PROOF COVER SHEET

Author(s): Ying Liu
Article title: Discovering contextual tags from product review using semantic relatedness
Article no: TJCI 895966
Enclosures: 1) Query sheet
2) Article proofs

Dear Author,

1. Please check these proofs carefully. It is the responsibility of the corresponding author to check these and approve or amend them. A second proof is not normally provided. Taylor & Francis cannot be held responsible for uncorrected errors, even if introduced during the production process. Once your corrections have been added to the article, it will be considered ready for publication.

Please limit changes at this stage to the correction of errors. You should not make trivial changes, improve prose style, add new material, or delete existing material at this stage. You may be charged if your corrections are excessive (we would not expect corrections to exceed 30 changes).

For detailed guidance on how to check your proofs, please paste this address into a new browser window: http://journalauthors.tandf.co.uk/production/checkingproofs.asp

Your PDF proof file has been enabled so that you can comment on the proof directly using Adobe Acrobat. If you wish to do this, please save the file to your hard disk first. For further information on marking corrections using Acrobat, please paste this address into a new browser window: http://journalauthors.tandf.co.uk/production/acrobat.asp

2. Please review the table of contributors below and confirm that the first and last names are structured correctly and that the authors are listed in the correct order of contribution. This check is to ensure that your name will appear correctly online and when the article is indexed.

Sequence	Prefix	Given name(s)	Surname	Suffix
1		Soon Chong Johnson	Lim	
2		Shilong	Wang	
3		Ying	Liu	

Queries are marked in the margins of the proofs, and you can also click the hyperlinks below.

AUTHOR QUERIES

General points:

1. **Permissions:** You have warranted that you have secured the necessary written permission from the appropriate copyright owner for the reproduction of any text, illustration, or other material in your article. Please see http://journalauthors.tandf.co.uk/permissions/usingThirdPartyMaterial.asp.
2. **Third-party content:** If there is third-party content in your article, please check that the rightsholder details for re-use are shown correctly.
3. **Affiliation:** The corresponding author is responsible for ensuring that address and email details are correct for all the co-authors. Affiliations given in the article should be the affiliation at the time the research was conducted. Please see http://journalauthors.tandf.co.uk/preparation/writing.asp.
4. **Funding:** Was your research for this article funded by a funding agency? If so, please insert 'This work was supported by <insert the name of the funding agency in full>', followed by the grant number in square brackets '[grant number xxxx]'.
5. **Supplemental data and underlying research materials:** Do you wish to include the location of the underlying research materials (e.g. data, samples or models) for your article? If so, please insert this sentence before the reference section: 'The underlying research materials for this article can be accessed at <full link>/ description of location [author to complete]'. If your article includes supplemental data, the link will also be provided in this paragraph. See <http://journalauthors.tandf.co.uk/preparation/multimedia.asp> for further explanation of supplemental data and underlying research materials.
6. The **CrossRef database** (www.crossref.org/) has been used to validate the references. Mismatches will have resulted in a query.

AQ1	Please provide the missing city for affiliation 'b'.
AQ2	Please provide missing city for Ref. [15].
AQ3	Ref. [22], the journal title has been replaced using data from CrossRef. Please check that this has been done correctly.

How to make corrections to your proofs using Adobe Acrobat/Reader

Taylor & Francis offers you a choice of options to help you make corrections to your proofs. Your PDF proof file has been enabled so that you can edit the proof directly using Adobe Acrobat/Reader. This is the simplest and best way for you to ensure that your corrections will be incorporated. If you wish to do this, please follow these instructions:

1. Save the file to your hard disk.

2. Check which version of Adobe Acrobat/Reader you have on your computer. You can do this by clicking on the “Help” tab, and then “About”.

   If Adobe Reader is not installed, you can get the latest version free from http://get.adobe.com/reader/.

3. If you have Adobe Acrobat/Reader 10 or a later version, click on the “Comment” link at the right-hand side to view the Comments pane.

4. You can then select any text and mark it up for deletion or replacement, or insert new text as needed. Please note that these will clearly be displayed in the Comments pane and secondary annotation is not needed to draw attention to your corrections. If you need to include new sections of text, it is also possible to add a comment to the proofs. To do this, use the Sticky Note tool in the task bar. Please also see our FAQs here: http://journalauthors.tandf.co.uk/production/index.asp.

5. Make sure that you save the file when you close the document before uploading it to CATS using the “Upload File” button on the online correction form. If you have more than one file, please zip them together and then upload the zip file.

   If you prefer, you can make your corrections using the CATS online correction form.

Troubleshooting

Acrobat help: http://helpx.adobe.com/acrobat.html
Reader help: http://helpx.adobe.com/reader.html

Please note that full user guides for earlier versions of these programs are available from the Adobe Help pages by clicking on the link “Previous versions” under the “Help and tutorials” heading from the relevant link above. Commenting functionality is available from Adobe Reader 8.0 onwards and from Adobe Acrobat 7.0 onwards.

Firefox users: Firefox’s inbuilt PDF Viewer is set to the default; please see the following for instructions on how to use this and download the PDF to your hard drive: http://support.mozilla.org/en-US/kb/view-pdf-files-firefox-without-downloading-them#w_using-a-pdf-reader-plugin

TJCI 895966
27 February 2014
Initial
CE: AS QA: JV
Coll: QC:

Journal of Industrial and Production Engineering, 2014
http://dx.doi.org/10.1080/21681015.2014.895966

Discovering contextual tags from product review using semantic relatedness

Soon Chong Johnson Lim[a], Shilong Wang[b] and Ying Liu[a]*

[a]*Faculty of Technical and Vocational Education, University Tun Hussein Onn Malaysia, Johor, Malaysia;* [b]*The State Key Lab of Mechanical Transmission, Chongqing University, China*

AQ1

(*Received October 2013; revised January 2014; accepted February 2014*)

In the design community, while a number of studies that have focused on studying product reviews in various design analysis perspectives, contextual annotation of identified terms (e.g. product features) has not been fully explored. This paper proposed a learnable approach towards discovering contextual tags from product reviews. A ranking algorithm, FacetRank, is proposed to rank important key terms along with an approach to discover contextual annotation of the terms from review documents. The evaluation of our proposal is performed using two annotated corpus to examine our algorithm's contextual tagging performance. A case study using a small collection of laptop reviews is also reported to showcase how our algorithm can be applied towards product feature understanding and multi-faceted product ontology development. Finally, we conclude this paper with some indications for future work.

Keywords: product design; contextual annotation; product online review; semantic relatedness

1. Introduction

Nowadays, the rapid development of Internet technologies and the advent of Web 2.0 applications, such as online forums, e-commerce portals, and blogs have allowed Internet users to easily share their views online. From a product user perspective, it has become a common scene for customers to write reviews related to their feature preferences, views, or actual user experience associated with a product. Whether the opinions come from an average customer or from a professional user point of view, the availability of these reviews has presented huge potential for mining useful product information. However, due to the sheer amount of product reviews available in various distributed sources, it is not possible for product designers to gather and analyze all of these reviews manually. Thus, automated processing of reviews is a more feasible and practical approach towards mining interesting patterns from these reviews.

Previously, there are a number of studies that focused on analyzing product reviews. Works in the area of opinion mining or sentiment analysis [7,16], product review summarization [12,32], and determining design review helpfulness [9] are among the notable ones. While most of the current works emphasized on identifying semantic orientation of reviews towards certain product feature or generating review summary according to elicited topical words, a common task in these works involves identifying the key terms from documents. However, contextual annotation of key terms, i.e. the task of discovering descriptive tags for these identified terms, has not been fully explored. Technically, a key term usually represents salient product feature identified from review and its contextual annotation provide more information on the feature under various facet, such as preferences, emotions, or usage experience. For instance, in reviews about camera, contextual annotation enables the suggestion of related product features (e.g. "lens", "focus") with topical word of interest ("picture quality"), or discovery of other contextually similar product features ("video recording"). For product designers, this allows better understanding of component associations, usage experience, emotions, etc. from various customer or professional viewpoints.

Realizing the benefits of contextual product information, the research issue is as follows: how can we automatically learn relevant, contextual tags that correspond to a query from a collection of product reviews? To the best of our knowledge, there are relatively few studies that have explored on contextual tags discovery and particularly, on how its application can benefit the design community. Technically, we are interested to study how to generate meaningful related annotations, given a term, from multiple review documents where each document may describe multiple context of a product. In this paper, we propose an approach to automatically learn contextual tags from a collection of product review using semantic relatedness. Using this feature, an iterative ranking approach, FacetRank, is proposed for contextual discovery of tags from review documents. The rest of this paper is discussed as follows: Section 2 presents the current state of research in product review mining, key term extraction and annotation of key terms followed by a summary of issues. Section 3 describes our proposal of discovering contextual tags from product documents, followed by evaluation of our approach using two annotated corpus in Section 4. In Section 5, we present a case study using a small corpus of laptop reviews and discuss some potential applications, and finally Section 6 concludes our work with further discussions.

*Corresponding author. Email: LiuY81@cardiff.ac.uk

TJCI 895966
27 February 2014
Initial
CE: AS QA: JV
Coll: QC:

2. Related work

2.1. *Review analysis for product design*

Product review is an increasingly valuable user-generated content towards product designers for various purposes of understanding customer concerns, rectifying product issues, etc. Unfortunately, the rapid growth of online customer reviews hinders the possibility of manual processing. Automated processing tasks of these reviews, on the other hand, is non-trivial due to the inherent features of product reviews, such as heterogeneous descriptions, distributed locations, and language ambiguity [14]. In the past several years, automated parsing and analysis of online reviews have received notable attention in major research forums, such as SIGKDD and SIGIR [67]. These works focused mainly in sentiment analysis, either positive- or negative-oriented descriptions, that are related to various product features.

In relation, the application of mined information from review documents towards product design has just gained considerable attention in more recent years. From literature, among the notable ones are: Lee [11] attempted a hierarchical, two-staged process that includes association rules for assessing changing user needs based on online reviews. Loh et al. [16] proposed a hybrid opinion extraction framework that extracts features and predict semantic orientation of expressed opinions from free text. Ling et al. [12], attempted the issue of generating multi-faceted semantic overviews of arbitrary topics in a text collection for a query term. Their work focused on generating faceted models using a few user-supplied keywords that describe user-interested facets (e.g. cost). Zhan et al. [32] proposed an approach to automatically summarize multiple customer reviews based on their internal topic structure. Final summary of multiple reviews is then created using the topic structure that is ranked according to importance. Jin and Liu [9] studied the quality of product reviews and the correlation between the ratings by customers and those by designers in order to determine review helpfulness.

2.2. *Review analysis for product design*

A term refers to either a word (i.e. single word) or a phrase that captures the main topics discussed in a document [23]. While the word “keyphrase” is widely used in literature to indicate a salient phrase or a word, this study uses “key term” to avoid this confusion. In literature, the process of key term extraction is performed in two steps: candidate term extraction and key term selection. The first task aims to extract a list of potential terms. In the second task, significant candidates are selected based on certain document features. Generally, key term extraction can be viewed based on their learning approaches: supervised and unsupervised. Supervised approaches require labeled terms to train classifiers in order to correctly tag unseen or new key terms. Among the studies using supervised approaches are *GenEx* [27], *Kea* [29], Nguyen and Kan [23], and *Maui* [18]. These studies have applied a number of features, such as term frequency (TF), inverse term frequency (IDF), and position of first occurrence from a small collection of labeled key terms for classifier training. Unsupervised approaches, on the other hand, do not require training corpora. In comparison, unsupervised approaches adopted strategies, such as POS tagging [20,30] and shallow parsing [3] to improve the quality of terms extracted.

For key terms selection, majority of the supervised approaches use probabilistic classifier such as the Naïve-Bayes classifier to perform selection of key terms. While the number of features considered for training classifiers differ among the studies reviewed, features that are commonly considered consist of document-related features (e.g. TF × IDF), corpora-related features (e.g. keyphraseness), and measures of semantic relatedness (e.g. co-occurrence). It is noted that a combination of these multiple features can improve the identification of key terms [18]. In contrast, unsupervised approaches use similar features that are formulated using a scoring or ranking function, e.g. frequency-based weighted score [3] and corpus-based scoring method by Wu et al. [30]. Mihalcea and Tarau [20] proposed TextRank, an iterative ranking algorithm based on PageRank [4] using co-occurrence statistics between words.

2.3. *Annotation of key term*

The aim of annotating key terms is to discover contextual and meaningful description of a key term and its relationship with other key terms. One of the approaches to tackle this issue is through statistical approaches. Previously, researchers have deployed techniques, such as closed frequent patterns [24] or maximal frequent sequences [1] in order to highly summarize similar key term patterns into a general pattern that provides better information beyond word support. Another stream of researchers have tried document summarization techniques [10,31] in order to discover meaningful topical phrases that describe a document. While these methods can successfully reduce the redundancy of key terms extracted and present to users only the meaningful ones, the further annotation of key terms is still very much limited to statistical information (e.g. support, significance) and not meaningful key terms that are semantically related. In order to annotate meaningful key terms with semantically related terms, another approach is to use pre-defined controlled vocabulary list, such as WordNet [21] or some other domain-specific thesauri. Under this perspective, this issue can be viewed as a problem of term or category assignment. Example of related studies in term assignment are medical text indexer [2] and medical vocabulary-based topic generation [17] that emphasize on terms matching. One of the drawbacks of this approach is that building and maintaining a domain-specific controlled vocabulary requires considerable amount of efforts and often are limited to certain domain. Another disadvantage lies in the need to prepare large training sets for machine learning-based matching (e.g. classification), which limits its effectiveness over untrained key terms. Realizing this limitation, later studies [18] have applied open-domain corpus such as Wikipedia. Wikipedia provides a better solution as a user-contributed source of domain terminology that is collaboratively maintained. While open-domain data such as Wikipedia may be a better option, it is only often restricted to known examples and may not be applicable for new entry of key terms.

2.4. *Summary of literatures*

From our literature survey, current works in product design domain are mainly focused on identifying sentiment of reviews towards certain product feature and review summarization. Among these studies, identifying salient product features or topical words automatically from product review collections is a common processing task regardless of the analysis perspective. In the technical perspective of key term extraction, the main disadvantage of supervised approaches is the requirement of labeled training examples, which are not always available in practice. While the inclusion of several additional new features can be helpful for training better classifiers, the issue lies in the best mix of these features where it can be corpus dependant. On the other hand, unsupervised approaches are independent of the features of a particular document set and can be applied even without training data-set. This feature provides greater flexibility over supervised approaches. However, unsupervised approaches also often produce ill-formed key terms without significant meanings.

In general, we observed that the issue of contextual annotation of identified tags has not been fully investigated. To the best of the author's knowledge, Mei et al. [19] is the first research group that has formally addressed the issue of semantic annotation of frequent patterns. They proposed a framework and dictionary analogy where semantic annotation of a frequent pattern consists of context models, a set of representative transactions and a set of semantically similar pattern. For studies related to product reviews, works by Ling et al. [12] on generating faceted overview of topical words in review is a closer example. Nevertheless, similar to the outcome by Zhan et al. [32], their work is more focused on generating a summarized form of review and not contextually related term associations that is intended in this study.

In this paper, we proposed an approach towards discovering contextual annotations that are relevant to a term. An unsupervised key term extraction approach that utilizes semantic relatedness information of domain-specific corpus is preferred while avoiding the requirement of training examples. The idea is key term extraction of a document using its own semantic relatedness feature. Based on this feature, a suitable ranking approach is proposed to determine important key terms. For semantic tags discovery for key terms, the dictionary analogy as proposed by Mei et al. [19] is adopted. We attempt to generate contextual tags for a key term using similar analogy through building contextual/faceted model for the key term and retrieve contextual tags through comparing faceted models of potential key terms.

3. Discovering contextual tags from product reviews

This section details our proposal for discovering contextual tags from product reviews using semantic relatedness. Semantic relatedness generally refers to the degree to which a given pair of terms is related. Computationally, a semantic relatedness measure serves as a feature metric to indicate the strength of these relationships bonding. There are various semantic relatedness measures that have been proposed, from statistical-based co-occurrence [22] and mutual information [5] of terms to hierarchy-based information content [26] and features matching [28]. In this study, the strength of semantic relatedness is defined using pointwise mutual information (PMI) [5], a commonly used metric in information theory to measure the strength of association between two terms based on co-occurrence probability. An assumption is made where candidate terms that occur together are semantically associated. The formula for calculating PMI between terms is indicated in Equation (1), where t represents a candidate term.

$$\mathrm{PMI}(t_1, t_2) = \log \frac{\Pr(t_1, t_2)}{\Pr(t_1) \cdot \Pr(t_2)} \tag{1}$$

Using PMI measures, a suitable ranking algorithm is required to judge the importance of each term based on these associations. In this study, the ranking problem is modeled using a graph-based ranking algorithm that is adapted from the PageRank algorithm, originally introduced by Brin and Page [4]. The original PageRank is modified to form FacetRank (*FR*), a ranking algorithm using semantic relatedness between terms as in Equation (2). PMI measure in our context is non-directed. Thus, $in(V)$ and $out(V)$ are similar representations of undirected term associations. The parameter d in Equation (2) is a damping factor that is usually set at 0.85[4].

$$FR(V_i) = (1-d) + d \cdot \sum_{V_j \in in(V_i)} \frac{\mathrm{PMI}_{i,j} \cdot FR(V_j)}{\sum_{V_k \in out(V_j)} \mathrm{PMI}_{j,k}} \tag{2}$$

3.1. *Key terms extraction*

Key terms extraction can be divided into two sub-tasks: candidate term extraction and key term selection. In this study, candidate term extraction process follows the pre-processing steps as proposed in *Kea* algorithm [29]. The use of linguistic features, e.g. POS tagging and shallow parsing, is not considered in this study despite its popularity in unsupervised approaches. This approach is taken to avoid the use of extra tagging and selection process (which slows down the overall performance, especially on large, heterogeneous product review documents) and the use of specific linguistic corpus for shallow parsing.

For *Kea* algorithm, pre-processing steps are simple and heuristic-based, enabling a faster pre-processing process. The pre-processing steps start with input cleaning, where texts are tokenized with several text modifications (e.g. apostrophes removal, phrase boundaries identification, etc.). Three simple but effective heuristic rules [29] are applied:

(i) Candidate phrases are limited to a certain maximum length (usually three words).
(ii) Candidate phrases cannot propose names (i.e. single words that only ever appear with an initial capital).
(iii) Candidate phrases cannot begin or end with a stop-word.

The stop-word list used in this study contains 425 words in nine syntactic classes (e.g. conjunctions, articles, particles, etc.) similar to the one used in Witten et al. [29]. Candidate terms are case-folded (i.e. to lower case) and stemmed using Porter stemmer [25] to discard any suffixes. The original form of candidate terms, however, is still retained for presentation purpose. Stemming is applied for comparison between candidate key terms and actual gold standard matching during evaluation. Candidate terms are then ranked using FacetRank for key terms selection. Statistical-based semantic relatedness metric in this study provides a flexible approach towards key terms extraction using different semantic relatedness information. The overall key terms extraction process is illustrated in (Figure 1).

3.2. *Contextual annotation of key terms*

Upon the selection of candidate key terms, the next task is to generate contextual annotations of a key term. In a single document, different combination of terms can suggest different facets of interest. A facet can be referred to as a specific point of view that is of interest to a user regarding a domain of consideration. For instance, for product review documents that describe a digital camera model, possible product features extracted from this document can be "flash," "lens," "image quality," "image processor," "auto focus," and "intelligent lighting." Under the product component context, the entities "flash," "lens," and "image processor" represent the generic camera components; function wise, the phrase "auto focus" represents a camera function that is associated with "lens;" from a professional photographer's perspective, "image quality" can be associated with both "image processor" and "lens." This is as illustrated in (Figure 2).

The aim of our contextual annotation is to discover all these possible associations. In this study, we suggest associations at smaller granularity of sentence level where we assumed that key terms contained in a sentence are semantically associated and describe a particular facet. As shown in (Figure 2), such a group of key terms corresponding to original document sentences is named entity set (ES). The algorithm for generating ES is as shown in (Table 1(a)). The algorithm produces a collection of entity set, ES by comparing each extracted entity, $e \in E$ with every sentence in a document, $d \in D$. Using FacetRank, each key term that is contained in an *es* is iteratively ranked. Ranked ES is named as faceted unit (FU), the basic building block of faceted modeling that describes an entry key term. For a FU, the highest ranked term is selected as faceted indicator, a representative key term that indicates the facet of an FU. In order to reduce redundant FUs, clustering is performed to aggregate similar FUs together. Hierarchical agglomerative clustering (HAC) [8] is proposed for this purpose with algorithm as shown in (Table 1(b)). For this, similarity between two FUs is determined using Euclidean distance measure. From Equation (3), note that f_i and g_i are faceted rank values for common key terms. For uncommon key terms, one of the values of either f_i or g_i is assigned zero. For similarity between clusters, single-linkage scheme is used where distance between two cluster pairs is the smallest distance between two FUs in both clusters. A maximum of two-faceted indicators from each representative *fu* are aggregated as a cluster's concepts.

$$d(fu_1, fu_2) = \sqrt{\sum_{i=1}^{n} (f_i - g_i)^2} \qquad (3)$$

For contextual annotation, given a term *t*, the task of annotation is the process of selecting representative FUs where *t* occurs at least once in the sentences corresponding to the FUs. Once related FUs are selected, the corresponding faceted indicators of these FUs are identified to determine faceted weight, a measure for strength of association between *t* and related faceted indicators. In this study, PMI is used as faceted weight. Consequently, a faceted model for a term is defined via faceted indicator with corresponding faceted weights. Contextual annotation for the term consists of faceted indicators, associated sentences, and other related terms that have similar faceted models with that of the term's. In this case, related terms can be an important terminology pre-determined by user or few important top key terms from each document. For comparison, Let FM_{t1} and FM_{t2} denote the two faceted models for query terms t_1 and t_2, respectively. The two query terms are associated if the difference or distance between their faceted models, $diff(FM_{t1}, FM_{t2}) \leq k$, where k is a user defined threshold value. While there are

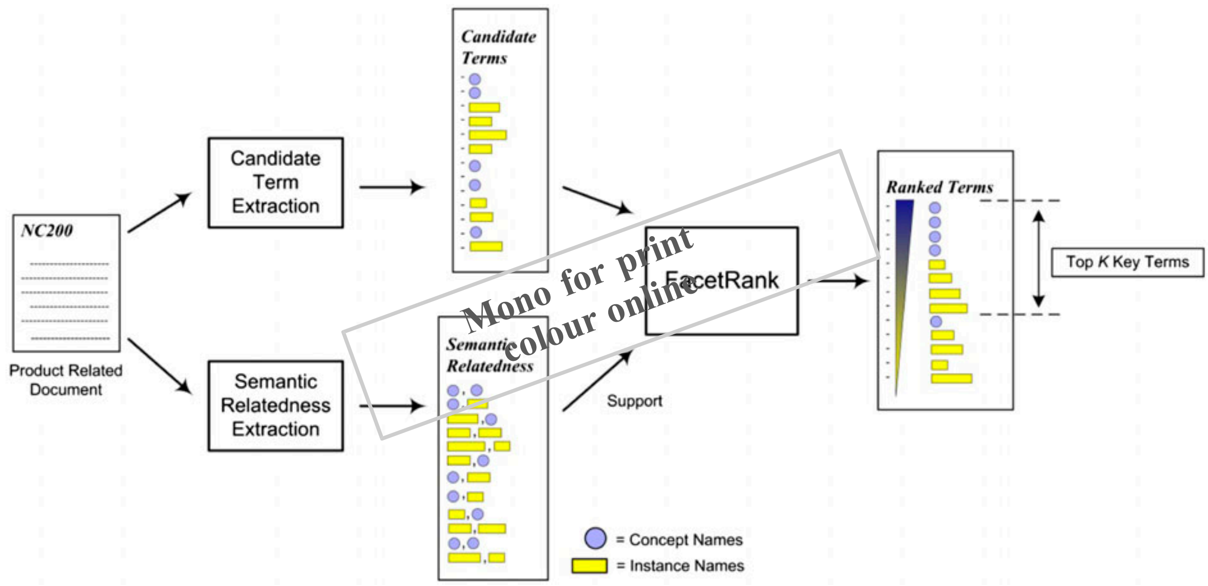

Figure 1. Overview of key term extraction process using FacetRank.

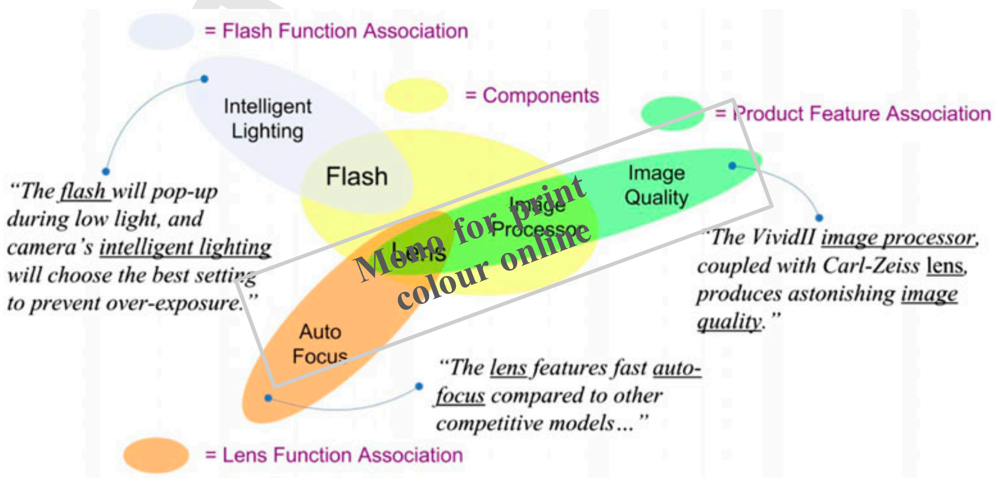

Figure 2. Faceted associations at sentence level.

a number of different similarities or distance measures that can be applied, the simplest Euclidean distance is applied in this study that only considers partial matches (i.e. only common key terms are considered).

4. Performance evaluation

4.1. *Key terms extraction*

In order to evaluate the effectiveness of key terms extraction, the use of annotated corpus is essential. While there are a number of corpus choices for evaluation purpose, we are looking for an annotated corpus that involves more than one human tagger in assigning key terms. This is essential as key terms assigned by single human tagger may introduce biases. This issue can be mitigated via the involvement of a number of human taggers, where the most representative key terms for a document can be determined via mutually agreed key terms. For such a purpose, an annotated corpus, *CiteULike-180* [18] is selected for evaluation purpose. *CiteULike-180* is a collaboratively tagged corpus, consisting of 180 publicly available documents with 332 human taggers from *citeulike.org*, and a bookmarking service for scholarly papers. According to the original authors, the corpus contains 946 tags that are agreed at least by two human taggers, resulting in accurate tag sets that contain an average of five tags per document. Following the methodology by Medelyan [18], the ground truth for a document in *CiteULike-180* consists of at least three tags on which two users have agreed. In overall, there are on average five such tags per document. A selected key term is considered "correct" if it

Table 1. Algorithms used for annotation of key terms.

(a) Entity set (ES) generation algorithm

Input: (i) Original data-set, D of m documents = $\{d_1, d_2, d_3,\ldots, d_m\}$;
(ii) A set of n extracted entities, E = $\{e_1, e_2, e_3,\ldots, e_n\}$
Output: A set of j entity sets, ES = $\{es_1, es_2, es_3,\ldots, es_j\}$

initialize empty set M, ES, SS
for each ($d_\alpha \in$ D)
  initialize sentence sets $SS_\alpha = \{ss_1, ss_2, ss_3,\ldots, ss_n\}$
  for each ($ss_u \in SS_\alpha$)
    for each ($e_v \in$ E)
      match e_v with ss_u
      if (ss_u contains e_v)
        add e_v into matched set, M
    if (M is not empty)
      add M as new entity set, $es \in$ ES
      update ES with es
output ES
end

(b) HAC clustering algorithm

Input: (i) A set of j Faceted Units, FU = $\{fu_1, fu_2, fu_3,\ldots, fu_j\}$
(ii) Clustering Threshold, k where k~[0,1], $t \in$ R
Output: A set of k clusters, C = $\{c_1, c_2, c_3,\ldots, c_k\}$

initialize empty sets D
initialize m clusters $c \in$ C, each contains a faceted unit, fu
compute distance set, D where $d_{ij} = d(c_i, c_j)$, $d_{ij} \in$ D among set C
find initial minimum distance, $d_{min} = argmax$ D
while ($d_{min} \le k$) // clustering starts
  select $d_{i,j}$ where $(i,j) = argmax_{i,j}$ D
  merge clusters c_i and c_j into a new cluster c_u
  remove c_i and c_j from C
  remove $d_{i,*} = d(c_i,*)$ and $d_{j,*} = d(c_j,*)$ from D
  update C with c_u
  foreach $c_v \ne c_u$
    compute $d_{uv} = f_{dist}(c_u, c_v)$
    update D with d_{uv}
  find $d_{min} = \min(d_{ij})$
output C // clusters generated
end

matches one of the ground truth tags after stemming. We have used standard performance metrics, namely precision (Equation (4)), recall (Equation (5)), and F-measure (Equation (6)) for evaluation. For this study, $F-1$ measure is used ($\beta = 1$).

$$\text{Precision}, P = \frac{\#\text{matched terms}}{\#\text{all extracted terms}} \tag{4}$$

$$\text{Recall}, R = \frac{\#\text{matched terms}}{\#\text{manually assigned terms}} \tag{5}$$

$$F\text{-Measure}, F - \beta = \frac{(1+\beta^2)PR}{\beta^2 P + R} \tag{6}$$

The FacetRank proposed in this study follows an unsupervised approach in general. Therefore, only unsupervised key term extraction algorithms will be compared. Among the unsupervised key term extraction algorithms, a few have been identified for comparison: TextRank [20], Document Profile (DP) Model [15], and KIP [30]. The details of experimental settings for each algorithm are as indicated in (Table 2).

A summary of evaluation results is given in (Table 3) with evaluation performed for top five key terms and top ten key terms for each algorithm. From the table, it has been discovered that in general, results using top five key terms are generally better than top ten ones in terms of F1 measure. The inclusion of extra key terms helped to boost recall at the expense of precision. Among the unsupervised approaches, KIP produces the poorest results of F-measure at 0.15. The performance results indicate that DP Model with averaged PMI selection comes second with F-measure at 0.28. FacetRank is better than DP Model at $F-1 = 0.35$. Comparatively, TextRank produces the best results with F-measure of 0.40. Compared with TF × IDF baseline method, it has been discovered that the performance of all unsupervised approaches are better except for KIP.

Our experimental results show that FacetRank is good at generating a better variety of salient key terms that consists of keywords and keyphrases. The drawback, however, is slower iterative ranking computation compared to TextRank that uses only single words. TextRank is able to generate better candidate words that consist of nouns and adjectives annotated using POS tagging. This explains the relatively good performance of TextRank over FacetRank. In comparison, FacetRank is also able to generate promising candidate terms using much simpler pre-processing steps and better suited to heterogeneous descriptions of review documents. While the use of morphological analysis, e.g. POS tagging, may improve the performance by identifying better candidate terms through selected POS tags on terms (e.g. nouns and adjective-noun term pairs), the disadvantage of such an approach is that POS taggers are only limited to a few languages. Our results showed that relative good results can be obtained by using simple selection heuristics at the expense of computational time.

4.2. *Contextual annotation of key term*

For evaluation of contextual annotation of key term, we are unable to find any annotated corpus that is specially designed for such an evaluation purpose. In order to judge the effectiveness and quality of annotations, an annotated and classified document corpus, Manufacturing

Table 2. Summary of experimental settings for algorithm in comparison.

Algorithm	TextRank	DP model	KIP	FacetRank
Experimental settings	Undirected Co-occurrence window = 3 Damping factor, $d = 0.85$ Iterative ranking threshold, $\delta = 0.001$	Support, $s = [2,10]$ Gap, $g = [0,27]$ 27 sets of DP Averaged PMI for evaluation	Default settings without pre-weighted keywords "1 word – 3 words" selected	Damping factor, $d = 0.85$ Iterative ranking threshold, $\delta = 0.001$

Table 3. Summary of evaluation results for CiteULike-180 data-set.

	Top 5 key terms			Top 10 key terms		
Algorithms	Pr	Rec	$F-1$	Pr	Rec	$F-1$
TextRank	0.31	0.54	0.40	0.20	0.67	0.31
FacetRank	0.29	0.46	0.35	0.16	0.52	0.25
DP model + averaged PMI	0.20	0.49	0.28	0.11	0.50	0.18
KIP	0.30	0.10	0.15	0.22	0.14	0.17
TF × IDF baseline [18]	0.14	0.16	0.15	N/A	N/A	N/A

Corpus Version 1 (MCV1) [13] is used. MCV1 is an archive of 1434 manufacturing-related engineering papers that have been gathered from the Society of Mechanical Engineers (SME). It combines all engineering technical papers from SME between year 1998 and year 2000, classified into 18 major categories that follow the field classification of SME. The goal of evaluation is to manually judge the quality of semantic annotations and how input terms can be associated with category labels or classified manufacturing concepts. For this purpose, a few input terms that exist in document text are selected randomly. Prior to the evaluation process, preprocessing tasks for MCV1 corpus are performed. ES are generated using top 15 key terms. Damping factor, $d = 0.85$ and iterative ranking threshold at $\delta = 0.001$ are applied for FacetRank and similar FUs are aggregated in cluster groups. The distance threshold value used in this study is $k = 1.0$. Implementation wise, all the essential information at the document level, such as file name, sentence id, FUs, cluster groups, etc. are indexed using Lucene, a Java-based full-text search application programming interface for smoother indexing and retrieval. Faceted model for each main category and sub-category labels of MCV1 and top four terms (two keywords and two keyphrases) from each document are built for later comparison with input query terms.

The results for four input terms for evaluation are selectively shown in (Table 4). Feasible contextual annotations are in a dictionary-like format for various input terms. From experimental results, it is noted that feasible faceted models can also be generated for less occurring key terms, such as the first two examples of "automated guided vehicle" and "computer aided manufacturing." Based on the available category labels, some interesting annotations (e.g. "materials handling" for "automated guided vehicle") are discovered. Besides category label associations, other contextually similar terms are also extracted. For instance, the terms "control strategy" and "process planning" actually do not co-occur with "automated guided vehicle," but are suggested because their contexts are similar. Another example is "tool failure" and "tool condition," which are feasible annotations to "acoustic emission". Technically, the aforementioned contextually similar terms are selected because their faceted models are similar to our query term. From our results, we find that in both cases, the annotations are meaningful suggestions to our example input query term.

5. Case study

In order to illustrate our approach, we have performed a case study using a small corpus of laptop computers. The corpus contains a collection of 47 web documents: with 26 documents related to the ThinkPad SL410 series and 21 documents related to the ThinkPad X200 series. There are about 8000 words totally, with 1700 unique words in about 500 sentences for the ThinkPad SL410 data-set. The ThinkPad X200 data-set consists of about 16,000 words, with about 2800 unique words in about 970 sentences, which is a bigger data-set. Following the methodology for key term extraction using FacetRank as explained in Section 3, a list of key terms are initially extracted using FacetRank. ES were generated using top 15 key terms using ES generation algorithm. FacetRank was applied to generate FUs from ES that have at least three entities. As a result, there were about 150 FUs gen-

TJCI 895966
27 February 2014
Initial
CE: AS QA: JV
Coll: QC:

Table 4. Evaluation results for contextual annotation of example input query.

Input query (hits)	automated guided vehicle (5)	acoustic emission (13)
Faceted indicator (Weight)	concept (4.5221), high level (2.97), system (3.4124)	wavelet (7.381), wear (5.2434), sensor (5.8785), common (4.7029)
Representative sentences	**concept (4.5221)** … a new *automated guided vehicle* (agv) dispatching algorithm based on a bidding *concept*… **system (3.4124)** … *automated guided vehicle system* (agvs) simulation *system* (agvsimnet)… … an *automated guided vehicle* (agv) is a mobile robot commonly used to carry loads in material handling *systems* (mhs)…	**wavelet (7.381)** … a flank wear estimation technique in turning through *wavelet* representation of *acoustic emission* (ae) signals… **sensor (5.8785)** … *sensor* fusion method using both an *acoustic emission* (ae) *sensor* and a built in force *sensor* is introduced… … two different types of *sensor*, the *acoustic emission* (ae) and the power *sensor* …
Contextual category labels	material handling, kanban, flexible manufacturing system, cad	electric discharge machine, process design, carbide
Contextually similar terms	control strategy, process planning, control macro, job shop, net model	detect cut, tool failure, tool condition, flank wear, tool wear

erated from the corpus. These FUs were clustered using cluster distance threshold of $k = 1.8$, half of the average distance value between all initial clusters. This is an optimal cluster setting value to balance between the number of single FU clusters and similar FU clusters. As a result of this setting, a total of 40 clusters are produced. Using the FUs generated, faceted modeling for a collection of important key terms (top ten key terms from each document) were generated. There are about 240 highly important key terms with this in regard. Faceted models for these key terms were generated for later comparison with query term. (Table 5) shows the contextual annotation generated for two example queries: "screen" and "business." From the table, it has been shown that FacetRank is able to generate semantically related annotations. For faceted models, faceted indicators such as "widescreen" and "12.1 inch" for query term "screen" are informative to users. For the query term "business," the faceted indicators generated such as "owner," "user," and "superb" are also descriptive. From the results, it is discovered that a few contextual terms are also annotated, such as "wide aspect" and "display" for query term "screen;" and "performance" and "travel" for query term "business." These annotations provide useful indicators of a query term's context according to the corpus in consideration.

Contextual tagging learned from product reviews has a number of potential applications for product design purposes. In this study, we have applied the annotations generated from Table 5 in design-related ontology development. In this aspect, contextual or faceted description of a term allows an ontologist to learn a key term's context from multiple domain specific corpus. (Figure 3) shows an example of laptop ontology. For laptop, the annotations are useful to deduce the associations between different input terms and ontological concepts (e.g. "battery"). Further, the product ontology can be annotated with product functions, customer experience, emotions, etc. allowing designers to better understand a product from different angles. The realization of these features are helpful towards reducing the time and resources needed during ontology development process where new concept associations can be better discovered and erroneous annotation can be avoided. Another useful application is for contextual information search, retrieval, and information presentation. Presenting information contextually (e.g. Table 5) allows designers to have a better overview of their product query term's context and how it is related to other contextually similar terms. For instance, in Table 5, the term "business" can be related to "owner" (user concept), "appeal" (affordance/"Kansei" words) and "travel" (usage). Such an annotation facilitates better understanding of products as perceived from the user's perspective. In relation, designers can better compare the context of a similar product feature under different user's angle (e.g. average user vs. professionals), or to compare two different products under the same user's perspective.

6. Conclusion and future work

In product design, the availability of vast online product reviews has presented a great resource for product designers to elicit useful design-related information. This paper has presented an approach towards contextual tags discovery from product review using semantic relatedness feature and FacetRank, an iterative ranking approach. The outcome of evaluation and case study shows that our approach is feasible in suggesting contextually similar tags towards a given term of interest. Nevertheless, there are a few limitations that are identified. Firstly, a query term needs to occur in the corpus for at least once. We noticed that there are situations where the faceted model of an input term only consists of very few faceted indicators. There are also situations where the sentence that contain the input term may not have FUs, or is associated with very few FUs. In relation, as

Table 5. Contextual annotation results of two input query.

Input query (hits)	screen (49)	business (59)
Faceted indicator (Weight)	screen (6.1174), wide (4.3804), widescreen (4.1174), 12.1 inch (4.0019), inch (3.9352), program (2.5324), notebook (1.073)	owner (5.9896), appeal (4.4047), user (4.2527), superb (4.1152), haven (3.9896), like (2.2083)
Representative sentences	**wide (4.3804)** … *wide screen* display features a sharp 1280 × 800 native resolution… **widescreen (4.1174)** … 12.1 inch *widescreen* not only lends extra on *screen* workspace, it also…	**owner (5.9896)** … lenovo for creating a laptop that the small *business owner* can afford… **appeal (5.6338)** … built to *appeal* to the small to medium *business* user…
Contextually similar terms	screen, adapt, wide aspect, display, size	business, performance, travel, notebook, quality, design, price

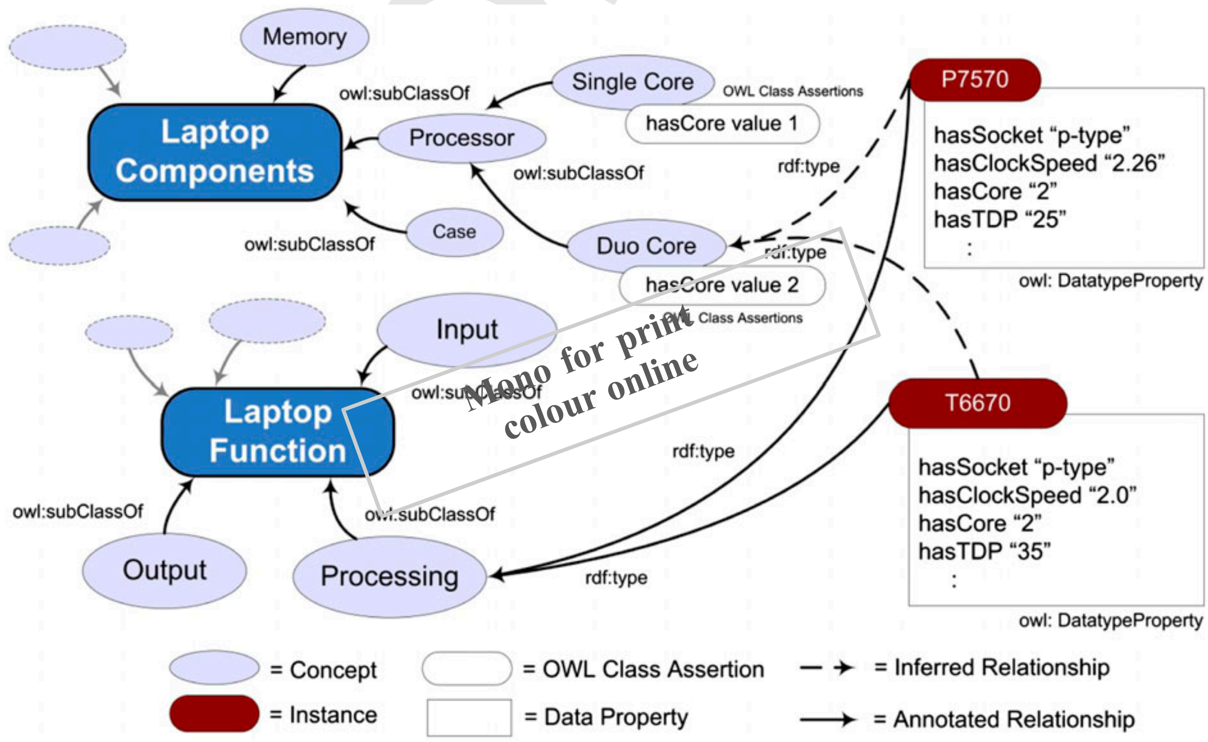

Figure 3. An example of laptop ontology.

FacetRank is based on a graph-based iterative ranking approach, there should be at least three terms in each FU for importance ranking. Our experimental results show that this is generally affected by the initial number of selected key terms for FU generation. Strategies such as adaptive number of key terms according to document size may improve the situation. Secondly, while the quality of annotation can certainly be improved by including greater features, such as information content, an annotated corpus that is specifically built for evaluating the quality of contextual annotation is, to the best of our knowledge, is lacking. In constructing such a corpus, the overall corpus design, selection of annotated input terms, inter-consistency of human annotators, etc. are all non-trivial issues. In spite of this, we believe that such a corpus is important for future studies. Application wise, we are also interested to see how our proposal can be realized to actually assist product designers in designing better products, or novice engineers in better design understanding. Thus, user studies are also recommended for validating the purpose in real-world scenarios.

Notes on contributors

Soon Chong Johnson Lim is a senior lecturer in the Department of Engineering Education, Faculty of Technical and Vocational Education, Universiti Tun Hussein Onn Malaysia (UTHM). He received his PhD from the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University in 2012. His research interests include design informatics, knowledge discovery and management in design and manufacturing, design methodology, and process and design education.

Shilong Wang received his BS, MS, and PhD degrees in Mechanical Engineering from Chongqing University, China, in 1988, 1991, and 1995, respectively. Dr Wang currently is a professor at the School of Mechanical Engineering, Chongqing University, Chongqing, China. He serves as a director of the Chinese Journal of Mechanical Engineering. Dr Wang's research interests include manufacturing automation, computer integrated manufacturing, and enterprises informatization.

Ying Liu is currently a senior lecturer in the Institute of Mechanical and Manufacturing Engineering at the School of Engineering in Cardiff University. He obtained his bachelor and master both in Mechanical Engineering from Chongqing University, China in 1998 and 2001, and then MSc and PhD from the Innovation in Manufacturing Systems and Technology (IMST) program under the Singapore MIT Alliance (SMA) at Nanyang Technological University and National University of Singapore in 2002 and 2006, respectively. His research interests focus primarily on design informatics, manufacturing informatics, intelligent manufacturing, design methodology and process, product design, and ICT in design and manufacturing.

References

[1] Ahonen-Myka, H., "Finding all maximal frequent sequences in text," Proceedings of the 16th International conference on Machine Learning ICML-99 Workshop on Machine Learning in Text Data Analysis, J. Stefan Institue, Ljubljana (1999).

[2] Aronson, A. R., O. Bodenreider, et al., "The NLM indexing initiative," Annual Fall Symposium of the American Medical Informatics Associations, AMIA '00, American Medical Informatics Association, Los Angeles, CA (2000).

[3] Barker, K. and N. Cornacchia, "Using noun phrase heads to extract document keyphrases," Advances in Artificial Intelligence, Proceedings 13th Canadian Conference on AI, Montreal, Canada (2000).

[4] Brin, S. and L. Page, "The anatomy of a large-scale hypertextual web search engine," 7th International Conference World WIde Web, WWW'08, Brisbane, Australia (1998).

[5] Church, K. W. and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational Linguistics*, 16(1), 22–29 (1990).

[6] Ding, X. and B. Liu, "The utility of linguistic rules in opinion mining," Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Amsterdam, The Netherlands (2007).

[7] Hu, M. and B. Liu, "Mining and summarizing customer reviews," Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, Seattle, WA, USA, 168–177 (2004).

[8] Jain, A. K., M. N. Murty, et al., "Data clustering: A review," *ACM Computing Surveys*, 31(3), 264–323 (1999).

[9] Jin, J. and Y. Liu, "How to interpret the helpfulness of online product reviews: bridging the needs between customers and designers," SMUC '10 Proceedings of the 2nd international workshop on search and mining user-generated contents ACM, Toronto, Canada, 87–94 (2010).

[10] Ledeneva, Y., A. Gelbukh, et al., "Terms derived from frequent sequences for extractive text summarization," in Gelbukh, A. (ed), Computational Linguistics and Intelligent Text Processing, 593–604 (2008).

[11] Lee, T. Y., "Needs-based analysis of online customer reviews," Proceedings of the ninth international conference on electronic commerce, ACM, Minneapolis, MN, USA, 311–318 (2007).

[12] Ling, X., Q. Mei, et al., "Mining multi-faceted overviews of arbitrary topics in a text collection," Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Las Vegas, NV, USA, 497–505 (2008).

[13] Liu, M., W. Shen, et al., "An weighted ontology-based semantic similarity algorithm for web service," *Expert Systems with Applications*, 36(10), 12480–12490 (2009).

[14] Liu, Y., "Beyond opinion mining: How can automatic online opinion analysis help in product design?," 6th International Conference on Web Information Systems and Technologies (WEBIST) Valencia, Spain (2010).

[15] Liu, Y., H. T. Loh, et al., "Deriving taxonomy from documents at sentence level," in Prado, H. A. d. and E. Ferneda (eds), Emerging Technologies of Text Mining: Techniques and Applications, Idea Group, 99–119 (2007). AQ2

[16] Loh, H. T., J. Sun, et al., "Opinion extraction from customer reviews," Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2009), Aug. 30-Sep. 2, ASME, San Diego, CA, USA (2009).

[17] Markó, K., U. Hahn, et al., "Interlingual indexing across different languages," International Conference Recherche d'Information Assistée par Ordinateur, RIAO'04, Avignon, France (2004).

[18] Medelyan, O., E. Frank, et al., "Human-competitive tagging using automatic keyphrase extraction," 2009 Conference on Empirical Methods in Natural Language Processing, Aug. 6-7, Singapore (2009).

[19] Mei, Q., D. Xin, et al., "Semantic annotation of frequent patterns," *ACM Transactions on Knowledge Discovery from Data*, 1(3), 11–30 (2007).

[20] Mihalcea, R. and P. Tarau, "TextRank–bringing order into texts," Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 4 (2004).

[21] Miller, G. A., "WordNet: A lexical database for English," *Communications of the ACM*, 38, 39–41 (1995).

[22] Murphy, M. L., Semantic relations and the Lexicon, Cambridge University Press, Cambridge (2003). AQ3

[23] Nguyen, T. D. and M.-Y. Kan, "Keyphrase extraction in scientific publications," International Conference on Asian Digital Libraries, ICADL'07, Hanoi, Vietnam (2007).

[24] Pasquier, N., Y. Bastide, et al., "Discovering frequent closed itemsets for association rules," 7th International Conference on Database Theory, 398–416 (1999).

[25] Porter, M. F., "An algorithm for suffix stripping," *Program: Electronic Library and Information Systems*, 14 (3), 130–137 (1980).

[26] Resnik, P., "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research*, 11(1), 95–130 (1999).

[27] Turney, P. D., "Learning algorithms for keyphrase extraction," *Information Retrieval*, 2(4), 303–336 (2000).

[28] Tversky, A., "Features of similarity," *Psychological Review*, 84(4), 327–352 (1977).

[29] Witten, I. H., G. W. Paynter, et al., "KEA: Practical automatic keyphrase extraction," Proceedings of the Fourth ACM Conference on Digital Libraries, DL'99, ACM, Berkeley, CA, USA (1999).

[30] Wu, Y.-f. B., Q. Li, et al., "Finding nuggets in documents: A machine learning approach," *Journal of the American Society for Information Science and Technology*, 57(6), 740–752 (2006).

[31] Ye, S., T.-S. Chua, et al., "Document concept lattice for text understanding and summarization," *Information Processing & Management*, 43(6), 1643–1662 (2007).

[32] Zhan, J., H. T. Loh, et al., "Gather customer concerns from online product reviews – A text summarization approach," *Expert Systems with Applications*, 36(2), 2107–2115 (2009).