrieve high quality opinions that satisfy the user query. Result opinions should contain at least one of the query terms and be ranked according to a query-specific opinion score. Additionally, we propose to boost or decrease the score of an opinion based on the *reactions* of users to it. Users often start debates around a given opinion by providing feedbacks, supportive opinions, opposing opinions, or complementary ones. We capture the impact of these reactions around the opinion by introducing the concept of *prominence*. Both *relevance* and *prominence* scores are used to rank opinions that best match the user query. Formally, we define the score of an opinion O about a news article, given a query Q, as follows:

$$Score(O,Q) = \alpha Rel(O,Q) + (1-\alpha)Pro(O)$$

where $Rel(O,Q)$ reflects the relevance of opinion O to query Q, $Pro(O)$ reflects the prominence of opinion O, and α is a parameter used to balance the two components of the model.

3.1 Opinion Relevance

To compute the relevance of an opinion to user query about a news article A, we use BM25 (or Okapi) scoring function given by:

$$BM25(O,q_i) = IDF(q_i)\frac{f(q_i,O).(k_1+1)}{f(q_i,O)+k_1.(1-b+b.\frac{|O|}{avgol})}$$

Where $f(q_i,O)$ is the count of term q_i in opinion O, $|O|$ is the length of opinion O, $avgol$ is the average opinion length in the collection of opinions about news article A, $k_1 = 1.2$ and $b = 0.75$. $IDF(q_i)$ is the inverse document frequency weight of the query term q_i which is computed as:

$$IDF(q_i) = log\frac{N_e - n(q_i) + 0.5}{n(q_i) + 0.5}$$

where N_e is the total number of opinions about a news article A, and $n(q_i)$ is the number of opinions about a news article A containing term q_i. Thus, the relevance score of an opinion is given by:

$$Rel(O,Q) = \sum_{i=1}^{n} BM25(O,q_i)$$

3.2 Opinion Prominence

An opinion might trigger reactions in the news platform and thus becomes the starting point of a debate. We call this kind of opinions *seed opinions*. A seed opinion can get replies from other users, then these replies get other replies and so on forming a debate. We call an opinion replying to another opinion a *nested opinion*. Based on these patterns, we model the structure of a debate as a graph of opinions. More specifically, we use a directed tree where the root represents a seed opinion. Each non root node is a nested opinion that replies to its parent. Leaf nodes are nested opinions that do not get any reply. Figure 2 shows an example of a debate structure. Edges are directed from children to parents where each link can be either positive or negative reflecting the sentiment the child expresses for its parent. Note that

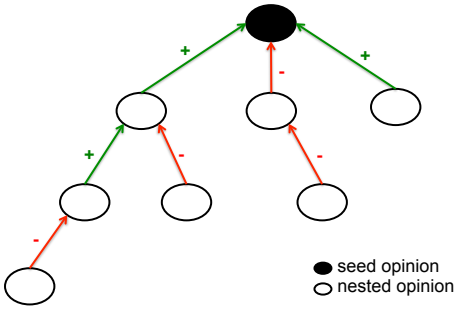

Figure 2: Debate Graph.

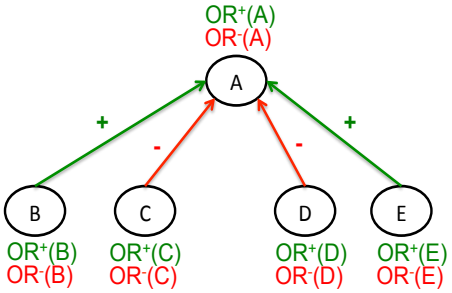

Figure 3: Example of OpinionRank.

to get information about the sentiment orientation of nested opinions we have used Alchemy API[3]. Using the debate graph, we compute the prominence of each opinion based on the number and quality of its incoming links. The underlying assumption is that prominent opinions are likely to receive many positive links from other opinions while less prominent ones are more likely (i) to receive more negative links or (ii) not to receive any reaction. To this end, we adopt the PageRank Algorithm to compute the prominence scores of seed opinions as described in the next section. Note that nested opinions do not take part of the results because considering them as independent components risk to be meaningless. A nested opinion is answering another opinion, so getting it as a single result would be like looking at a part of a discussion without knowing why it started and what is exactly about. Thus, we return to the user only seed opinions since they are certainly self contained, and we use the nested opinions to compute the final score of their related seed opinions. When the user is interested in a seed opinion, then he can click on it to have access to the debate that includes all related nested opinions.

4 OPINIONRANK ALGORITHM

OpinionRank adopts the same principle of PageRank Algorithm that models user behavior in a hyperlink graph, where a random surfer visits a web page with a certain probability based on the page's PageRank. The probability that the random surfer clicks on one link is solely given by the number of links on that page. So, the probability for the random surfer reaching one page is the sum of probabilities for the random surfer following links to this page. It is assumed that the surfer does not click on an infinite number of links, but gets bored sometimes and jumps to another page at random. Formally, the PageRank algorithm is given by:

[3]http://www.alchemyapi.com/api/

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)}\right)$$

where $PR(A)$ is the PageRank of page A, $PR(T_i)$ is the PageRank of pages T_i which link to page A, $C(T_i)$ is the number of outgoing links of page T_i, and d is a damping factor which can be set between 0 and 1. As we can see, The PageRank of page A is recursively defined by the PageRanks of pages which link to it. The PageRank of a page T is always weighted by the number of its outgoing links. The weighted PageRank of pages T_i is then added up. Finally, the sum of the weighted PageRanks of all pages T_i is multiplied with a damping factor d which reflects the probability for the random surfer not stopping to click on links.

We propose OpinionRank which adapts the PageRank algorithm to the requirement of our approach. Looking at the debate graph, we note that an opinion has only one outgoing link because it answers exactly one opinion. Thus, all $C(T_i)$ are set to 1. Additionally, links between opinions can reflect either positive or negative sentiments showing an agreement or a disagreement between opinions. Thus, a positive incoming link for page A should increase $A's$ PageRank, while a negative incoming link should decrease $A's$ PageRank. However, including subtractions will violate the properties of the probability distribution and give non trivial interpretation for the behavior of the random surfer. Thus, we propose to compute two OpinionRank scores for each opinion A: (1) a score that reflects the probability that the surfer reaches A following positive sentiments for A and (2) a score that reflects the probability that the surfer reaches A following negative sentiments for A. Formally, we define the OpinionRank Algorithm as follows:

$$OR^+(A) = (1-d) + d\left(\sum_{i=1}^{k} OR^+(P_i) + \sum_{j=1}^{m} OR^-(N_j)\right)$$

and

$$OR^-(A) = (1-d) + d\left(\sum_{i=1}^{k} OR^-(P_i) + \sum_{j=1}^{m} OR^+(N_j)\right)$$

where $OR^+(P_i)$ and $OR^-(P_i)$ are the OpinionRanks of opinions P_i which have a positive link to A. Similarly, $OR^+(N_j)$ and $OR^-(N_j)$ are the OpinionRanks of opinions N_i which have a negative link to A. $OR^+(A)$ reflects the probability of reaching A following positive sentiments and $OR^-(A)$ reflects the probability of reaching A following negative sentiments. As shown in figure 3, reaching A can be done via opinions B and E that agrees with A or via opinions C and D that disagrees with A. The intuition is that what agrees with B and E consequently agrees with A and what disagrees with C and D consequently agrees with A. Thus, $OR^+(A)$ is computed as the sum of $OR^+(B)$, $OR^+(E)$, $OR^-(B)$, $OR^-(E)$. Similarly, what disagrees with B and E consequently disagrees with A and what agrees with C and D consequently disagrees with A. Thus, $OR^-(A)$ is computed as the sum of $OR^-(B)$, $OR^-(E)$, $OR^+(B)$, $OR^+(E)$.

Typically, PageRank assumes a probability distribution between 0 and 1. Hence, the initial value for the score of each page is $\frac{1}{N}$ where N is the total number of pages in the graph. In our setting, we assume a non uniform probability distribution where the initial score of each opinion is a function of the number of feedbacks it receives from users. The intuition is to boost opinions receiving positive feedbacks and penalize those receiving negative feedbacks. In news media platforms, an opinion can receive two kinds of feedbacks: *like* and *dislike*. Thus, for each opinion O_i we set the OpinionRank score for positive sentiments $OR^+(O_i) = \frac{L_i}{F}$ and the OpinionRank score for negative sentiments $OR^-(O_i) = \frac{D_i}{F}$ where L_i and D_i are the number of likes and dislikes for opinion O_i, and F is the total number of feedbacks for all opinions about the news article of interest. By walking through the debate graph the dislikes of an opinion can be transitively considered as likes for an opposite opinion and vice versa. Thus, we divide by the total number of feedbacks to avoid that probabilities go beyond 1.

When we compute for an opinion O the OpinionRank for positive sentiments and the OpinionRank for negative sentiments, its prominence score can be computed by:

$$Pro(O) = OR^+(O) - OR^-(O)$$

The prominence score gives more importance to opinions receiving many positive reactions and few negative reactions.

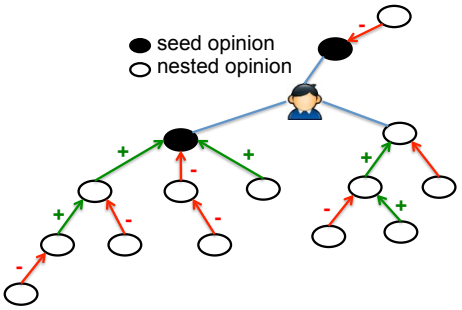

Figure 4: User Graph for a given Topic T.

5 USER-SENSTIVE OPINIONRANK

We propose an extension to the OpinionRank Algorithm that weights the impact of an opinion based on the confidence of its provider. For a given news article of topic T, the intuition is to give more importance to opinions provided by users with high confidence in topic T. To this end, at the initialization step, for each opinion O related to topic T we multiply its OpinionRank scores by the confidence of its provider on topic T. The key question here is how to compute user confidence. The confidence represents the expertise of a user on a given topic. Computing user confidence in social networks have been addressed in many studies (e.g. (Akram Al-Kouz, 2011; Zhang et al., 2007a)) based on different measures including posts content, groups, and relationships between users. However these techniques have specific assumptions to their applications which make them hard to adapt in our work. For this reason, we use a simple and intuitive way of computing user confidence based on the reactions the user receives for his provided contents. A user can provide different types of content including seed opinions, nested opinions, or feedbacks. Each of the seed and nested opinions belongs to a given topic and might receive reactions from other users. We represent the actions and the reactions a user performs within a given topic T by the graph shown in figure 4. To compute the confidence of the user, we first compute the prominence score for each of its provided opinions. Then the confidence of the user is computed as the sum of the prominence scores of all the opinions he provided within topic T. Formally, user confidence is given by:

$$C_{U,T} = \frac{\sum_{i=1}^{n} Pro(O_i)}{Max_{C_T}}$$

Note that user confidence is normalized over the maximum user confidence value Max_{C_T} in Topic T.

Table 1: Datasets Statistics.

	#News articles	#Users	#Seed Opinions	#Nested Opinions	#Feedbacks
CNN	40, 334	753, 185	12, 516, 409	23, 389, 867	80, 585, 030
Telegraph	40, 136	151, 813	7, 096, 741	11, 822, 323	122, 895, 681
Independent	10, 408	62, 171	747, 665	1, 411, 996	14, 445, 661

6 EXPERIMENTS

6.1 Experimental Setup

6.1.1 Datasets

We have crawled three datasets of Web News from `CNN`[4], `The Telegraph`[5], and `Independent`[6], which have a social service allowing users to communicate, discuss around topics, and perform a variety of rating actions. The choice of these datasets was based on their rich content of opinions and the possibility to get information about all actions and feedbacks of users allowing us to have a complete implementation of our model and validate our approach. Note that we could not implement our approach on any of the opinion datasets available in the literature, such as TREC datasets (Gerani et al., 2010; Zhang et al., 2007b; Santos et al., 2009), due to their lack of information about user reactions and the relations between direct opinions and nested opinions. We have crawled 40,334 articles from `CNN`, 40,136 articles from `The Telegraph`, and 10,408 from `Independent`. We have extracted all direct opinions related to these articles together with their nested opinions, and feedbacks. For each user who provided opinions, we have extracted his activities and the feedbacks he received for them. More statistics about these datasets are shown in Table 1.

6.1.2 Baselines

We used two baselines from the literature to assess the effectiveness of our approach. As a first baseline, we choose BM25 (or Okapi) scoring function to compute the score of each opinion, and most importantly to highlight the impact of prominence score. As a second baseline, we use the RevRank technique (Tsur and Rappoport, 2009), an opinion ranking model. The idea of this work is to use the dominant terms as indicators for the key-concepts with respect to a specific news article, in order to compute a helpfulness score for each opinion. For example, the terms *election* or *Obama* are usually very frequent in the opinions about a video of *Obama presidential campaign*. However their contribution to the helpfulness of an opinion is limited as they do not provide the user any new information or any new insights beyond the most trivial. On the other hand, terms like *foreign policy* and *government* are not as frequent but are potentially important, therefore the scoring algorithm should allow them to gain a dominance score. The process of identifying dominant terms is done in two stages. First we compute, for each news article, the frequency of all terms that appear in its related opinions. Each term is scored by its frequency, thus frequent terms are considered more dominant than others. Second, we re-rank the resulting terms by their frequency in the British National Corpus (BNC). RevRank technique (Tsur and Rappoport, 2009) is independent from the query, thus relevant opinions might end up having a low rank if the query terms are unimportant in the BNC corpus. For this reason, we have excluded the query terms from the process of defining dominant terms. In fact, for each news article a and query term q_i, we select the d most dominant terms to define a feature vector representation of opinions containing term q_i. We refer to the feature vector having 1 in all of its coordinates as the core vector (CV_i) related to the query q_i. Each opinion O of news article e containing the query term q_i is mapped to V_O, a feature vector representation such that a coordinate k is 1 or 0 depending on whether or not the opinion O contains the k^{th} dominant term. Based on the feature vector representation of opinions and CV_i, we define the helpfulness score of an opinion O as follows:

$$Help(O,q_i) = b(O,q_i) \times \frac{V_O.CV_i}{p(|O|) \times |O|}$$

Where $b(O,q_i)$ equals to 1 if the opinion O contains the query term q_i and 0 otherwise, $V_O.CV_i$ is the dot product of the representation vector of opinion O and CV_i, $|O|$ is the length of the opinion O, and $p(|r|)$ is a penalization factor equals to f[7] if $|O| < |\overline{O}|$ and 1 otherwise. The penalization factor f is needed to penalize opinions that are too short while the penalization for an excessive length is already given by the denominator $|O|$.

[4]http://www.cnn.com/

[5]http://www.telegraph.co.uk/

[6]http://www.independent.co.uk/

[7]We have experimentally chosen $f = 3$

Table 2: Precision and NDCG values per DATASET

		P@10	P@20	NDCG@10	NDCG@20
CNN	*ORel*	0.610	0.624	0.816	0.798
	RevRank	0.708	0.645	**0.844**	0.797
	Rel+ Pro	0.771	0.737	0.800	0.799
	Rel+ Pro (Conf)	**0.801**	**0.772**	0.836	**0.832**
Telegraph	*ORel*	0.665	0.674	0.811	0.804
	RevRank	0.789	0.704	0.862	0.844
	Rel+ Pro	0.839	0.807	0.858	0.848
	Rel+ Pro (Conf)	**0.851**	**0.835**	**0.870**	**0.858**
Independent	*ORel*	0.694	0.652	0.843	0.832
	RevRank	0.773	0.710	**0.879**	**0.866**
	Rel+ Pro	0.794	0.760	0.849	0.825
	Rel+ Pro (Conf)	**0.805**	**0.778**	0.854	0.831

6.1.3 Evaluation Metrics

To compare the results of the different methods, we use two quality measures: Precision at k ($P@k$) and Normalized Discounted Cumulative Gain (NDCG). The $P@k$ is the fraction of retrieved opinions that are relevant to the query considering only the top-k results. It is given by:

$$P@k = \frac{|Relevant\_Opinions \cap topk\_Opinions\_Results|}{k}$$

Additionally, we compute *NDCG* to measure the usefulness (gain) of opinions based on their (geometrically weighted) positions in the result list.

$$NDCG(E,k) = Z_k \sum_{i=1}^{k} \frac{2^{rel(i)} - 1}{log_2(1+i)}$$

where Z_k is a normalization factor calculated to make *NDCG* at k equal to 1 in case of perfect ranking, and $rel(i)$ is the relevance score of an opinion at rank i. In our setting, relevance scores $rel(i)$ have three different values: 2(very relevant), 1(relevant), and 0(not relevant).

6.2 Strategies Under Comparison

We evaluate the effectiveness of our scoring model by using different strategies. For each news article, we rank its related opinions using the following strategies:

Relevance (Rel). Results are ranked using the BM25 scoring as described in section 3.1.

RevRank. Results are ranked based on RevRank technique as described in section 6.1.2.

OpinionRank (Rel+Pro). Results are ranked based on relevance and prominence computed using the OpinionRank algorithm described in section 4.

User-Sensitive OpinionRank (Rel+Pro(Conf)). Results are ranked based relevance and prominence computed using the User-Sensitive OpinionRank algorithm described in section 5

6.2.1 Setup

Finding a good set of queries is not an easy task since users might be interested in searching opinions on different aspects of a given news article, depending on their personal context and interests. Thus, we have conducted a user study with manual query selection and assessment. The task was carried out by 30 human assessors who where researchers and students not involved in this project. We have asked our human assessors to choose news articles of interests and suggest queries related to them according to their interests. This process resulted in 206 queries posed on 206 topics from the three datasets. More precisely, we have tested 108 queries on `CNN`, 70 queries on `The Telegraph` and 28 queries on `Independent`. For each query, we have applied the strategies described earlier and got the top 20 results for each strategy. We have shown the pool of all results to our human assessors who evaluated them according to the following guidelines: (1) an opinion is considered non relevant, and gets a score of 0, if it does not give a comment related to the query, (2) an opinion is considered relevant if it contains information about the query. In this case it gets a score of 1, (3) an opinion is considered very relevant if it is relevant to the query and provides additional information that was not given by the news article itself such as new view point, new arguments, or references to more information about the query topic. In this case it gets a score of 2. The assessment is done without having any idea about the adopted strategy.

Table 3: Precision and NDCG values for Relevance-based Ranking per category.

		Precision		NDCG	
		P@10	**P@20**	**NDCG@10**	**NDCG@20**
Business	*Rel*	0.612	0.625	0.798	0.790
	RevRank	0.768	0.656	**0.907**	0.890
	Rel+ Pro	0.862	0.815	0.892	0.889
	Rel+Pro (Conf)	**0.875**	**0.837**	0.902	**0.895**
Media	*Rel*	0.587	0.556	**0.857**	0.813
	RevRank	0.668	0.571	0.849	**0.828**
	Rel+ Pro	0.687	0.646	0.761	0.750
	Rel+ Pro (Conf)	**0.737**	**0.696**	0.832	0.813
Living	*Rel*	0.742	0.725	0.812	0.813
	RevRank	0.814	0.760	0.852	0.836
	Rel+ Pro	0.814	0.817	0.858	0.845
	Rel+ Pro (Conf)	**0.821**	**0.835**	**0.863**	**0.861**
Opinion	*Rel*	0.621	0.683	0.797	0.794
	RevRank	0.757	0.681	0.859	0.830
	Rel+ Pro	0.866	0.821	0.860	0.853
	Rel+ Pro (Conf)	**0.872**	**0.846**	**0.869**	**0.862**
Politics	*Rel*	0.667	0.645	0.789	0.772
	RevRank	0.739	0.611	0.845	0.747
	Rel+ Pro	0.791	0.746	0.812	**0.805**
	Rel+ Pro (Conf)	**0.803**	**0.775**	**0.861**	0.803

6.3 Results and Analysis

6.3.1 Results

The overall results on the three datasets are shown in Table 2. We can see that our approach almost always outperforms the baselines by an improvement of $3-13\%$ in terms of precision, and $2-4\%$ in terms of NDCG. This shows that the prominence component of the model plays an important role in improving the satisfaction of users. The datasets CNN, and The Telegraph have very similar performances while Independent is slightly worse regarding *NDCG* values. The reason is that Independent dataset does not contain a lot of user reactions, which makes the prominence component weaker. Additionally, it is the dataset with less queries which makes it very sensitive to outliers. It is also observed that User-sensitive OpinionRank improves the effectiveness of OpinionRank algorithm. Therefore, including user confidence to compute prominence scores gives better results for opinion ranking. One explanation is that opinions given by experts have more relevance and impact than opinions given by novice users. To have a more insightful analysis, we looked at the topic of news articles for 193 queries falling into 5 categories: Business (33 queries), Media (32 queries), Living (34 queries), Opinions (40 queries), and Politics (54 queries). Our results by category are shown in table 3. For all categories, our strategy almost always improves the two measures of Precision and NDCG. It is also noteworthy to say that the gain varies with the topic of the news article. For example, in the *Politics* category the Precision@10 increases from 73.9% using RevRank technique to 80.3% using our User-sensitive opinionRank model and the NDCG@10 improves from 84.5% to 86.1%. By contrast, for the Living category, the absolute improvement is of 0.8% in Precision@10, and 1.1% in NDCG@10.

The average of precision values, for all categories, shows a gain between 0.8% to 18%. For *NDCG*, we have a slight improvement as compared to the RevRank baseline. However, the gain can be much higher for many individual queries. Examples are shown in table 4. For instance, considering the query *Osama bin laden death*, the precision was raised from 50% using RevRank technique to 100% using our User-sensitive OpinionRank approach, and the NDCG from 73.9% to 98.1%. Similarly, we improve the precision of the query *US gun control and suicide* from 60% to 100%, and the NDCG from 85.8% to 92.2%, giving high quality results.

To have a concrete idea about the results of our approach, we take our motivation example of the news article Boston Bombing and we retrieve the top5 opinions about the topic by using (1) relevance only and (2) relevance and prominence. We can see, in Figure 5 that both result lists are relevant, however using

Table 4: Individual Precision and NDCG values for Relevance-based and Insightfulness-based Ranking.

	P@10				NDCG@10			
	Rel	RevRank	Rel+ Pro	Rel+ Pro(Conf)	Rel	RevRank	Rel+ Pro	Rel+ Pro(Conf)
Titan workers problem	0.65	0.65	0.85	**0.95**	0.681	0.805	0.977	**0.988**
Italy election Europolitcs	0.6	0.7	0.9	**0.95**	0.816	0.939	0.979	**0.985**
School overspend	0.55	0.6	0.55	**0.65**	0.756	0.823	0.780	**0.882**
Kevin Hart arrest	0.4	0.45	0.75	**0.85**	0.808	0.825	0.746	**0.981**
Rennard sexual scandal	0.45	0.55	0.8	**0.85**	0.749	0.708	0.788	**0.785**
antibiotics resistance	0.75	0.8	0.9	**0.95**	0.876	0.961	0.955	**0.962**
Osama bin laden death	0.45	0.5	0.9	**1.0**	0.733	0.739	0.960	**0.981**
Gun control and suicide	0.6	0.6	0.7	**1.0**	0.779	0.858	0.637	**0.922**
Job plans US candidates	0.65	0.65	0.8	**0.85**	0.619	0.776	0.935	**0.942**
Russian Chechen War	0.6	0.6	0.7	**0.85**	0.695	0.673	0.767	**0.920**

the prominence score returns more opinions that bring new insights about the topic which is clearly an added value.

6.3.2 Discussion

The experimental results show that our model almost always outperforms the native rankings of opinions by a significant margin. In some cases, however, the gains are small and generally depend on the category type. One explanation to this behavior is that topics of *Business*, *Opinion*, and *Politics* are usually very popular, gossip appealing, controversial, and daily life subjects. Examples include, *Gun control and suicide*, *Kevin hart arrest*, and *Italian election Euro-politics*. Due to the large number of opinions and discussions in these categories, the prominence score improves the overall performance of our model. By contrast, the other categories, such as *Living*, generally contain topics that are not very controversial and consequently generate less debates and discussion between users (e.g.,*School academies overspend*). Consequently, our model is less effective in some categories, which is due to their unpopularity. A closer look at Table 4 shows that we are performing particularly well for news articles about highly controversial and highly popular topics, which are subject to gossiping. For instance, we can clearly see the difference between the query about the *Gun control and suicide*, a highly controversial topic, and the *School academies overspend* where opinions are less diverse. The precision improves for the first query from 60% using RevRank technique to 100% using User-sensitive OpinionRank technique, while there is a slight improvement for the second query.

To summarize, our model works best for topics with very large number of opinions and debates which is a very promising step towards our initial goal of selecting valuable information from the increasing amount of opinions in news media. It is also clear that our model does not perform well with categories and news articles related to unpopular topics. To cope with that, one solution could be to determine the prominence of opinions based on how many opinions are similar to it. This means, we create a graph of opinions where a link exist between two opinions if they are similar. In this way, an opinion with many incoming links will be prominent.

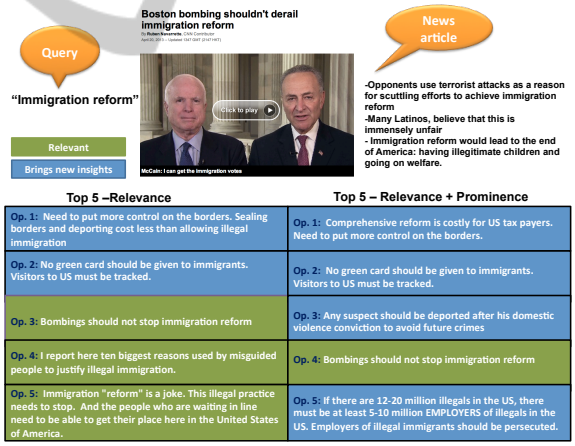

Figure 5: An Example of Opinion Ranking for immigration reform.

7 CONCLUSIONS

Retrieval and ranking of opinions in product reviews has received great attention in prior works. In this paper, we generalized this problem to retrieving and ranking opinions in news media, and paid particular attention to the exploitation of users' debates in such platforms to retrieve the most prominent opinions. Our experiments showed that these debates,

enhanced by explicit feedbacks, are definitely valuable and should be taken into account for ranking opinions. For pragmatic reasons, our experiments included news datasets having similar structures. However, exploring other datasets of different types of entities, of users, and kinds of opinions is worthwhile in order to show the wide applicability of our model. To this end, we are planning to assess the effectiveness of our approach using a dataset crawled from Youtube, which is more subject to noise. We are currently investigating these points for further improvements.

ACKNOWLEDGEMENTS

This work was supported by RARE project.

REFERENCES

Akram Al-Kouz, Ernesto William De Luca, S. A. (2011). Latent semantic social graph model for expert discovery in facebook. In *11th International Conference on Innovative Internet Community Systems*, page 269. GI Edition.

Amiri, H. and Chua, T.-S. (2012). Mining sentiment terminology through time. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 2060–2064, New York, NY, USA. ACM.

Bespalov, D., Bai, B., Qi, Y., and Shokoufandeh, A. (2011). Sentiment classification based on supervised latent n-gram analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 375–382, New York, NY, USA. ACM.

Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J., and Lee, L. (2009). How opinions are received by online communities: a case study on amazon.com helpfulness votes. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 141–150, New York, NY, USA. ACM.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528.

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *WSDM*, pages 231–240.

Dragut, E. C., Yu, C., Sistla, P., and Meng, W. (2010). Construction of a sentimental word dictionary. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1761–1764, New York, NY, USA. ACM.

Gao, S. and Li, H. (2011). A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1047–1052, New York, NY, USA. ACM.

Gerani, S., Carman, M. J., and Crestani, F. (2010). Proximity-based opinion retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 403–410, New York, NY, USA. ACM.

He, Y. (2010). Learning sentiment classification model from labeled features. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1685–1688, New York, NY, USA. ACM.

Hong, Y., Lu, J., Yao, J., Zhu, Q., and Zhou, G. (2012). What reviews are satisfactory: novel features for automatic helpfulness voting. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 495–504. ACM.

Jia, L., Yu, C., and Meng, W. (2009). The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1827–1830, New York, NY, USA. ACM.

Kim, S., Pantel, P., Chklovski, T., and Pennacchiotti, M. (2006). Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 423–430.

Lin, C. and He, Y. (2009). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 375–384, New York, NY, USA. ACM.

Lin, Z., Tan, S., and Cheng, X. (2011). Language-independent sentiment classification using three common words. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1041–1046, New York, NY, USA. ACM.

Liu, J., Cao, Y., Lin, C.-Y., Huang, Y., and Zhou, M. (2007). Low-quality product review detection in opinion summarization. In *EMNLP-CoNLL*, pages 334–342.

Liu, Y., Huang, X., An, A., and Yu, X. (2008). Modeling and predicting the helpfulness of online reviews. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 443–452. IEEE.

Santos, R. L. T., He, B., Macdonald, C., and Ounis, I. (2009). Integrating proximity to subjective sentences for blog opinion retrieval. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 325–336, Berlin, Heidelberg. Springer-Verlag.

Tsur, O. and Rappoport, A. (2009). Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *International AAAI Conference on Weblogs and Social Media*.

Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424.

Wilson, T., Wiebe, J., and Hwa, R. (2004). Just how mad are you? finding strong and weak opinion clauses. In *AAAI*, pages 761–769.

Zhang, J., Ackerman, M. S., and Adamic, L. (2007a). Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 221–230, New York, NY, USA. ACM.

Zhang, W., Yu, C., and Meng, W. (2007b). Opinion retrieval from blogs. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 831–840, New York, NY, USA. ACM.