Towards an Intelligent Reviewer's Assistant: Recommending Topics to Help Users to Write Better Product Reviews

Ruihai Dong, Kevin McCarthy, Michael P. O'Mahony, Markus Schaal, Barry Smyth
Clarity - Centre for Sensor Web Technologies,
School of Computer Science and Informatics,
University College Dublin, Belfield, Dublin 4, Ireland
{firstname.lastname}@ucd.ie

ABSTRACT

User opinions and reviews are an important part of the modern web and all major e-commerce sites typically provide their users with the ability to provide and access customer reviews across their product catalog. Indeed this has become a vital part of the service provided by sites like Amazon and TripAdvisor, so much so that many of us will routinely check appropriate product reviews before making a purchase decision, regardless of whether we intend to purchase online or not. The importance of reviews has highlighted the need to help users to produce better reviews and in this paper we describe the development and evaluation of a *Reviewer's Assistant* for this purpose. We describe a browser plugin that is designed to work with major sites like Amazon and to provide users with suggestions as they write their reviews. These suggestions take the form of topics (e.g. product features) that a reviewer may wish to write about and the suggestions automatically adapt as the user writes their review. We describe and evaluate a number of different algorithms to identify useful topics to recommend to the user and go on to describe the results of a preliminary live-user trial.

Author Keywords

Intelligent user interfaces; text mining; writer's assistance; browser plugin.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces; H.3.1. Information Storage and Retrieval: Content Analysis and Indexing

General Terms

Design; Algorithms; Experimentation.

IUI'12, February 14–17, 2012, Lisbon, Portugal.

INTRODUCTION

Increasingly, in the world of the social web and user-generated content, users who had previously played the role of content consumers are now increasingly involved in the creation of content whether through creating our own web pages or blogs or simply commenting on the articles/posts of others. One segment of user generated content that has experienced incredible growth over the last 5 years is user reviews. Today millions of users contribute their opinions of products and services on sites like TripAdvisor, Hotels.com, Amazon, BestBuy, IMDB, and countless more. In fact, user reviews are routinely providing an important and unique service to buyers every day. Indeed for many of us it is all but unthinkable that we would book a hotel without checking out its TripAdvisor reviews. And millions of shoppers every day use Amazon as a source of product review data even if they are planning to purchase elsewhere.

The growing importance of this type of user-generated content has highlighted a number of interesting research challenges. First and foremost, considerable research attention has recently been paid to better understanding the quality and fairness of user-generated reviews. For example Mahony and Smyth [9] uses reviewer's reputation and Liu et al.[7] uses reviewer's genre familiarity to predict the quality or helpfulness of a review. In addition, review length and unigram distribution, cf. Kim et al. [6], or recency of reviews, cf. Liu et al. [7], have also shown good performance as predictive classifiers for review quality. The power of combining multiple criteria has been investigated by Wu et al. [12]. Wu et al. [13] also investigated distortion as a validation measure for the classification of suspicious reviews.

In this paper we are also interested in ensuring review quality but we adopt a very different, albeit complementary, approach. Rather than evaluating the quality of reviews that have already been written, we support the process of writing good reviews. Inspired by the GhostWriter system [4, 5], we have developed the *Reviewer's Assistant* system to work with sites like TripAdvisor and Amazon and to make suggestions to users as they start to write a review. These suggestions are optional and of course the user may choose to ignore them. The sugges-

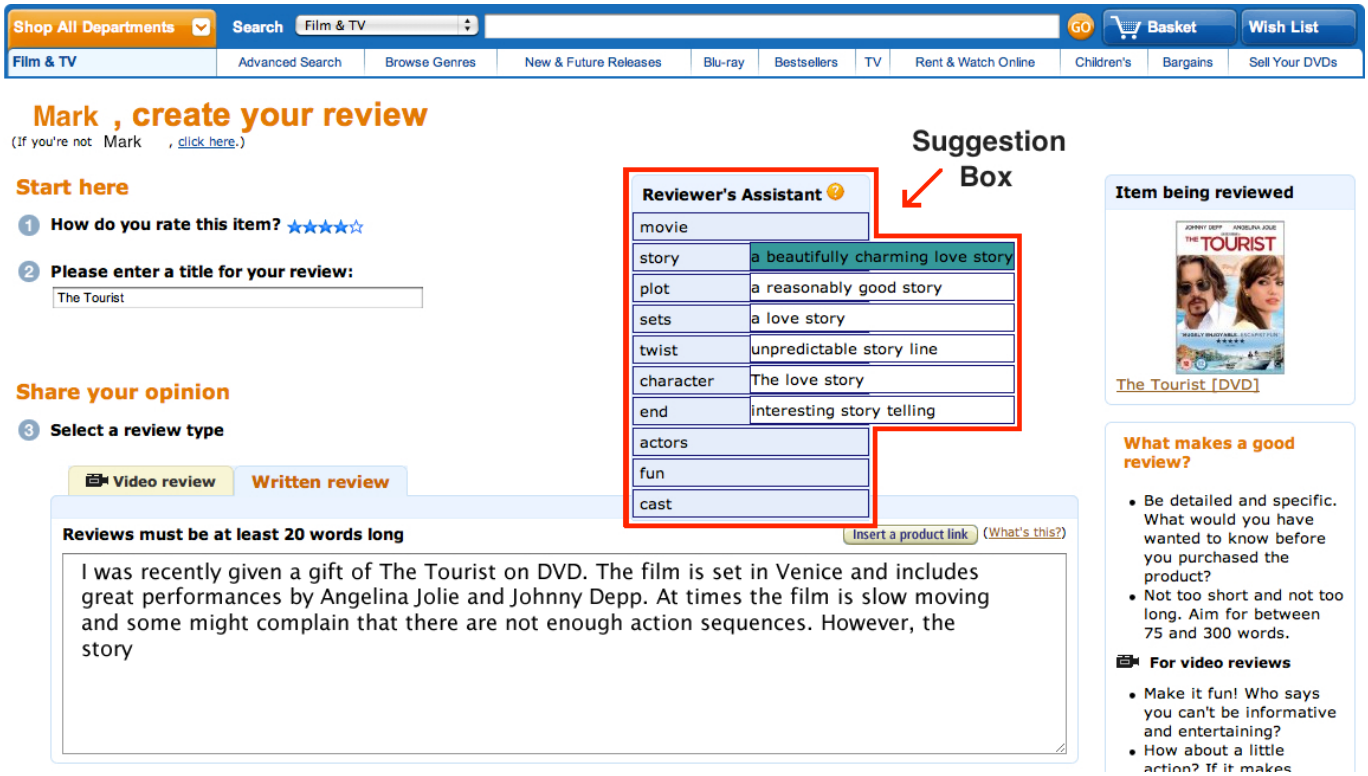


Figure 1: The *Reviewer's Assistant* browser plugin

tions take the form of a set of words that are designed to highlight key features/topics related to the product or service that the user is writing about, which have been extracted from other related reviews. As they write their review these suggestions change in response to what they write. For example, in reviewing a Nikon D90 SLR camera on Amazon, initial suggestions to the user might encourage them to discuss the *price* of the camera or its *resolution*; and via the assistant tool, the user can always see what other users have written about these features if they wish. Moreover, as the review becomes more complete, these suggestions will change. Features that have already been covered will fall away and new, possibly more niche features, will be suggested. For example, the reviewer might be encouraged to discuss the camera's *battery* life or the *weight* of the body or even its *shutter* type. Of course these suggestions are entirely optional. At all times it is up to the user whether they choose to write about particular suggestions but because they are available our hope is that some users will benefit some of the time and that this will lead, in due course, to a greater number of high quality, helpful reviews. An example screen shot of the Reviewer's Assistant is presented in Figure 1. Briefly it shows the Reviewer's Assistant's suggestions box with a number of suggested topics related to the review that the user has started to write; in this case it is a review about the movie The Tourist starring Angelina Jolie and Johnny Depp. As indicated, if the user mouses over one of these suggestions then they see a short list of review fragments from related reviews that have touched on this topic. We will return to this example later and discuss the various techniques that have been used to extract these recommendations from review content in real-time.

The main contribution of this paper is as follows. First we review the GhostWriter work and propose that its focus on noun-phrase suggestions is not ideal when it comes to providing review-writing assistance because it can lead to a cut-and-paste type reviewer behavior. Rather, we propose a focus on extracting *topics* (nouns) from related reviews and suggesting these topics to users to encourage them to form their own opinion on these topics. We describe a number of different recommendation strategies based on this idea and evaluate them on a range of real-world data sets. In addition, we describe the development of the *Reviewer's Assistant* browser plugin and describe the results of a preliminary live-user trial on Amazon.

THE GHOSTWRITER SYSTEM

The GhostWriter system was first proposed by Bridge et al. [4] and Healy and Bridge [5] as an approach to guide users in the construction of short snippets of user generated content. Originally, this took the form of guiding users during the creating of classified adverts for products they wished to sell/exchange. Subsequently, the GhostWriter researchers turned their attention to helping users produce review content on Amazon.

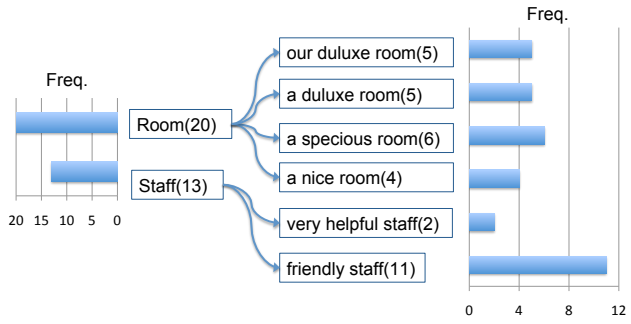


Figure 2: *Multiple noun phrases representing the same topic*

The system is inspired by Conversational Case-Based-Reasoning (CCBR) techniques [3] where existing reviews are treated as past cases and the current review fragment as a query. Currently GhostWriter 2.0 [5] extracts noun phrases from these past product reviews (cases), and suggests these phrases directly to the reviewer/user. At the beginning of our work on providing a similar type of assistance to budding reviewers we had in mind two important questions about the GhostWriter approach. Are noun phrases really the right type of objects to guide a writer of reviews? How can we measure (and improve) the quality of suggestions presented to the user?

The noun phrases suggested by the GhostWriter system to the editor of a review typically carry two types of information: highly structured knowledge from prior reviews which are presented as sentence fragments; and topical knowledge contained in those fragments. And the primary aim of GhostWriter was to provide a set of useful fragments that could be copied directly into an evolving review. However, there are two problems with this approach. Very often the review fragment (phrase) that is suggested does not make for a clear talking point for the reviewer; the central topic can be somewhat hidden in the larger review fragment. Moreover, the intent to remove review friction by facilitating a copy-and-paste approach can inevitably lead to reviews that lack novelty and interest.

Concretely then there are two key issues with GhostWriter that are considered as opportunities for improvement in this paper:

1. Nouns might be represented by multiple noun phrases, as illustrated in Figure 2. Assuming there are two features hidden in the reviews: *room* and *staff*. *room* is mentioned 20 times and *staff* is mentioned 13 times. If we only recommend one feature based on the frequency from the noun words point of view, *room* is the best choice. However, if we extract the feature from the noun phrase directly, *friendly staff* is a best choice based on the frequency calculation.

2. The relationship between current writing and topics to be covered in the remainder of the writing process is not sufficiently considered by GhostWriter 2.0. In particular, association rule mining, which is generally useful for discovering interesting relationships hidden in large data sets, may help to reduce this inaccuracy. For example, in market basket analysis, $\{Bread, Milk\}$ is an example of a frequent itemset. It indicates that a relationship exists between the sale of *bread* and *milk* because many customers who buy *bread* also buy *milk*.

THE REVIEWER'S ASSISTANT

The *Reviewer's Assistant* has been developed as a browser plugin so that it can integrate directly with review systems across a wide variety of web sites. In this paper we focus on its application to Amazon but it will work in a similar manner with sites such as BestBuy, Hotels.com, TripAdvisor etc. As mentioned previously the Reviewer's Assistant takes the form of an additional recommendation module that appears on the review-creation pages of Amazon as shown in Figure 1. Quite simply the module presents an updating list of topic recommendations to the user as they write their review. These suggestions are extracted from the texts of related reviews in real-time and based on the review content that the reviewer has provided so far. As such the suggestions adapt to the review as the reviewer writes it: as topics are covered, these suggestions fall away and are replaced with additional suggestions. The module is fully interactive and the user can, for example, hover over a suggestion to see additional information, such as the review fragments from related reviews that discuss the topic. One simple difference between the Reviewer's Assistance and GhostWriter, that is worth highlighting at this stage, is that the former recommends nouns (topics) to the user in the first instance, with noun phrases (review fragments) available on request, whereas the latter recommends noun phrases directly.

This move from noun phrases to nouns, while simple, is important. It is motivated by the observation that many words in noun phrases do not carry the actual meaning and can distract from the topic in question. Also, it is less attractive to use nouns for a cut and paste writing style, and we hope to increase the quality of the review authors by shifting the focus from *coverage* (What do I need to write about?) to *evaluation* (How good is feature xyz?).

In what follows we will describe the techniques that the Reviewer's Assistant users to extract, recommend, and rank review topics.

System Overview

The overall Reviewer's Assistant system architecture is presented in Figure 3 and can be best understood with reference to the following core components:

1. *Filtering.* Select *good quality* User Generated Content (UGC) as knowledge base. UGC is created by common users without supervision. These common users are

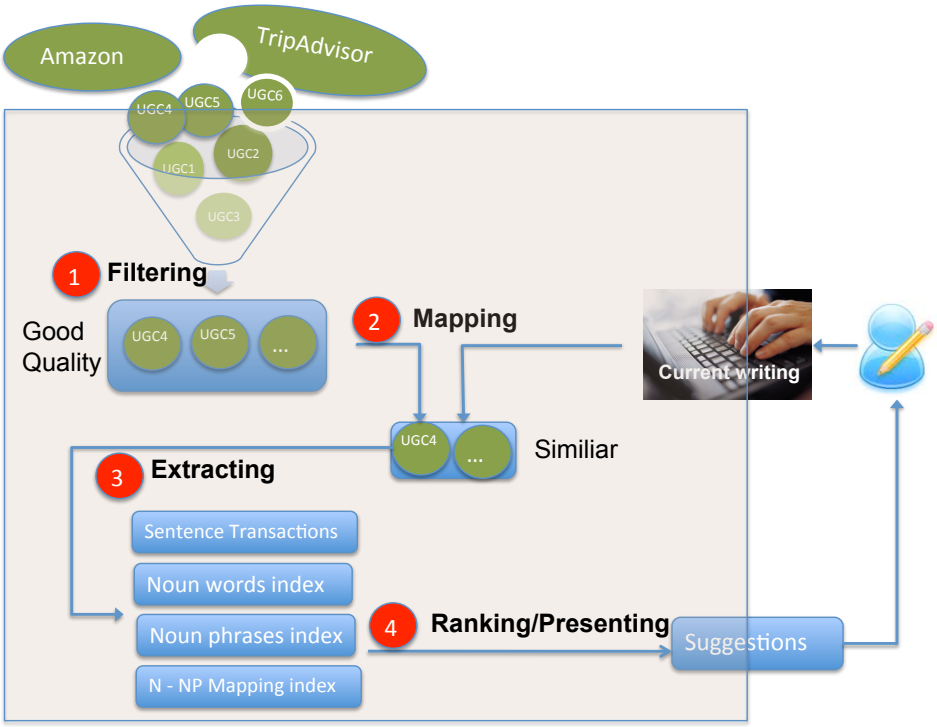


Figure 3: *High level system view of the Reviewer's Assistant*

not professional or technical writers. Therefore the quality of UGC is an issue.

2. *Mapping.* Assuming good quality UGC is identified, the similarity between the current writing of the user (in progress) and similar texts in the knowledge base is computed to identify *relevant* UGC. For example, when a user writes a camera review, books reviews might be not useful for her writing even though these reviews have a good quality.

3. *Extracting.* Extracting features from relevant and good quality UGC is the basis for generating and ranking suggestions. Features can be manifold, including indices to nouns or noun phrases, transactions for association rule mining, or mappings between topics (nouns) and noun phrases.

4. *Presenting.* In order to present topics to the user, suggestions must be generated from the extracted features and ranked in order to pick top N suggestions.

While being an important system component to ensure quality, *Filtering* is not investigated here. For our experimental knowledge bases we used review helpfulness.

For *Mapping*, all system variants investigated here use the Jaccard similarity coefficient to measure the similarity between the current user's writing and similar texts in the knowledge base, and the top 50 reviews have been selected consistently for further processing.

Let c_i, c_t be two texts, then the Jaccard similarity coefficient is calculated by equation 1; where $|c_i \cap c_t|$ is the number of words (ignoring stopwords) that both pieces of text have in common.

$$sim(c_i, c_t) = \frac{|c_i \cap c_t|}{|c_i \cup c_t|} \quad (1)$$

Note, the topics that can be presented to the user are limited by the broadness of the knowledge base. We have considered the exploitation of related knowledge bases for cases of insufficient coverage, but we consider this to be beyond the scope of this work.

Extracting for Reviewer's Assistant

We use OpenNLP[1] to split reviews into sentences, and to extract noun words and noun phrases from each sentence. For noun phrases, we record the frequency and the length of noun phrases. For noun words, only frequency is recorded. For association rule mining, we also generate transactions from sentences. A transaction is simply the set of noun words without stop words that is contained in a sentence. Figure 4 shows an example of converting a review to a group of sentence transactions.

Strategies for Generating Suggestions

We will compare the noun phrase selection strategy of Ghostwriter 2.0 with various noun selection strategies of

[1]OpenNLP: `http://incubator.apache.org/opennlp/`

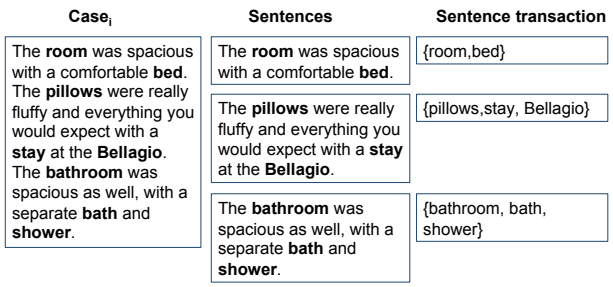


Figure 4: *Example of sentence transactions*

Reviewer's Assistant. All strategies propose exactly 10 (topical) suggestions to the user.

The following algorithm simulates the Ghostwriter generation of noun phrase suggestions ($gw-$ denotes GhostWriter, np denotes noun phrase):

1. *gw-np* is an approximation of the GhostWriter 2.0 system. It generates noun phrases as suggestions and ranks these suggestions by the product of frequency and length of the noun phrase, e.g. if s is the suggested noun phrase and $freq(s)$ the total count of occurrences of the noun phrase in the similar reviews, then
$$score(s) = freq(s) \times length(s).$$
Some suggestions share the same score. To break ties, the GhostWriter 2.0 system (and *gw-np*) sums and compares the helpfulness of the reviews that contain the two suggestions.

Algorithm 1: The $ra-ans-n$ Algorithm

Input: T, $CurrWriting$
Output: S
$S = \emptyset$;
F_3 =k-itemsets($T, k = 3, min\_support = 2$);
$Rules$ =rules($F_3, min\_confidence = 0.5$);
$count = 0$;
while $Rules \neq \emptyset$ **do**
  select $r \in Rules$ with highest confidence;
  $Rules = Rules \backslash \{r\}$;
  if *left side of* r *in CurrWriting* **then**
    $S = S \cup \{$right side of $r\}$;
    $count++$;
  end
end
if $count < 10$ **then**
  fill-up S from $tf-n$;
end

Four different new algorithms for extracting the Top 10 topical nouns were evaluated ($ra-$ denotes Reviewer's Assistant, ans denotes a-priori for nouns with sentence transactions):

1. *ra-n* is the baseline for noun word suggestions. It works like the original GhostWriter, but ranks nouns instead of noun phrases, e.g. if s is the suggested noun, then $freq(s)$ is the total count of occurrences of the noun in the similar reviews. Note, nouns are considered as noun phrases of length one, so the score is identical to the frequency.

2. *ra-ans-n* uses Association Rule Mining [10, 1] to search for missing topics once topical nouns have been identified. For the Reviewer's Assistant, we deployed the efficient a-priori association rule mining algorithm by Agrawal and Srikant [2]. Algorithm 1 shows the entire *ra-ans-n* Algorithm. It takes the sentence transactions (T) and the current writing as input and produces a set of 10 suggestions (S) as output. First we generate frequent 2- and 3-itemsets from *sentence transactions* with fixed minimum support of 2[2]. We then generate the rules for a minimum confidence of 0.5, and add the right sides of applicable rules to our suggestion set S until we have 10 suggestions. If the algorithm does not generate sufficient noun words, it uses *ra-n* as a fall-back strategy, i.e. the remaining noun word suggestions are generated from *ra-n*.

3. *ra-df* uses document frequency of noun words instead of term frequency, e.g. if s is the suggested noun, then $freq(s)$ is the total count of reviews (documents) in which the suggestion occurs.

4. *ra-ans-df* is identical to *ra-ans-n*, except it uses *ra-df* as fall-back strategy if less than 10 noun words are generated.

EXPERIMENTAL EVALUATION: OFF-LINE STUDY

So far we have motivated the Reviewer's Assistant as a tool to help users during product review tasks. Our central hypothesis is that by extracting key topics from past related reviews we can provide new reviewers with a series of hints or suggestions about the type of topic that they may wish to cover in their own review. In this section we will describe a two-part evaluation of the Reviewer's Assistant. First we will focus on an off-line, leave-one-out style evaluation of the different topic extraction techniques that we have described in order to assess their potential effectiveness when it comes to identifying topics that are likely to be written about by reviewers. Of course for a system such as the Reviewer's Assistant and off-line study such as this only tells part of the story and the real test is whether users find the service useful and whether it helps them to write better reviews. Therefore, we also describe a live-user trial in which a group of users spent time using the Reviewer's Assistant in a realistic setting as they wrote product reviews. We will describe their observations and summarize their assessment of the utility of the system, leaving the question of whether it helped them create better reviews as part of future work.

[2]The value for the maximum itemset size has been determined, then we chose a pareto-optimal combination of minimum support and minimum confidence by means of prior experiments across the entire set of possible parameter values.

Off-Line Datasets

The data for our experiments originally consisted of two different product review data sets from Amazon and TripAdvisor. Each data set was produced by mining review content directly from Amazon and TripAdvisor. In addition, for the purpose of this evaluation we also created a second Amazon data set that focused on a small set of six frequently reviewed products; the purpose of this was to look at the evaluation in the context of a mixture of sparsely populated and more densely populated review data sets.

A summary of the data contained in these three data sets is presented in Tables 1, 2, and 3. In the case of the Amazon data sets the reviews cover a range of products and product types (books, consumer electronics, DVDs etc.) as shown. In the case of the first Amazon data set (Amazon I) there are 100 randomly chosen reviews for each product category and reviews were selected only if they received at least 5 positive votes for helpfulness according to the Amazon review data. As per Table 1 each review contains from 6-9 sentences and includes about 30-50 noun words as candidate topics. The second Amazon data set (Amazon II) focused on a particular set of 6 popular products and for each product we have included from 57-100 reviews. Similar statistics are presented for the number of sentences and nouns in these more focused review data, cf. Table 2.

The TripAdvisor data set is based on a randomly selected set of 100 reviews for each class of hotels as shown in Table 3. As shown the TripAdvisor reviews tend to be a lot longer than their Amazon equivalents, ranging in length from 13-20 sentences. In terms of content focus, the TripAdvisor data set is more focused than the Amazon I data set, but broader than the Amazon II data set.

Category	#Rev	#Sen	#Words	#NW
books	100	7.85	182.77	49.23
electronices	100	9.23	174.83	42.1
camera	100	7.61	144.81	34.71
dvd	100	8.61	199.85	52.83
music	100	6.58	146.95	39.12

Table 1: Amazon I data set (categories, broad)

Item	#Rev	#Sen	#Words	#NW
The Kindle	99	16.23	407.40	85.65
A Vacuum Clean.	100	7.16	158.23	34.11
A Book on Chin.	57	4.56	96.18	21.09
The Nikon D90	77	6.88	156.84	36.84
A Samsung TV	66	4.55	99.76	22.97
The Tourist DVD	65	6.21	142.34	35.8

Table 2: Amazon II data set (products, focussed)

Experimental Setting

For evaluation purposes, we have adapted a standard leave-one-out methodology. For each of the review data sets we temporarily remove each review in turn; this is the *target review* and plays the role of the review that is currently being constructed in this evaluation. In turn we remove content from this review to reflect the review at various stages of completion. For example, we use different ablation levels from 20%-80% such that, for example, at the 80% ablation level it means that only 20% of the original review is available for use. The remaining review, denoted by Q (for query) is then used as the basis for the Reviewer's Assistant recommender engine to generate suggestions. The removed review is not used for the generation of suggestions, i.e. neither for the computation of frequency counts nor for the generation of frequent item sets. The ablated portion of the review is denoted as T and is used to test these suggestions.

Hotel Class	#Rev	#Sen	#Words	#NW
2.5 of 5	100	14.29	258.65	63.04
3.0 of 5	100	13.84	275.65	67.86
3.5 of 5	100	19.87	364.95	90.2
4.0 of 5	100	20.26	408.33	101.32
4.5 of 5	100	18.38	361.99	91.63
5.0 of 5	100	20.77	392.85	97.68

Table 3: Tripadvisor data set

We use each of the different generation/ ranking strategies described previously in turn to generate 10 suggestions, denoted by S per strategy for Q. To evaluate these suggestions we turn to two common metrics, *Precision* and *Recall* by comparing the suggestions from Q to the remainder of the review T. The intuition here is that if many of the suggestions are contained within the remainder of the review then this is good because the suggestions were actually written about by the original reviewer. This precision is the percentage of suggestions that are contained in T. Conversely, if many of the nouns in T are contained in the suggestions S then this is also good because it implies that the suggestions provide good coverage of the topics that were important to the original reviewer. Thus recall is the percentage of nouns present in T that are contained within the suggestions set S.

$$precision = \frac{|T \cap S|}{|S|} \tag{2}$$

$$recall = \frac{|T \cap S|}{|T|} \tag{3}$$

In the experiments, there are two kinds of suggestions, *single noun words* and *noun phrases*. For single noun words, $n \in T \cap S$ is easy to decide. A suggestion is in the ablated part of the review, if and only if it appears there. For noun phrase suggestions, the situation is slightly more complex. We therefore define a range of precision and recall for noun phrases in the following way:

- All (Min. Precision/ Recall): We consider a suggestion to be in the ablated part of the review iff *all* of its nouns are present.

- Any (Max. Precision/ Recall): We consider a suggestion to be in the ablated part of the review iff *at least one* of its nouns is present.

Ablation Experiment

Even though a quantitative comparison between GhostWriter and Reviewer's Assistant is impossible, due to the natural differences between nouns and noun phrases, we have tried to illustrate the differences in a qualitative manner.

In Figure 5, we show the comparison of precision and recall for each of the 3 data sets (6 sub-figures). Each of the sub-figures provides a comparison between GhostWriter (precision range from $gw - np(min)$ to $gw - np(max)$) and different strategies of Reviewer's Assistant (denoted by $ra - X$). For some of the data sets, there was no visual difference between some of the strategies. In these cases we represented multiple methods by a single line of mean values.

The following qualitative observations motivated us to do the Pilot Study with the Reviewer's Assistant plugin:

- Reviewer's Assistant (nouns) is consistently better than the minimum of the GhostWriter (noun phrases) precision range, i.e. at least nouns are a reasonable candidate for topical suggestions instead of noun phrases.

- Reviewer's Assistant consistently improves for higher ablations relative to the maximum of the GhostWriter precision range.

- For the Amazon I data set and for the Trip Advisor data set, Reviewer's Assistant is best. For the Amazon II data set, Reviewer's Assistant is not stronger than Ghostwriter, but improves for higher ablations and is equally good as the maximum of the GhostWriter precision range for 80% ablation.

EXPERIMENTAL EVALUATION: LIVE-USER STUDY

In the previous section we described the results of an off-line ablation study to illustrate the relative effectiveness of our different recommendation strategies across 3 different data sets. The true test of a system like Reviewer's Assistant however is clearly how it performs in a real-world setting with live users. To that end we describe a preliminary live-user study in this section.

Study Setup

The purpose of this study is to evaluate the performance of the Reviewer's Assistant in a realistic setting in which users are asked to write reviews for particular products. In this case we focused on Amazon and configured the Reviewer's Assistant to generate its suggestions using the $ra - ans - n$ technique described earlier.

In total there were 19 participants, recruited from the School of Computer Science and Informatics in University College Dublin. Each user was instructed to install the Reviewer's Assistant Browser plugin in their browser (Google Chrome in this case) and they were asked to write reviews for a particular set of products that were familiar to them. Only 4 out of the 19 users had previously written an Amazon review prior to the study and in total the 19 users produced 40 different reviews.

During the study we logged the suggestions provided by the Reviewer's Assistant and the reviews produced as they were written. The participants also completed a post-study questionnaire. We were interested in understanding a number of key aspects of the system. First and foremost, was there evidence that the suggestions made by the system were useful to the reviewers; was there any evidence that they preferred the noun-based approach to the noun-phrase refinements? And ultimately, having used the system did users find it to be useful and would they use it in the future if available?

Suggestion Relevance

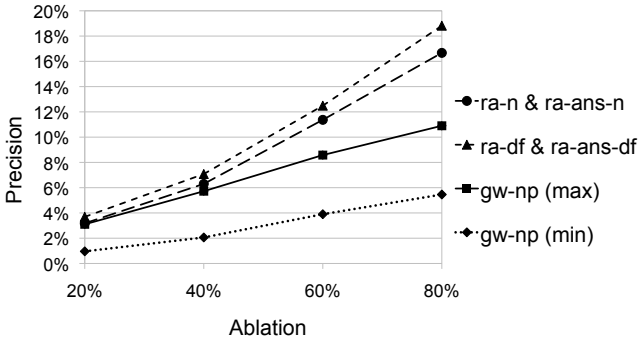
To evaluate the suggestions made to the user (S) we use an indirect measure of relevance by comparing these suggestions to the noun content of the review the user actually wrote (N). This provides precision- and recall-like measures as follows; in a technical sense we are measuring *precision* and *recall* of our suggestions against the complete review written by the user.

- $\frac{|S \cap N|}{|S|}$ or Precision: The percentage of suggestions the user actually wrote into the review.

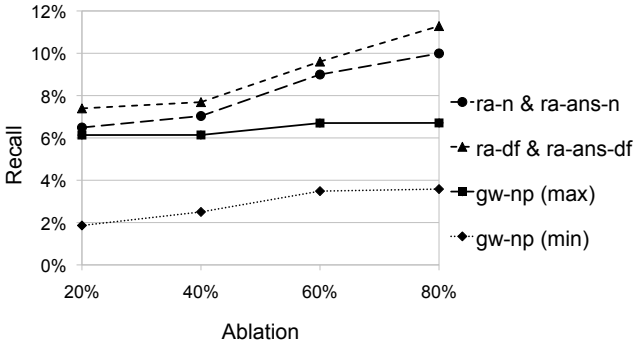
- $\frac{|S \cap N|}{|N|}$ or Recall: The percentage of suggestions among the nouns written by the user.

Of course strictly speaking it is not possible to be fully confident that just because a suggestion was made by the system and written about by the user does mean that the user was influenced by the suggestion. Perhaps they did not notice the suggestion but wrote about the feature anyway. This is certainly a limitation but since our participants were generally paying attention to the suggestions made by the Reviewer's Assistant during this study we can be somewhat confident in our measures, and even if a particular reviewer did not notice a suggestion, the fact that they subsequently wrote about it still speaks to the effectiveness of the recommendation strategy.

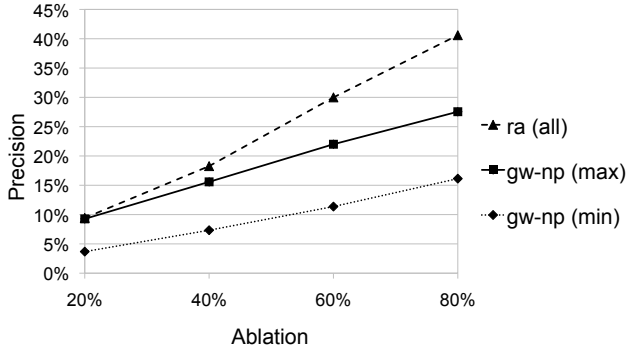
The results are presented in Figure 6 as a graph of precision and recall (as defined above) versus review length (number of words), averaged over the 40 reviews produced by the 19 users; note that the x-axes also indicates the number of actual reviews. Overall we can see that the Reviewer's Assistant performs well across different length of reviews. As expected precision improves with review length, since longer reviews have a greater opportunity to include suggested topics. For example, we can see that shorter reviews (up to 40 words) contain



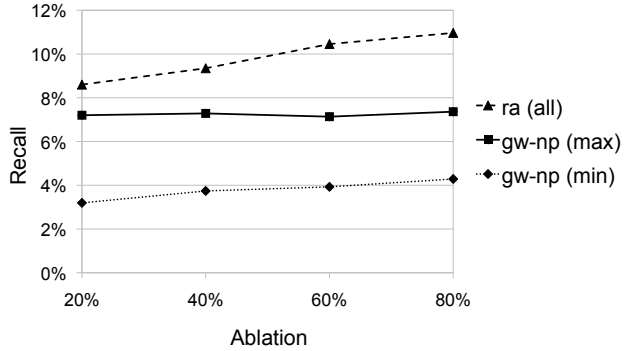
(a) Amazon I (5 categories) - precision



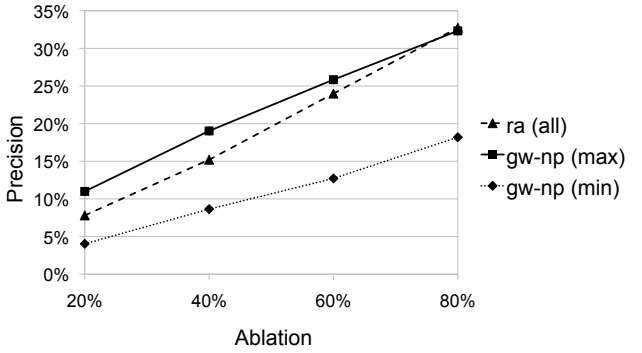
(b) Amazon I (5 categories) - recall



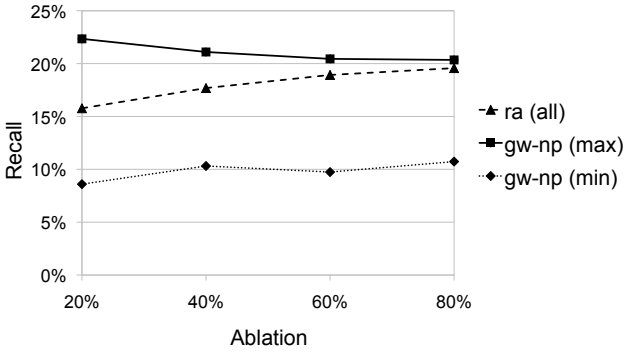
(c) Trip Advisor - precision



(d) Trip Advisor - recall



(e) Amazon II (6 products) - precision



(f) Amazon II (6 products) - recall

Figure 5: Comparison

about 25% of the topics suggested during their construction and this rises to more than 40% for longer reviews with more than 120 words. Likewise recall is seen to fall as review length increases indicating that proportionally fewer suggestions are being used in longer reviews: for short reviews recall approaches 60% and falls to just over 30% for longer reviews.

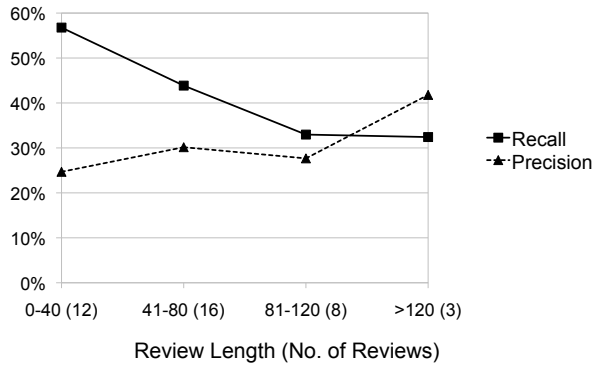


Figure 6: *Suggestion relevance*

Nouns vs. Noun Phrases

As described previously, the Reviewer's Assistant allows users to move from noun-based suggestions to noun-phrases that represent review fragments. In the study, noun-based suggestions were presented by default, but participants were reminded that they could see further details and review fragments and we were interested to see how often users would make this switch and when they did, whether they would continue to use noun-phrases. In fact, in only 10 out of the 40 reviews created did users ever switch to noun phrases. The total time they spent on average an noun phrases and nouns during each review is illustrated in Figure 7 as opposed to the average time the other 30 users spent on their reviews in total (on nouns).

Interestingly, even those 10 users who explored *noun phrases* as suggestions, switched back to *nouns* after an average time of 25 seconds, while they stayed for an average of 44 seconds with *nouns* even after they discovered *noun phrases* as an alternative.

User Feedback

As mentioned previously, upon completion of the study each user was asked to complete a short questionnaire to provide feedback on the Reviewer's Assistant. In particular we asked each participant to indicate their agreement/disagreement on four key statements:

1. The suggestions made by the Reviewer's Assistant were relevant to the product you were reviewing.
2. The suggestions made were helpful with respect to the review you were writing.
3. Taken together the suggestions made provided good coverage of the product you were reviewing.

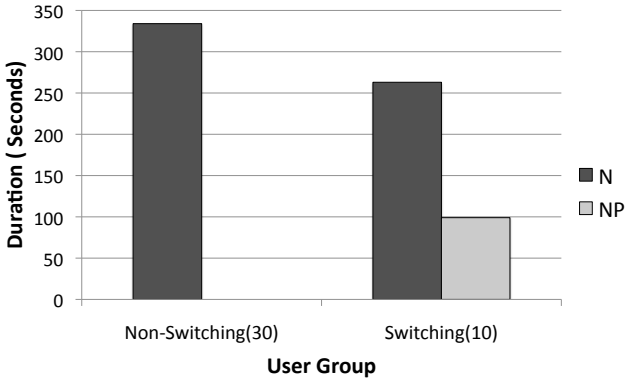


Figure 7: *Time spent on nouns vs. noun phrases*

4. You were satisfied with the utility of the Reviewer's Assistant and would use it again if available.

The results are summarized in Figure 8 and show the strong levels of support expressed by the participants. For example, we can see that more than 90% of users agreed that the suggestions being made were relevant to the product they were reviewing. About 75% of users found these suggestions to be helpful and comprehensive. And overall, more than 70% of users expressed a level of satisfaction with the Reviewer's Assistant that would lead them to use the tool in the future if available.

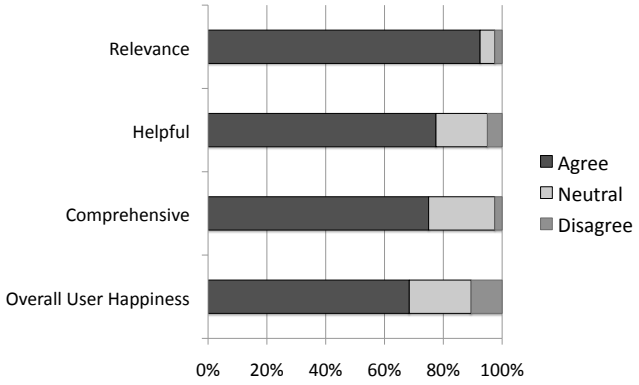


Figure 8: *User feedback*

CONCLUSIONS AND FUTURE WORK

As more and more users become producers of online content there is a need for a new generation of supporting tools, especially to help to guide users during their initial forays into user-generated content. This seems particularly relevant when it comes to producing reviews, which more users are inclined to do, and which are proving to be an increasingly important repository of decision informing opinions.

In this paper we have described the Reviewer's Assistant tool that is designed to support users as they write product reviews. We have described an initial set of

techniques for automatically extracting useful suggestions from past reviews which can be recommended to new reviewers as they write. We have evaluated these strategies and demonstrated their practical usefulness in a live-user trial.

Undoubtedly there is still much to do in relation to this area of research. And certainly this paper represents an initial foray, presenting a set of benchmark techniques, and proving some preliminary evaluation results. These results indicate the the noun-based approach used by the Reviewer's Assistant performs better than previous approaches that have proposed a noun-phrase based approach. Overall, user feedback indicates that the Reviewer's Assistant is capable of making relevant, helpful, and comprehensive suggestions to reviewers as they write and the majority of users indicated that that they would use the system in the future. An open question is the quality of the resulting reviews, i.e. is it true that user satisfaction with the Reviewer's Assistant is reflected in higher review quality?

There is a obvious opportunity to improve the sophistication of the recommendation techniques used in the Reviewer's Assistant and as a next step, we plan to explore related methods for the improvement of topical suggestions with nouns. In particular, we will look into topic detection and topical relationships among nouns. Discovering conceptual relationships from text corpora, cf. e.g. Maedche and Staab [8], may help to extract topical features after mapping is completed. Topic discovery, e.g. the identification of descriptive terms for a set of texts, cf. e.g. Schaal et al. [11], may help to select the right nouns, especially if combined with word sense disambiguation.

ACKNOWLEDGMENTS

This work is supported by Science Foundation Ireland under grant 07/CE/I1147.

REFERENCES

1. R Agrawal, Tomasz Imieliński, and A Swami. Mining Association Rules between Sets of Items in Large Databases. *ACM SIGMOD Record*, 22(May):207–216, 1993.
2. Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
3. David W Aha, Leonard A Breslow, and Héctor Muñoz-Avila. Conversational Case-Based Reasoning. *Applied Intelligence*, 14(1):9–32, 2001.
4. Derek Bridge and Aidan Waugh. Using experience on the read/write web: The ghostwriter system. In Derek Bridge, Enric Plaza, and Nirmalie Wiratunga, editors, *Procs. of WebCBR: The Workshop on Reasoning from Experiences on the Web (Workshop Programme of the Eighth International Conference on Case-Based Reasoning)*, pages 15–24, 2009.
5. Paul Healy and Derek Bridge. The GhostWriter-2.0 System: Creating a Virtuous Circle in Web 2.0 Product Reviewing. In Derek Bridge, Sarah Jane Delany, Enric Plaza, Barry Smyth, and Nirmalie Wiratunga, editors, *Procs. of WebCBR: The Workshop on Reasoning from Experiences on the Web (Workshop Programme of the Eighteenth International Conference on Case-Based Reasoning)*, pages 121–130, 2010.
6. S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 423–430, Sydney, Australia, July 22-23 2006.
7. Y. Liu, X. Huang, A. An, and X. Yu. Modeling and predicting the helpfulness of online reviews. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM 2008)*, pages 443–452, Pisa, Italy, December 15-19 2008. IEEE Computer Society.
8. Alexander Maedche and Steffen Staab. Discovering Conceptual Relations from Text. In *ECAI'00*, pages 321–325, 2000.
9. M. P. O'Mahony and B. Smyth. Learning to recommend helpful hotel reviews. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys 2009)*, New York, NY, USA, October 22-25 2009.
10. Gregory Piatetsky-Shapiro. *Discovery, Analysis, and Presentation of Strong Rules*, volume 229, pages 229–248. AAAI/MIT Press, 1991.
11. Markus Schaal, Roland M. Müller, Marko Brunzel, and Myra Spiliopoulou. RELFIN - Topic Discovery for Ontology Enhancement and Annotation. In *ESWC'05*, pages 608–622, 2005.
12. G. Wu, D. Greene, and P. Cunningham. Merging Multiple Criteria to Identify Suspicious Reviews. In *Proc. 4th ACM Conference on Recommender Systems (RecSys'10)*, 2010.
13. Guangyu Wu, Derek Greene, Barry Smyth, and Pádraig Cunningham. Distortion as a Validation Criterion in the Identification of Suspicious Reviews. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 10–13, New York, NY, USA, 2010. ACM.