

Opinion and Suggestion Analysis for Expert Recommendations

Anna Stavrianou and Caroline Brun

Xerox Research Centre Europe

6, chemin de Maupertuis

38240 Meylan, France

{anna.stavrianou, caroline.brun}@xrce.xerox.com

Abstract

In this paper, we propose the use of fine-grained information such as opinions and suggestions extracted from users' reviews about products, in order to improve a recommendation system. While typical recommender systems compare a user profile with some reference characteristics to rate unseen items, they rarely make use of the content of reviews users have done on a given product. In this paper, we show how we applied an opinion extraction system to extract opinions but also suggestions from the content of the reviews, use the results to compare other products with the reviewed one, and eventually recommend a better product to the user.

1 Introduction

Social media has enabled web users to interact through social platforms, express their opinions, comment and review various products/items. Such user-generated content has been analysed from a social as well as content-oriented point of view. For instance, social network analysis techniques have been used to identify user roles (Agarwal et al., 2008; Domingos and Richardson, 2001; Fisher et al., 2006; Zhang et al., 2007) and text or opinion mining techniques have been applied to identify positive/negative tendencies within user online review comments (Ding and Liu, 2007; Ghose et al., 2007; Hu and Liu, 2004; Leskovec et al., 2010). In the applicative context, recommender systems (Adomavicius and Tuzhilin, 2005) make use of the opinion information (such as in star-rating systems) and recommend items (movies, products, news articles, etc.) or social elements (i.e. propositions to connect

with other people or communities), that are likely to be of interest to a specific user.

Typically, a recommender system compares a user profile with some reference characteristics, and seeks to predict the “preference” or “rating” that a user would give to an item not yet considered. These characteristics may be part of the information item (the content-based approach) or the user's social environment (the collaborative filtering approach). Comments published on social networking or review web sites are sometimes used by recommender systems (Aciar et al., 2007; Jakob et al., 2009) in order to find out similarities between users that comment on the same items in the same way. However, extracting explicit semantic information carried out in these comments (e.g. “this printer is slow”) is of great interest in order to detect what a user has liked or disliked about a given topic (e.g. the speed of the printer) and consequently take it into account to make recommendations.

In this paper, we propose the extraction of opinions and suggestions from user reviews or free text and their use as input information to improve recommender systems. This technique could be used on top of standard recommender techniques in order to further fine-grain the recommendation according to the user comments.

To the best of our knowledge, no existing approach takes advantage of the fine-grained opinions or suggestions the user **explicitly** expresses using natural language within a review or a free text. As aforementioned, some works consider the product reviews as a means to get user opinions on certain products and use this information for recommendation purposes. Nevertheless, they all assign a polarity (“negative” or “positive”) to

the review or they update the rating (e.g. giving a value from 1 to 5) without going further down exploiting the exact phrases. More particularly they do not detect what aspects of the product have been appreciated or not. For example, no approach considers using the user-stated phrase “I would prefer a lighter camera” in order to recommend to a user a camera that satisfies all the desired features and on top of this being lighter than the reviewed one.

The paper continues with a state-of-the art discussion. Section 3 is divided into two parts; a description of the methodology followed in order to extract opinion information from reviews through NLP techniques and a description of how this information is used for recommending product items. Section 4 shows an example and Section 5 presents a first attempt of an evaluation. Section 6 concludes and discusses future work.

2 Related Work

Although there are no works that use the explicit semantics extracted from reviews for recommendation purposes, our approach has some similarities with the analysis of reviews state-of-the-art.

Identifying the opinion of customer reviews has concerned different research communities. Some significant works infer opinion polarities based on comparisons with a pre-defined seed-list of adjectives (Ding and Liu, 2007; Hu and Liu, 2004) or implicitly through observing the changes in the respective product prices of reputation systems (Ghose et al., 2007). An attempt of extracting suggestions (and not just opinions) from customer reviews has also been presented in (Vishwanath and Aishwarya, 2011), in which ontologies and feedback rules are used for this purpose.

Combining knowledge of opinions extracted from reviews and recommender systems has also some applications. For example, (Jakob et al., 2009), have analysed opinions of movie reviews. They use pre-defined categories of movie features (acting, production, soundtrack, cinematography and storyline), and they assign polarities (negative or positive) to each category according to the per-feature opinion words expressed for each review. For example, if a movie review contains the sentence “the acting is flat”, they assign a negative polarity to the category “acting” and they just avoid recommending the specific movie to the users. They do not explicitly use the opinion in-

formation in order to make comparisons with similar movies and propose one “less flat” to the user.

Similarly to (Jakob et al., 2009), most research works that use opinion information for recommendation purposes consider only the polarity and not the explicit semantics of the opinions. For instance, in (Aciar et al., 2007) or (Poirier, 2011) they assign a kind of “rating” on each review regarding the product. Comparisons are not included.

(Sun et al., 2009) include opinion-based and feature-based comparisons in order to recommend products to users. Their approach takes into account a whole set of reviews (as opposed to individual ones) and it involves no NLP parsing. The opinions are aggregated into a sentiment value and this value points out mainly whether a product feature is better or not when it comes to comparing different models of the same product.

NLP techniques have, in some cases, been used for recommendation. As an example, in the paper of (Chai et al., 2002) the user can “chat” with the system in order to describe what type of product she desires, receiving in return a list of recommended products. Although, in this case, comparisons between products take place in the database, opinion identification is not included. The user neither expresses a complaint nor she suggests an improvement, thus, no opinion detection takes place.

3 Opinion mining for expert recommendations

In this section we describe the approach followed in order to initially parse the user reviews regarding manufactured products, extract opinion information from them and, then, use this information for the purpose of providing expert recommendations.

Each product review concerns one specific product whose brand and model are clearly mentioned each time. In web sites such as “epinions.com” this information appears in the title of the review and it is straightforward to extract. In order to make use of the content of the reviews, we apply a system relying on a deep semantic analysis that detects opinions and suggestions within the customer reviews. Natural language techniques allow the detection of the weaknesses of the product (focusing on specific features) or the potential improvements, according to

the user's point of view.

The information extracted from the reviews is then confronted to a database of products containing information such as product characteristics, usage details, average price, etc. For the purposes of this paper, we consider only product characteristics whose values can be boolean or numeric and as such they can be compared with the traditional methods. The system selects, within this database, one or more similar products that compensate for the problems or improvement needs identified within the review. Then, pointers to these products can be explicitly associated with the specific review as "expert recommendations", and constitute an automatic enrichment of the review.

The advantage for readers of these enriched reviews is to benefit from a contextualized recommendation that takes into account the semantic information conveyed in reviews of people who have used a given product. Moreover, the review's reader may be helped in her product search and may have a recommendation on a product she did not even know it exists. Figure 1 shows a schema of the process followed which is explained in more detail in the next sections.

3.1 Semantic Extraction

Our approach begins with the extraction of semantic information from each review and more specifically the identification of the user's suggestion(s) and/or opinion(s) together with the product features and respective comparison words.

For the purpose of identifying the weaknesses or the possible improvements mentioned in the text, we need to extract the opinion of a user about a given characteristic of a product. Thus, we apply an opinion detection system that is able to perform feature-based opinion mining, relating the main concept (e.g. a printer) to several features (e.g. quality, print speed and resolution), that can be evaluated separately.

Formally, our system adopts the representation of a given opinion as proposed by (Liu, 2010), where an opinion is a five place predicate of the form $(o_j, f_jk, s_{ijkl}, h_i, t_l)$, where:

- o_j is the target object of the opinion (the main concept)
- f_jk is a feature associated to the object

- s_{ijkl} is the value (positive or negative) of the opinion expressed by the opinion holder about the feature
- h_i is the opinion holder
- t_l is the time when the opinion is expressed.

The opinion extraction system is designed on top of the XIP robust syntactic parser (Aït-Mokhtar et al., 2002), which is used as a fundamental component, in order to extract deep syntactic dependencies, from which semantic relations of opinion are calculated. These semantic relations are intermediary steps to instantiate the five place predicates which are compliant with the aforementioned model. Having syntactic relations already extracted by a general dependency grammar, we use the robust parser by combining lexical information about word polarities, subcategorization information and syntactic dependencies to extract the semantic relations that will then instantiate this model.

There exist other systems, such as the one described in (Kim and Hovy, 2006), that use syntactic dependencies to link the source and target of the opinions. Our system (Brun, 2011) belongs to this family, since we believe that the syntactic processing of complex phenomena (negation, comparison and anaphora) is a necessary step in order to perform feature-based opinion mining. Another characteristic of our system is that it respects a two-level architecture; it relies on a generic level, applicable to all domains and corpora, and on a domain-dependent level, adapted for each sub-domain of application.

Moreover, our system includes a semantic mapping between polar vocabulary and the features it corresponds to. For instance, the opinion word "fast" is mapped to the feature "speed", the word "expensive" to the feature "price", the word "clunk" to "noise" and so on. This mapping enables us to further exploit the comments of the user by referring to specific product characteristics.

When analyzing an example like "*The photo quality of my prints is astonishing. This printer is really not that expensive.*", our system extracts two relations of opinion :

- OPINION_POSITIVE(astonishing,photo quality): the dependency parser extracts an

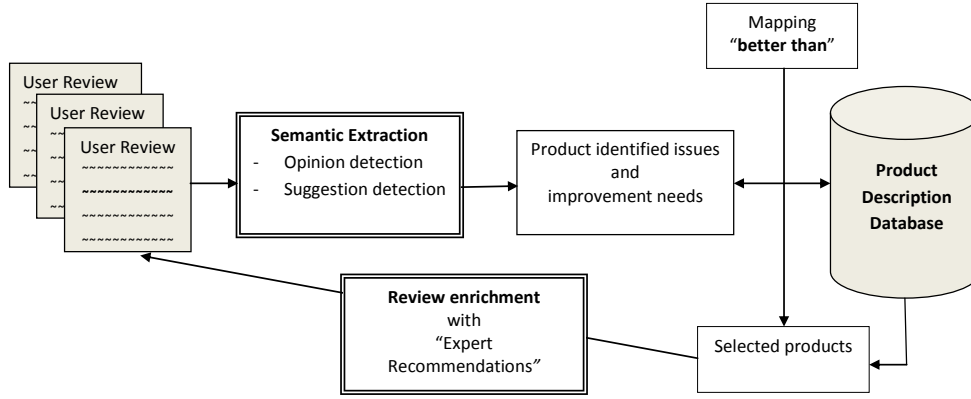


Figure 1: Extracting opinion semantic information from product reviews and provide expert recommendations.

attributive syntactic relation between the subject “*photo quality*” and the positive adjectival attribute “*astonishing*” from which this relation of opinion is inferred about the feature “*photo quality*”

- **OPINION_POSITIVE(expensive,printer):** the dependency parser also extracts an attributive syntactic relation between the subject “*printer*” and the negative adjective attribute “*expensive*”, but it also extracts a negation on the main verb: the polarity of the final relation is inverted, i.e. is finally positive. As we have also encoded that the adjective “*expensive*” is semantically linked to “*price*”, this opinion is linked to the feature “*price*”.

In addition, the system includes a specific detection of suggestions of improvements, which goes beyond the scope of traditional opinion detection. Suggestions of improvements are expressed with two discursive figures denoting “wishes” or “regrets”. To detect these specific discourse patterns, we use again information extracted by the parser, i.e. syntactic relations such as SUBJECT, OBJECT, MODIFIER, but also information about verbal tenses, modality and verbal aspect, combined with terminological information about the domain, in our case, the domain of printers.

Some examples follow that show what the system would output considering certain input sen-

tences extracted from customer reviews about printers:

1. *Input:* “I think they should have put a faster scanner on the machine, one at least as fast as the printer.”

Output:

SUGGESTION_IMPROVE(scanner, speed)

In this example, the system identifies from the input sentence that the user is not satisfied with the speed of the scanner and would have liked it to be quicker.

2. *Input:* “I like this printer, but I think it is too expensive.”

Output: OPINION_POSITIVE(printer, _), OPINION_NEGATIVE(printer, price).

In this example, the system identifies that the user is not happy with the price of the printer although the rest of its characteristics satisfy him.

3. *Input:* “The problem of this printer is the fuser.”

Output:

OPINION_NEGATIVE(printer, fuser).

In this example, the system identifies that the problem lies in the fuser of the printer.

The first two examples can be further exploited by the approach we propose. For instance, for the

second example, the reader of this review could benefit from a recommendation of a similar but cheaper printer. The third example contains information that is not measured (it has neither boolean nor numeric values) and as such it is out of the scope of this paper.

3.2 Review enrichment

Following the detection of the opinions or suggestions regarding specific product features, we identify products that match the non-mentioned or positive characteristics of the reviewed product while at the same time satisfying the user suggestions.

We consider a database that stores products together with their features. Same type of products are stored similarly for evident reasons. The database can be populated either manually or automatically through the web sites that hold product information and it needs to be updated so that new products appear and old ones are never recommended. Access to the database is done through standard SQL queries.

The system retrieves products of the same usage (e.g. a user that is reading a review for a PC laptop will not need a recommendation for a PC desktop), while selecting those ones whose features are within the same or “better” range. The features that should definitely be in “better” range are the ones retrieved with the help of the opinion detection system described previously. These features would be suggestions or negative opinions the user has expressed about a product.

The ranges can be defined in many ways and they can be subject to change. For example, the prices may be considered to change ranges every 50 Euro or 500 Euro depending on the average price of the product. The feature requested by the user (e.g. “cheaper”) should have a value in a different range in order to really satisfy her this time (e.g. a computer that costs 5 Euro less than the reviewed one is not really considered as “cheaper”).

Defining what “better” range refers to, depends on the feature. For instance, the lower the price, the better it is, whereas, the higher the speed the better. In order to avoid this confusion we keep the descending (e.g. in the case of price) or ascending (e.g. in the case of speed) semantics of the feature within the database.

Once the system has identified the products that seem to be closer to the user requirements, it high-

lights these products by presenting them as “expert recommendations”. These recommendations may appear on each review as enrichments assuming that the characteristics not mentioned as negative by the user have satisfied her, so she would be happy with a similar product having basically the mentioned features improved. The recommendation is mainly useful to the reader of the review that is in the decision process before buying a product.

Some special - sometimes often appearing - matching cases worth mentioning:

Multiple features: If more than one feature needs to be improved, priorities can be defined dependent on the order in which the features are mentioned in the review.

No comparable features: for this paper features are taken into account only if they are numeric or boolean (presence/absence) and can be subjectively compared.

Many matching products: more than one product can be recommended. The limit of the number of products can be pre-defined and the products may appear to the user in the order of less-to-more expensive.

No better answer: if no product is found that may satisfy the user then the search can go on in products of a different brand. The system has also the choice to remain “silent” and give no recommendation.

A non-demanded feature changes: in the case that a requested product is found but it is more expensive than the reviewed product, the recommendation would include some information regarding this feature (e.g. “A proposed product is “...” whose price, though, is higher”).

4 Example

Before evaluating our approach we present an example that shows the semantic extraction and recommendation process. We consider a small set of printers together with their characteristics and prices. These data are taken from epinions.com at a date just before the submission of this paper. The data appear in Table 1 in descending order of price.

Brand	Model	Usage	Technology	Black speed	Capacity	Price(\$)
X	8560 Laser	Workgroup	Color	30	1675	930
X	6360V Laser	Workgroup	Color	42	1250	754
X	6180 Laser	Workgroup	Color	26	300	750
X	4118 All-in-One Laser	All-in-One	Monochrome	18	650	747
HP	Laserjet Cp2025n	Workgroup	Color	20	300	349
HP	Laserjet M1212nf	All-in-One	Monochrome	19	150	139

Table 1: Printer information used for the purposes of the example(source: www.epinions.com).

In the examples that follow, the input is a sentence that is assumed to be in the review of a given product.

1. Review about the “6180 Laser” printer.

Input:“I think they should have allowed for a higher capacity.”

Semantic Extraction step:

SUGGESTION_IMPROVE(printer, capacity)

Identify similar products step:

- identify reviewed characteristics: workgroup, laser, color, 26 ppm black speed, 300 sheets capacity, \$750 price
- identify similar printers where capacity is higher (next range) than 300 sheets

Expert recommendation: A proposed printer with a higher capacity is the “6360V Laser Printer”.

2. Review about the “6180 Laser” printer.

Input:“I like it but it is expensive!”

Semantic Extraction step:

OPINION_NEGATIVE(printer, price)

Identify similar products step:

- identify reviewed characteristics: workgroup, laser, color, 26 ppm black speed, 300 sheets capacity, \$750 price
- identify similar printers where price is lower than \$750.

Expert recommendation: A proposed cheaper printer of the same type is “HP, LaserJet Cp2025n”.

5 Evaluation

The evaluation of the proposed system concerns two modules; the semantic extraction and the review enrichment.

The first module has already been evaluated previously showing encouraging results. The system has been evaluated as to whether it correctly classifies the reviews according to the overall opinion. The structure of the “epinions.com” web site has been used for the evaluation since each author has tagged the respective review with a tag “recommended” or “not recommended”, the corpus can be thus considered as annotated for classification. The SVM classifier (Joachims, 1998) has been used with a training set of opinions extracted by our system from 313 reviews and a test set of 2735 reviews, giving a 93% accuracy.

The review enrichment module evaluation, presented in this paper, focuses on whether the recommended products enrich the specific review and may satisfy the user by improving at least one of the negative features mentioned or following a specified suggestion without worsen the range of the rest of the features. The experiments are run against a database of 5,772 printers whose details are extracted from the “epinions.com” site.

For the purposes of this evaluation, we have developed a product comparison module that takes as input, for our case, the reviewed printer model together with the opinion and suggestion relations as extracted by the opinion mining system. The output of the comparison module is a set of recommended printers which are similar to the reviewed one while improving the negative features (based on a comparison of the feature values).

The comparison module deals with features that are numeric or boolean (presence/absence). Printers are queried against their type (color-laser/inkjet, personal/workgroup, etc.), their functions (copier, scanner etc.) and their features

(speed, resolution, etc.). Ranges have been defined according to the average per-feature-ranges that are in the database. These ranges can be extended according to the number of recommendations we would like to have (the larger the range the more the recommendations).

Certain assumptions have been made in order to provide the recommendations. One such assumption is that the author of the review knows how to best make use of the printer she has bought. For example, if the user is complaining about the printer's resolution or print quality, we assume that she makes her printing decisions (paper size, landscape/portrait) based on her knowledge of the printer's resolution. Thus, the specific review can indeed be enriched with a recommendation of a printer with a better resolution rather than an advice on how to use the specific printer (e.g. by using a different media size).

Furthermore, certain issues had to be taken care of such as missing data and different measurement units that are not necessarily comparable. When the values of the features that are to be improved are missing, the respective products are not taken into account. The missing data case is also applied when the same feature is measured in different units between two similar products. At a later stage we may include such products in the recommendations and inform the user about the differences.

The experiments were run over 129 printer reviews from the "epinions.com" site containing negative opinions and/or suggestions. The reviews concerned 6 different brands while the database from which the recommended products are extracted contains printers from 14 different brands. Once the need-to-be improved features were extracted from the reviews, the comparison module was run in order to identify the recommended products.

The recommendation output is manually evaluated by looking at the technical features on the one side and by looking at the reviews of the recommended model on the other. It has to be noted that this is a first evaluation of the system having the usual problems that recommender systems evaluations have e.g. recall calculations, finding the right experts etc. Since we have used a printer dataset, the ideal experts to validate whether we propose better or not printers would be experts from the field of printers. Not having found such

experts at the moment, we limit our evaluations to the following two-faceted one:

Feature-based evaluation: Based on the feature values, our system has a 100% precision, meaning that the recommended products are indeed similar to the reviewed ones while improving at least one of the required features. As a result, in all cases the recommended products are technically better than the reviewed one and they can help in the review enrichment.

Rating-based evaluation: In order to see whether an average user could benefit from such a recommendation, we have also evaluated our approach by looking at the reviews of the recommended products. This evaluation is quite limited, though, because not all recommended products have had reviews.

Thus, we took into account only the recommended products that have had a review. We used the average rating values of the "epinions.com" site which is a rating that considers the number of reviews together with the star-system ratings. These average ratings range from "disappointing", "ok", "very good" and "excellent". For each product we accept the recommended products that have a rating other than "disappointing" which is at least as good as the product's rating.

Only 32 products out of the 129 reviewed were used because those were the ones which had an average rating value on the web site. The accuracy we have achieved is 80.34%. In Figure 2 the percentage of accepted versus rejected recommendations is shown per brand. The brand names are replaced by numbers.

Finally, we would like to point out that in printer reviews people complain mostly about issues that do not involve comparable features (e.g. paper jams, toner problems) or that are not given as part of the detailed characteristics (e.g. cartridge prices). As such, in the future, we would like to use a different product dataset/review-set to run the experiment over.

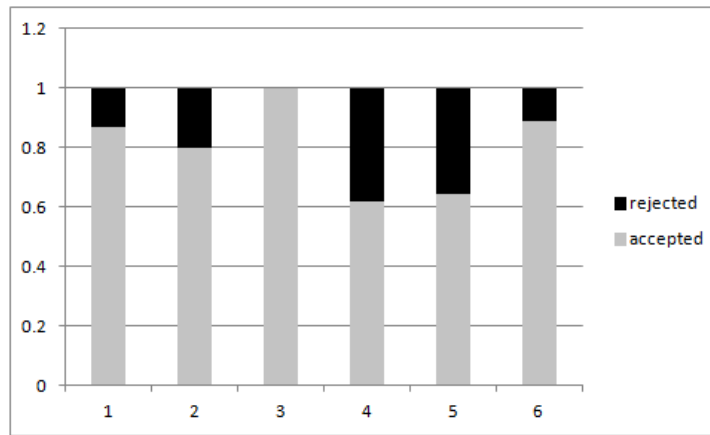


Figure 2: Rating-based evaluation results: rejected versus accepted recommendations over a number of different brands.

6 Conclusion

In this paper, we propose using written opinions and suggestions that are automatically extracted from user web reviews as input to a recommender system. This kind of opinions is analysed from a syntactic and semantic point of view and is used as a means to recommend items “better than” the reviewed one.

The novelty of our proposal lies in the fact that the semantics of opinions hidden in social media such as user reviews have not been explicitly used in order to generate recommendations. To the best of our knowledge, using the explicit comments of a user in order to enrich the reviews in a contextual manner has not yet appeared in literature.

In the future, our system could also consider the user’s role knowledge (e.g. expert or novice) in order to consider her suggestion from a different weighted-point-of-view. An expert may have already looked at certain existing products before buying something so she may need a more original or diverse recommendation provided. The role of the user could potentially be identified through the social network he is in (if there is one).

We realise that some reviews may be spam or they may be written by non-trustworthy users. However, our approach aims at providing expert recommendations as a response to a single review by considering only what is mentioned in this specific review. This means that the content of a review, even if it is spam, will not be used in order to provide recommendations for another review.

References

- Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham. 2007. Informed recommender: Basing recommendations on consumer product reviews. *IEEE Intelligent Systems*, 22(3).
- Gediminas Adomavicius and Alexander Tuzhilin. 2005. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.
- Nitin Agarwal, Huan Liu, Lei Tang, and Philip S. Yu. 2008. Identifying the influential bloggers in a community. In *WSDM ’08: Proceedings of the international conference on Web search and web data mining*, pages 207–218, New York, NY, USA. ACM.
- Salah Ait-Mokhtar, Jean-Pierre Chanod, and Claude Roux. 2002. Robustness beyond shallowness: incremental deep parsing. *Nat. Lang. Eng.*, 8:121–144, June.
- Caroline Brun. 2011. Detecting opinions using deep syntactic analysis. In *Recent Advances in Natural Language Processing (RANLP)*.
- Joyce Chai, Veronika Horvath, Nicolas Nicolov, Stys Margo, Nanda Kambhatla, Wlodek Zadrozny, and Prem Melville. 2002. Natural language assistant: A dialog system for online product recommendation. *AI Magazine*, 23(2).
- Xiaowen Ding and Bing Liu. 2007. The utility of linguistic rules in opinion mining. In *SIGIR-07*.
- Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *SIGKDD*, pages 57–66.
- Danyel Fisher, Marc Smith, and Howard T. Welser. 2006. You are who you talk to: Detecting roles in usenet newsgroups. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, pages 59b–59b.

- Anindya Ghose, Panagiotis G. Ipeirotis, and Arun Sundararajan. 2007. Opinion mining using econometrics: a case study on reputation systems. In *ACL*.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD 04*, pages 168–177. ACM.
- Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. 2009. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*.
- Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In *10th European Conference on Machine Learning (ECML)*, page 137142.
- Soo-Min Kim and Eduard Hovy. 2006. Identifying and analyzing judgment opinions. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06*, pages 200–207, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. 2010. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650.
- Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2nd ed.
- Damien Poirier. 2011. From text to recommendation (des textes communautaires a la recommandation). *PhD Dissertation*.
- Jianshu Sun, Chong Long, Xiaoyan Zhu, and Minlie Huang. 2009. Mining reviews for product comparison and recommendation. *Polibits*, 39:33–40.
- J. Vishwanath and S. Aishwarya. 2011. User suggestions extraction from customer reviews. *International Journal on Computer Science and Engineering*, 3(3).
- Jun Zhang, Mark S. Ackerman, and Lada Adamic. 2007. Expertise networks in online communities: Structure and algorithms. In *Proceedings of the 16th International conference on World Wide Web*, pages 221–230.