Automatic Classification of Web Search Results: Product Review vs. Non-review Documents

Tun Thura Thet, Jin-Cheon Na, and Christopher S.G. Khoo

Wee Kim Wee School of Communication and Information
Nanyang Technological University
31 Nanyang Link, Singapore 637718
{ut0001et,tjcna,assgkhoo}@ntu.edu.sg

Abstract. This study seeks to develop an automatic method to identify product review documents on the Web using the snippets (summary information that includes the URL, title, and summary text) returned by the Web search engine. The aim is to allow the user to extend topical search with genre-based filtering or categorization. Firstly we applied a common machine learning technique, SVM (Support Vector Machine), to investigate which features of the snippets are useful for classification. The best results were obtained using just the title and URL (domain and folder names) of the snippets as phrase terms (n-grams). Then we developed a heuristic approach that utilizes domain knowledge constructed semi-automatically, and found that it performs comparatively well, with only a small drop in accuracy rates. A hybrid approach which combines both the machine learning and heuristic approaches performs slightly better than the machine learning approach alone.

Keywords: Product Review Documents, Genre Classification, Snippets, Web Search Results.

1 Introduction

In recent years, we have witnessed tremendous growth of online discussion groups and review sites, where an important characteristic of the posted articles is their sentiment or overall opinion towards the subject matter. Researchers are turning their attention to a kind of non-topical classification called *sentiment classification* [9]. Research in automatic sentiment classification seeks to develop models (i.e. sentiment classifiers) for assigning category labels (positive or negative) to new documents or document segments based on a set of training documents that have been classified by domain experts.

In our previous work [8], a prototype meta search engine providing automatic sentiment classification was developed. It allows the user to specify a product name and subsequently categorizes the search results by the polarity of the desired reviews, such as *recommended* or *not recommended.* It can help the user to focus on Web articles containing either positive or negative comments. For instance, a user who is interested mainly in the negative aspects of a product (e.g. a digital camera) can look at Web articles under the negative review category.

D.H.-L. Goh et al. (Eds.): ICADL 2007, LNCS 4822, pp. 65–74, 2007.

For effective sentiment classification, non-review documents should first be filtered out so that further classification (i.e. sentiment classification) can focus on product review documents. We define a review document as a page that contains only a single review, since the sentiment classifier is designed to classify one review at a time. The Web search results from the meta search engine mainly consist of e-commerce Web pages selling the product, product specifications from manufacturing companies, on-line product review documents, etc.

This paper focuses on the filtering of product review documents from various documents in the Web search results. In this study, only snippets and not full text documents are used in the filtering process since full text documents would need more processing time. Determining whether a snippet is a review or non-review document is a challenging task, since the snippet usually does not contain many useful features for identifying review documents.

In the following sections, section 2 discusses related works of automatic text classification, section 3 presents our approaches for review classification and, finally, section 4 discusses future work and conclusion.

2 Related Works

Research in *automatic text classification* seeks to develop models for assigning category labels to new documents based on a set of training documents. For classification, documents are represented as sets of features from their content and style, called *document vectors*. Most studies of automatic text classification have focused on either "topical classification" classifying documents by subject or topic (e.g. *education* vs. *entertainment*), or "genre classification" classifying documents by document styles (e.g. *fiction vs. non-fiction*). A detailed introduction to automated text classification has been provided by Sebastiani [11].

Determining whether a snippet is a review or non-review document is considered as a genre classification problem. Documents (i.e. snippets) discussing the same topic (e.g. a digital camera) can be classified into different genres, such as *product specification* or *product review*. Compared to topical classification which mainly utilizes text features of documents, genre classification uses various document style features, such as *part-of-speech* and linguistic features (e.g., average sentence length), in addition to text features to analyze how documents are described. However, our study does not use document style features because snippets are too short to analyze them. Thus our approach mainly uses text features from summary text, in addition to the URL and link title. We have performed a preliminary study [12] on this problem and this paper discusses further extensions of the study by incorporating more snippets from various products and exploring effects of various feature selection approaches, such as phrase terms (n-grams) and feature reduction, on the classification.

For genre classification, most researchers use full-text documents rather than summary documents, such as snippets. For instance, Finn, Kushmerick, and Smyth [4] investigated a genre classification, which decides whether a document presents the opinion of its author or reports facts (i.e. genre of subjectivity). C4.5, a decision tree induction program [10], was used with various text features: *bag of words* (unigrams), *part-of-speech*, and hand-crafted shallow linguistic features. For the *part-of-speech* approach, a document is represented as a vector of 36 *part-of-speech* features, expressed as percentages of the total number of words for the document. They argued that the *part-of-speech* approach provided the best accuracy when the learned classifiers were generalized from the training corpus to a new domain corpus. As another work, Kessler, Nunberg, and Schutze [7] studied automatic detection of text genre using logistic regression and neural networks techniques. The genres they investigated were *reportage*, *editorial*, *scientific/technical*, *legal*, *non-fiction*, and *fiction*.

Boese and Howe [1] investigated the effects of Web document evolution on genre classification. They reported that documents in some genres change rarely, and the genre classifier trained with an old corpus performed well on recent Web pages, with only a small drop in accuracy rates. From their study, we may argue that genre classification using document style features is not significantly affected by document evolution, compared to topical classification using text features that change over time. Since our study does not use document style features, the genre classifier learned from our study may be affected by document (snippets) evolution. Other genre classification works are well summarized in [3].

Some researchers have developed classification/clustering tools to categorize Web search results to help users locate relevant and useful information on the World Wide Web. For the classification/clustering they generally use the snippets from the search engine to provide reasonable response time to the user. Chen and Dumais [2] designed a user interface that automatically groups Web search results into predefined topical categories such as *automotive*, *local interest*, using a machine learning algorithm, SVM. The tool devised by Zeng, He, Chen, Ma and Ma [14] provides clustering of Web search results, and uses salient phrases extracted from the ranked list of documents as cluster names. For instance, with a query input, *Jaguar*, the generated cluster names are *Jaguar Cars*, *Panthera onca*, *Mac OS*, *Big Cats*, *Clubs*, and *Others*. Vivisimo (http://vivisimo.com) is an example of an operational clustering tool for Web search results. These tools, however, focused mainly on topical categorization—categorizing documents by subject or topical area.

3 Review Classification

This study is conducted with a dataset of 1200 documents (i.e. snippets). The first 800 documents are used for training and testing of the machine learning model with 10-fold cross validation. The remaining 400 documents are kept as unseen documents for final evaluation of the approaches.

A search engine, Google, is used in this study to gather the snippets of 1200 documents by submitting around 120 queries. The queries are submitted in the format of a product name followed by the key word “Review”. For example, the query “Dell XPS M1710 Review” is used for the product “Dell XPS M1710”. When the results are returned by the search engine, they are manually classified as either review or non-review documents. The manual analysis of the content is done by following the URL of the snippets and reviewing the full text. If the content is found to be a user or an expert review with ratings, it is classified as a review document. In addition, a comprehensive full-review without rating is also classified as a review document. The documents with product specifications, multiple brief reviews, list of review links or non-review-related contents are classified as non-review documents. In this study, the domain of electronic products is selected and products such as digital camera, mobile phone, MP3 player, PC, PDA, notebook, printer and monitor are included.

3.1 Machine Learning Approach

In the study, SVM [5] is used as a machine learning approach and the various components of the snippets are experimented as document features for effective classification. Unigrams (individual words), n-grams (phrases), and feature reduction are also explored to improve the accuracy. As the input to SVM, the text is converted into bags of terms (called document vectors), which are stemmed using Porter's stemming algorithm [6] after removing the stop words. Term Frequency (TF) is used as a weighting factor for the terms.

The terms are extracted from the title, the summary text, the URL and the similar pages of the snippets. Five features are experimented as document features as shown in the Tables 1 and 2: "Title", "Summary Text", "URL Domain", "URL Folder" and "Similar pages". The feature "Title" comes from the title of the snippets which is the text "*Motorola RAZR V3 Reviews*" as in the following snippet example. The feature "Summary Text" comes from the plain text below the title which is the text "*User Reviews for the Motorola RAZR V3. Plus specs, features, discussion forum, photos, merchants, and accessories*". The URL is divided into two parts, "URL Domain" which is "*www.phonescoop.com*" and "URL Folder" which is "*phones*". Finally, the feature "Similar Pages" is extracted from the snippets of the similar pages by following the link "Similar pages", which is provided by the search engine to retrieve the similar pages which are related to the current snippet.

Motorola RAZR V3 Reviews
User Reviews for the Motorola RAZR V3. Plus specs, features, discussion forum, photos, merchants, and accessories.
www.phonescoop.com/phones/user_reviews.php?phone=547 Similar pages

The test results with the first 800 documents are shown in the Table 1. When unigrams are used, the best result comes from the feature selection option S6 (Table 1), which uses the Title and the URL domain and folder. When the summary text of the snippets is included as a document feature, it reduces the accuracy of the classification because the summary text can come from any part of the full text and, thus, it distracts the SVM model when it comes to classification. When the similar pages are included in the text, the accuracy of classifier also decreases significantly because the similar pages are mainly product information pages, but not product review documents.

The machine learning approach produces better results when using n-grams than when using just unigrams for model building and classification. In the study, the n-grams consist of unigrams, bigrams, and trigrams. For instance "review, "full review", and "unbiased review document" are valid terms in the n-grams. The best result comes from the S7 option, which uses n-grams of the Title and the URL domain and folder without any feature reduction. The options using the same features as S7 but with feature reduction, S8 and S9, do not perform better than S7 in terms of accuracies but the computation cost is significantly reduced since around 10,000 n-gram terms are reduced to 3,000 terms. For feature reduction, information gain and chi-square values are used [13].

Terms such as "unbiased review", "comparison" and "comprehensive", and domain names such as "reviews.cnet.com", "review.zdnet.com" and "www.pcmag.com" occur very frequently in the URL and text of the review snippets. For the non-review snippets, terms such as "price", "spec" and "shop" occur commonly.

Table 1. The SVM approach using 10-Fold Cross Validation with 800 documents

ID	Features					n-grams	Feature Reduction		Accuracy
	Title	Summary Text	URL Domain	URL Folder	Similar Pages		Chi Square	Information Gain	
S1	Y	Y	Y	Y	Y				74.25%
S2	Y	Y	Y	Y					79.14%
S3	Y	Y	Y						77.92%
S4	Y	Y							73.90%
S5	Y		Y						85.43%
S6	Y		Y	Y					86.07%
S7	Y		Y	Y		Y			**87.08%**
S8	Y		Y	Y		Y	Y		86.70%
S9	Y		Y	Y		Y		Y	86.58%

The test results of the machine learning approach when tested with the unseen 400 documents are shown in the Table 2. The SVM model which is built by training with the initial dataset of the 800 documents is tested for the unseen documents. The accuracies of the tests using n-gram terms are consistently better than the accuracies of the tests using unigram terms.

Table 2. The SVM approach with 400 unseen documents

ID	Features					n-grams	Feature Reduction		Accuracy
	Title	Summary Text	URL Domain	URL Folder	Similar Pages		Chi Square	Information Gain	
S1	Y	Y	Y	Y	Y				59.35%
S2	Y	Y	Y	Y					78.80%
S3	Y	Y	Y						75.81%
S4	Y	Y							71.32%
S5	Y		Y						81.55%
S6	Y		Y	Y					81.55%
S7	Y		Y	Y		Y			**83.04%**
S8	Y		Y	Y		Y	Y		82.79%
S9	Y		Y	Y		Y		Y	82.79%

3.2 Heuristic Approach

A heuristic approach is also developed to experiment if a simpler heuristic approach with semi-automatically constructed domain knowledge can perform as good as the machine learning approach. In contrast to the machine learning approach which uses thousands of terms, this approach uses only hundreds of terms for classification. It is based on the review and non-review lists of n-gram terms which are constructed by analyzing the 800 snippets. Through the analysis, meaningful terms with high information gain or chi-square values are taken into consideration. Also manually constructed terms are added to the lists. These lists then are used to distinguish the review and non-review documents using the title, the summary text and the URLs of the snippets. Some sample entries of the lists are shown in Table 3.

Table 3. Sample of n-gram terms for the heuristic approach

	Review	**Non-review**
Title	unbiased review * editor review * full review * review by * mobile review * guide exclusive comparison overview good (N$_{\text{Review-Title}}$ =25 entries)	shop * price * free * software download * best price * introduce service spec supply review image (N$_{\text{Non-Review-Title}}$ =25 entries)
Summary Text	unbiased review * editor review* cute * beauty * coverage * exclusive comprehensive footage guide compare compare editorial (N$_{\text{Review-Text}}$ =25 entries)	shop* share* sell* merchant * buyer * review tip photographic review article review write review image (N$_{\text{Non-Review-Text}}$ =25 entries)
URL	review.zdnet.com www.infosyncworld.com www.mobile-review.com www.trustedreviews.com asia.cnet.com www.pocket-lint.co.uk www.letsgodigital.org www.mobile-phones-uk.org.uk laptopmag.com cellphones.about.com (N$_{\text{Review-URL}}$ =25 entries)	www.amazon.com www.livingroom.org.au www.reviewcentre.com www.imobile.com.au mobilementalism.com www.dpreview.com www.steves-digicams.com www.letsgomobile.org www.pricerunner.com www.phonedog.com (N$_{\text{Non-Review-URL}}$ =25 entries)

*: indicates manually constructed terms

For the Title's review and non-review lists, n-gram terms with high information gain or chi-square values are collected first from the titles of the snippets, and the terms which appear mainly in review documents are added into the review list while those which appear more in non-review documents are added into the non-review list.

The distinguishing terms such as “editor review” for review titles and “software download” for non review titles are also included although they may not have high information gain or chi-square values, or may not appear in automatically generated n-grams. The review and non-review lists of the Summary Text are constructed in the same way. For the URL lists, only terms with high information gain or chi-square values are added into either the review or the non-review list.

The following mathematical formula is used for the heuristic approach. $W_{\mathrm{Heuristic}}$ represents the classification output value where the positive or negative value indicates a review document or a non-review document respectively. The parameters α, β and γ are weights on the title, summary text and URL. Based on our trial and error analysis, their optimal values are 0.3, 0.1 and 0.5 respectively. It shows that the URL is given a higher weight than others. If a snippet comes from a known review site, it is most likely to be a review document regardless of the other terms in the snippet.

$$W_{\mathrm{Heuristic}} = \alpha \,.\, W_{\mathrm{H.Title}} + \beta \,.\, W_{\mathrm{H.Summary}} + \gamma \,.\, W_{\mathrm{H.URL}}$$

$$W_{\mathrm{H.Title}} = \sum_{i=1}^{N_{\mathrm{Re}\,view\text{–}Title}} TF_i - \sum_{j=1}^{N_{Non\text{–}\mathrm{Re}\,view\text{–}Title}} TF_j$$

$$W_{\mathrm{H.Summary}} = \sum_{i=1}^{N_{\mathrm{Re}\,view\text{–}Text}} TF_i - \sum_{j=1}^{N_{Non\text{–}\mathrm{Re}\,view\text{–}Text}} TF_j$$

$$W_{\mathrm{H.URL}} = \begin{array}{ll} +1 & \textit{If URL} \in \textit{Review-URL List} \\ -1 & \textit{If URL} \in \textit{Non-Review-URL List} \\ 0 & \textit{Else} \end{array}$$

The heuristic approach is tested with the 800 snippets and the heuristic approach performs comparatively well as the machine learning approach, with only a small drop in accuracy rates. The best accuracy is achieved when the title, the summary text and the URL of the snippets are used together (H4 in Table 4).

Table 4. The heuristic approach with 800 documents

ID	Title	Summary Text	URL	Accuracy
H1	Y			65.37%
H2		Y		66.44%
H3			Y	77.41%
H4	Y	Y	Y	**84.08%**

Table 5 shows the results of the heuristic approach when tested with the unseen documents. The heuristic approach has lesser computation cost yet it performs quite close to the machine learning approach. The heuristic approach using only the URL shows significantly lower accuracy when it is tested with the 400 unseen documents (H3 in Table 5) than when it is tested with the initial 800 documents (H3 in Table 4). This is because some URLs from the 400 unseen documents do not match with URL terms in the URL lists collected from the 800 documents, and the URL lists alone are not enough to determine review documents.

Table 5. The heuristic approach with unseen 400 documents

ID	Title	Summary Text	URL	Accuracy
H1	Y			61.60%
H2		Y		64.59%
H3			Y	65.34%
H4	Y	Y	Y	**79.05%**

3.3 Hybrid Approach

This is to experiment and find out if the hybrid approach which is a combination of both the machine learning and the heuristic approach can be employed to improve the outcome of the classification. The results of the experiment show that the hybrid approach performs better than the machine learning and the heuristic approach though not very significantly. The outcome of the hybrid approach (W_{Hybrid}) is calculated by combining the outcomes of the SVM approach (W_{SVM}) and the heuristic approach ($W_{Heuristic}$). The parameters λ and μ are used to fine-tune the outcome. For this initial evaluation, the values are set as 1 to equally weigh the two approaches.

$$W_{Hybrid} \quad = \quad \lambda \, . \, W_{SVM} + \lambda \, . \, W_{Heuristic}$$

When testing with the 800 documents, the best options, HB2 and HB4 (Table 6), achieve an accuracy of 89.30%. HB2 is a combination of S7 (Table 1), a machine learning approach using n-grams without feature reduction, and H4 (Table 4), a heuristic approach using the title, summary text and the URL. On the other hand, HB4 is a combination of S9 (Table 1), a machine learning approach using n-grams with feature reduction using chi-square, and H4 (Table 4). It is also observed that the test results of hybrid approaches HB3 and HB4 which utilize feature reduction, perform slightly better than S7 (Table 1), a machine learning approach without any feature reduction.

Table 6. The hybrid approach with 10-fold cross validation

ID	SVM Option	Heuristic Option	Accuracy
HB1	S6	H4	87.81%
HB2	S7	H4	**89. 30%**
HB3	S8	H4	89.17%
HB4	S9	H4	**89.30%**

The same approaches are then tested with the 400 unseen documents. The hybrid approach generally performs better than the machine learning approach or the heuristic approach alone as shown in the Table 7. The best result comes from HB2, which is a combination of a machine learning approach using n-gram terms without feature reduction and a heuristic approach using the title, summary text and the URL.

Table 7. The hybrid approach with 400 unseen documents

ID	SVM Option	Heuristic Option	Accuracy
HB1	S6	H4	83.79%
HB2	S7	H4	**84.79%**
HB3	S8	H4	84.29%
HB4	S9	H4	84.04%

3.4 Error Analysis

When analyzing the errors encountered by the approaches, it is observed that some of the errors are inevitable mainly because the classification is done based on just snippets which are relatively very short and with incomplete sentences. In such a scenario, even human classifiers will not be able to distinguish snippets of the non-review documents from those of review documents without looking at the full texts.

The following is an example snippet of a review document with a rating which is wrongly classified by both approaches as a non-review document because the snippet does not have enough terms related to review documents.

URL: *www.vnunet.com/personal-computer-world/hardware/2187326/lexmark-c534dn*
Title: Review: Lexmark C534dn laser printer - vnunet.com
Summary Text: Fast monochrome and color printing in one compact device.

The following is an example snippet of a non-review document which is wrongly classified by both approaches as a review document because it has some terms related to review documents.

URL: *mobilereviews.o2.co.uk/userreview/home*
Title: Mobile reviews - Mobiles & Tariffs - O2
Summary Text: Welcome to O2. Read and write reviews on the latest mobile phones.

When testing the hybrid approach with the unseen 400 documents, it is observed that 28 out of 88 errors made by the heuristic approach are corrected by the machine learning approach. On the other hand, 7 out of 67 errors made by the machine learning approach are corrected by the heuristic approach. We believe that in the hybrid approach the machine learning and heuristic approaches use different logics and complement each other to give better performance.

4 Discussion and Conclusion

In conclusion, the machine learning approach using the SVM performs the best with just the title and URL (domain and folder names) of the snippets as phrase terms (n-grams) for classifying product review documents. When feature reduction techniques such as Information Gain and Chi-square statistics are applied, computation cost is reduced with only a slight drop in accuracies. The heuristic approach which mainly makes use of domain knowledge performs as well as the machine learning approach in our experiments. The heuristic approach with the title, URL and the summary text of the snippets gives the best performance. The hybrid approach which makes use of both machine learning techniques and domain knowledge performs slightly better than the machine learning approach alone.

The limitation of this study is that it is only conducted for the electronic product review documents through a search engine and it may not work consistently for other domains. For future work, more evaluations and experiments will be carried out with larger datasets and a wider range of products using various Web search engines. Furthermore, the heuristic approach can be improved by including more meaningful and distinguishing terms and enhancing the formula to achieve better performance for the unseen documents.

References

1. Boese, E.S., Howe, A.E.: Effects of Web Document Evolution on Genre Classification. In: Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM 2005), Bremen, Germany, pp. 632–639 (2005)
2. Chen, H., Dumais, S.T.: Bringing Order to the Web: Automatically Categorizing Search Results. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2000), pp. 145–152 (2000)
3. Choi, B., Yao, Z.: Web Page Classification, Foundations and Advances in Data Mining, Studies in Fuzziness and Soft Computing, vol. 180, pp. 221–274. Springer, Berlin (2005)
4. Finn, A., Kushmerick, N., Smyth, B.: Genre classification and domain transfer for information filtering. In: Crestani, F., Girolami, M., van Rijsbergen, C.J.K. (eds.) Advances in Information Retrieval. LNCS, vol. 2291, pp. 353–362. Springer, Heidelberg (2002)
5. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proceedings of 10th European Conference on Machine-learning, Chemnitz, Germany, April 21-24, pp. 137–142 (1998)
6. Jones, K.S., Willet, P.: Readings in Information Retrieval. Morgan Kaufman, San Francisco (1997)
7. Kessler, B., Nunberg, G., Schutze, H.: Automatic detection of text genre. In: Proceedings of the Eighth Conference on European Chapter of the ACL (Association for Computational Linguistics), pp. 32–38 (1997)
8. Na, J.-C., Khoo, C., Chan, S., Hamzah, N.B.: A sentiment-based search in digital libraries. In: Proceedings of Joint Conference on Digital Libraries 2005 (JCDL 2005), Denver, pp. 143–144 (2005)
9. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine-learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, July 6-7, pp. 79–86 (2002)
10. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufman, San Francisco (1993)
11. Sebastiani, F.: Machine-learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
12. Thet, T.T., Na, J.-C., Khoo, C.S.G.: Filtering Product Reviews from Web Search Results. In: Proceedings of ACM Symposium on Document Engineering (DocEng 2007), Winnipeg, Canada (August 28 - 31, 2007)
13. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of the fourteenth International Conference on Machine Learning, pp. 412–420. Morgan Kaufmann, San Francisco (1997)
14. Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., Ma, J.: Learning to Cluster Web Search Results. In: Proceedings of the 27th Annual International ACM SIGIR Conference, Sheffield, UK, pp. 210–217 (2004)