e-Markets Group Research Report

Series: Technologies

Information Supply and Discovery Technology

2008
Cite the corresponding material from the report as:


Abstract. The Curious Negotiator project aims at the automation (to the extent possible) of the delivery and use of information by negotiation agents in electronic market environment. This chapter presents a framework of using text mining agents to provide processed online information to negotiation agents. It includes a news extraction algorithm, a quantitative process model based on the extracted news information, which is exemplified by an exchange rate prediction model, and a communication protocol between data mining agents and negotiation agents. This information is critical for the negotiation agents to form their negotiation strategies.

1. Introduction

Electronic negotiation is an area in AI that has witnessed significant development over the last decade, aiming at increased opportunities for entrepreneurs for global trade. Successful negotiation relies on an understanding of how to ‘play’ the negotiation mechanism [1] and how to utilise contextual information available at the time of negotiation [2]. Identifying, requesting and evaluating contextual information is part of the negotiation strategies as the negotiation proceeds. The significance of information to the negotiation process was formally analysed by Milgrom and Weber [3] in which the Linkage Principle, relating the revelation of contextual information to the price that a purchaser is prepared to pay, was introduced. “Good negotiators, therefore, undertake integrated processes of knowledge acquisition that combine sources of knowledge obtained at and away from the negotiation table. “They learn in order to plan and plan in order to learn” [4]. Substantial effort has gone in the development of a variety of negotiation strategies, including offers with argumentation [5]. However, very little work has been done in automating the utilisation of contextual information and the development of mechanisms that allow incorporating this information in a computationally feasible (and executable) negotiation strategy.

This chapter focuses on the process of using intelligent agents for gathering, processing and presenting information to support negotiation agents. This work is a part of the Curious Negotiator project, of which one of the main aims is to automate (to the extent possible) the delivery and use of information by negotiation agents. The curious negotiator which was proposed by the authors is an internet-based multiagent system of competitive agents supporting multiple attributes, illustrated in figure 1.
The curious negotiator is designed to incorporate data mining and existing information discovery methods [6] that operate under time constraints, including methods from the area of topic detection and event tracking research [7]. The idea is encapsulated in the intelligent data mining agents in the system, which operates in tandem with human and/or negotiation agents. Initially the information is extracted from various sources including on-line news media, virtual communities, and company and government on-line publications (including websites). Extracted information is converted into a structured representation. Both pre and post processed representations are stored in the mining database. They are used for further analysis by different data mining algorithms, including different text and network mining agents. The output of the analysis results is used by the negotiation agents.

Illustrated by the example of currency exchange trading, this chapter describes the framework of online information extraction and data modelling for negotiation agents. The rest of the chapter is organised as follows: Section 2 presents an algorithm of online news extraction. An exchange rate model between currencies using news articles and economic data is provided in section 3. Section 4 describes the communication protocol between data mining agents and negotiation agents based on ontology, followed by the conclusions in section 5.

2. **On line news extraction**

Internet contains vast and timely information that can be used by negotiation agents. However, obtaining and verifying information from on-line sources takes time and resources. To reduce the impact of some delay factors on the net, the architecture of the data mining system in figure 2 allows not only just-in-time operation, but also...
‘pre-fetching’ some of the information that is expected to be necessary for a scheduled negotiation. In the context of news mining, the news bots fetch the news, which then are transformed into a structured form and both the structured and unstructured data are stored in the mining base for accessing by the mining agents (The fragment selected illustrated in Figure 2 shows only the text mining agent).

There are a number of challenges for online data extraction in real world that the smart data mining system needs to address, including (i) critical pieces of information being held in different repositories; (ii) non-standard formats; (iii) changes in formats at the same repository; (iv) possible duplicative, inconsistent and erroneous data. This section addresses the first three issues, and the fourth issue is partly addressed. Although there are many types of information available from the internet, news is unarguable the most important information source to cause market dynamics. Therefore, a generic news bot is developed to monitor varied web sites of major news papers and obtain the latest news from news searching engines.

To retrieve related news URL is not a problem. Most of news web sites nowadays provide RSS channels that provide news feed in XML format. However, they normally only contain news titles and URLs; the actual news data is not available to avoid readers to skip their advertisements. The other way to obtain the URLs of related news is to use searching engines such as Yahoo or Google. However, to develop a computer program to retrieve an individual news article from the URL that is obtained either from search results or from the pre-fetched list automatically could be a tedious job since the news content can come from different web sites. Different news sources have different layout and format as illustrated by the two examples of news websites in Figure 3 and Figure 4. The layout may vary from time to time even in the news coming from the same source. Hence when automating news retrieval, even for the same news site, it is impractical to develop a static template, as it will stop working when the layout is changed. It is even more impractical (if not impossible) to develop a predefined program (template) for each news web site in the whole Internet.
Data extraction from Web documents is usually performed by software modules called wrappers. To overcome the problems caused by using hard-coded wrapper, significant research has been done in the area of wrapper induction, which typically applies machine learning technology to generate wrappers automatically [8, 9]. WIEN is the first wrapper induction system that defined six wrapper classes (templates) to express the structures of web sites [10, 11]. STALKER - a wrapper, more efficient than WIEN [12], treats a web page as a tree-like structure and handles information extraction hierarchically. Gao and Sterling [13] have also done significant work on knowledge-based information extraction from the internet. However, most of the earlier wrapper techniques were tailored to particular types of documents and none are specific for news content retrieval. The more recent techniques aim on data extraction from general semi-structured documents. The application of general content identification and retrieval methods to news data brings unnecessary overhead in processing. A technique that takes into account the characteristics of news web pages was proposed by the authors [14]. Without loss of generality, the approach improves the processing efficiency and requires neither user specified examples nor priori knowledge of the pages.

The data extraction process is divided into three stages. The logical structure of the tagged (in our case, HTML) file is firstly identified and the text, which is most likely to be the news article, is extracted. During the second stage a filter is dynamically built and some extra text is filtered out if multiple documents from the same web site are available. During the third stage extracted data is validated by the developed keyword based validation method. The details are presented below.
Stage 1: Identifying the logical structure of the tagged file

News pages normally not only contain the news article, but more often, also related news headings, the news category, advertisements, and sometimes a search box. Although each web site may have a different format, web pages can always be broken down into content blocks. The layout in which these content blocks are arranged varies considerably across sites. The news article is expected to be the content block which is displayed on the “centre” of the page. Therefore, it is reasonable to assume that the biggest block of text on the news web page is the news article. Similar to McKeown et al.’s [15] approach, the biggest block of text is detected by counting the number of words in each block.

Most of web sites employ visible and invisible tables in conjunction with Cascading Style Sheets (CSS) to arrange their logical structures by using HTML table tags [16]. Tables are designed to organize data into logical rows and columns. A table is enclosed within the <table></table> tag. Nested tables are normally used to form a complex layout structure. It is common for news web sites to display advertisements within news articles to attract reader’s attention. This is normally done by inserting nested tables that contain advertisements and other contents in the table that contains the news article. The pseudo code of the process is presented in Figure 5.

<table>
<thead>
<tr>
<th>Input: HTML file</th>
<th>Output: The largest body of text contained in a table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Begin</strong></td>
<td></td>
</tr>
<tr>
<td>1. Break down the HTML file into a one dimensional array, where each cell contains a line of text or an HTML tag</td>
<td></td>
</tr>
<tr>
<td>2. Remove the HTML tags except &lt;table&gt; and &lt;/table&gt;</td>
<td></td>
</tr>
<tr>
<td>3. Set table_counter to 0</td>
<td></td>
</tr>
<tr>
<td>4. For each cell in the array:</td>
<td></td>
</tr>
<tr>
<td>a. if &lt;table&gt; tag is encountered, increase table_counter by 1</td>
<td></td>
</tr>
<tr>
<td>b. if &lt;/table&gt; tag is encountered, decrease table_counter by 1</td>
<td></td>
</tr>
<tr>
<td>c. if it is a text element, append it to the end of container[table_counter]</td>
<td></td>
</tr>
<tr>
<td>5. Return container[i] that contains the largest body of text by counting the number of words.</td>
<td></td>
</tr>
<tr>
<td><strong>End</strong></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. Pseudo code of the algorithm for identifying the largest text block.

Stage 2: Building internal filters dynamically

Although most of news web sites use tables for partitioning content blocks, there are some web sites that use other methods. Also, even for the web pages that use tables as the partition method, the table with the news article may contain a few extra lines of text at the beginning or the end of the article. Therefore, extraction accuracy can be
improved by developing algorithms that do not rely on table tag information. Many web sites use templates to automatically generate pages and fill them with results of a database query, in particular, for news web sites. Hence, news under the same category from the same source is often with the same format. When two or more web pages from the same source become available, a filter can be constructed by comparing the extracted text from these pages. The filter contains the common header and tail of the text. The text is compared sentence by sentence from the beginning to the end between two files. Common sentences are regarded as part of web page template. Therefore, they should be removed from the file. The pseudo code of the process is shown in 6. Once the filter is generated, text is refined by removing the common header and tail text in the filter. Since the filter is dynamically generated, it is adjusted automatically when the web site format is changed.

**Input:** two text files from the same web site, each contains a news article  
**Output:** a data structure contains:  
- String **URL**  
- String **Header**  
- String **Tail**  
  1. Remove all the html tags in the files.  
  2. Break down the files into one dimensional arrays (a and b), each cell contains a line of text.  
  3. For each cell of the array from beginning  
     1. if \( a[i] = b[i] \), append \( a[i] \) at the end of **Header** string  
     2. if \( a[i] \neq b[i] \), break;  
  4. For each cell of the array from the end  
     1. if \( a[i] = b[i] \), insert \( a[i] \) at the beginning of **Tail** string  
     2. if \( a[i] \neq b[i] \), break  
  5. Set the **URL** value to the common part of the URLs of two text files  

Return the data structure that contains **URL**, **Header** and **Tail**.

**End**

*Fig. 6.* The pseudo code of dynamic filter generation.

Stage 3: Keyword based validation

Incorrect and out of date URLs can cause errors in the results of data extraction. Such errors can not be identified by the data extracting methods described in the previous sections. A simple validation method based on keyword frequency is developed to validate the data retrieved by the algorithms in figure 5 and 6.

The basic assumption is that a good news title should succinctly express the article’s content. Therefore, the words contained in the news title are expected to be normally among the most frequent words appearing in a news article. Consequently, the words from the news title (except the stop words, which are filtered out) are considered as
keywords. For situations when the news title is not available at the time of text extraction, the words in the first paragraph of the extracted data are considered as keywords, based on the assumption that title is always placed at the beginning of an article. The extracted text is regarded as the requested news article if it satisfies the following condition:

$$\min \left( w_1 \frac{l}{l_m}, w_2 \frac{n_k}{t_k}, w_3 k_f \right) > th$$

where:
- $l_t$ total length
- $l_m$ minimum length (predefined)
- $n_k$ the number of keyword that appears in the text at least once
- $t_k$ total number of keywords
- $k_f$ average keyword frequency
- $w_1, w_2, w_3$ weighting values
- $th$ threshold value (predefined)

The first term in equation 2.1 considers the total length of the extracted text. If the text length is unreasonably short, the text is unlikely to be a news article. The second term in the equation represents the percentage of the keywords that appeared in the text. The third term in the equation stands for the average frequency of the keywords that appeared in the text. The validation value takes the minimum value of these three and then compares with a predefined threshold to validate if the extracted text is the news article.

3. Exchange rate modelling using text mining techniques

To assist the negotiation agents to form the negotiation strategies, data mining agents need to process the extracted information further. The implementation of this step varies depending upon the actual application. In this chapter, a currency trading example (in particular, between the Euro and the US dollar) is used to illustrate the procedures of processing news data using text mining agents.

Exchange rates prediction is one of the most challenging applications of modern time series forecasting. Until recently, most of the models are empirical models based on macro economic data. Among the enormous amount of empirical models, the sticky price monetary model of Dornbusch and Frankel remains the workhorse of policy-oriented analysis of exchange rate fluctuations [17], which can be expressed as follows:

$$s_t = \beta_0 + \beta_1 m + \beta_2 y + \beta_3 i + \beta_4 \pi + \mu_t$$

where $s$ is the changes of interest rate during each sampling period; $m$ and $y$ denote the logarithm of the money value and real GDP respectively; $i$ and $\pi$ are the interest and inflation rate, respectively; $\hat{\beta}$ denotes the inter-country difference of the corresponding variable; $\mu$ is the error term. These models have performed reasonably
well in explaining exchange rate development in the long term, but little success in predicting exchange rate in short and middle term movement. The general consensus of the poor performance of the traditional empirical models using economic fundamentals to account for exchange rate developments on short to medium term is caused by the irrationality of the market participants, bubbles, and herd behaviour, which are hard to be captured in econometric models.

Recent literature shows that news about fundamentals has played an important role in creating market dynamics. Prast and De Vor [18] have studied the reaction of investors in foreign exchange markets to news information about the euro area and the United States on days of large changes in the euro-dollar exchange rates. Unlike the traditional models, daily changes in the euro/dollar rate on news about economic variables in the United States and the euro area, and the variables capturing news in the two economies were used in the regression model, which is:

$$E_t = \alpha + \sum_{i=1}^{8} \beta_i D_i + \epsilon$$  \hspace{1cm} (3)

where $E_t$ is the percentage daily change in the euro-dollar exchange rate; $D_{1-8}$ represent the following variables: 1 - real economy, euro area; 2 - inflation, euro area; 3 - change in official interest rate, ECB; 4 - statements/political events, euro area; 5 - real economy, United States; 6 - inflation, United States; 7 - change in official interest rate, United States; 8 - statements/political events, United States. It has been found that there is strong correlation between exchange rate daily movement and the market participants’ responses to the daily economy news and political events.

More recent research has confirmed that news has statistically significant effects on daily exchange rate movement. Ehrmann and Fratzscher [19] have evaluated the overall impact of macro news by analysing the daily exchange rate responses using similar regression models with news variables. Three key results were found. Firstly, the news about fundamentals can explain relatively well the direction, but only a much smaller extent to the magnitude of exchange rate development. Secondly, news about US economy has a larger impact on exchange rates than news about the euro area. Thirdly, higher degree of market uncertainty will lead to more significant effects of news releases on exchange rate movements.

The above findings motivated the research reported in this chapter. By using the text mining techniques, the manual process of identification and classification of positive and negative news can be automated. As the correlation between news and currency exchange rate has only been identified recently, there is not much work reported in this area. Eddelbüttel [20] and Wong [21] both tried to use the keywords in news headlines to forecast intraday currency exchange rate movements. Eddelbüttel used a set of keywords to identify the relevant news and sorted them into three groups: “All”, “DEM” and “USD”. Then the number of news pieces in three groups are calculated and used as the variable in the GARCH(1,1) model for prediction. The news analysis is restricted to the counting of the number of relevant news headlines to avoid qualitative judgement about “good” and “bad” news. Wong etc. proposed a prediction model based on the occurrence of keywords in news titles. The keywords in the news title are identified by selecting the words with the highest weighting values. A set of
rules for predicting the exchange rate movement direction from the keywords in the news titles are generated. These over simplified approaches only utilise news information to a very limited extent. A more sophisticated text mining approach for news filtering and classification was firstly proposed by the authors and is presented in this chapter. The design of this sub-system is shown in figure 7.

Fig. 7. Structure of the exchange rate model

3.1 Data Collection and Pre-processing

Before incorporating the news effect into an exchange rate model, it is important to identify the relevant news and classify them into “good” or “bad” news category, which would have opposite impact on the market behaviours. “good” news are the news that may cause an appreciation of one of the currencies, in this case, the US dollars; and the “bad” news have vice versa effect. This section describes the training process of news filtering and classification.

News articles used in the prediction model are retrieved from online news sources. Prior to the processing, the news articles used for training are manually classified into two groups: news affect exchange rate (target corpus) and other news (generic corpus). Choosing the news articles in the target corpus is crucial for the process since the target corpus contains the underlying knowledge of what factors affect exchange rate movement. Much research has studied the factors that affect currency exchange rate, which can be macroeconomic data, statements by central bankers and politicians and political events that affect macroeconomics. Therefore, only the news that is relevant to these is chosen. To improve the process efficiency and avoid noise distraction, stop words in the target corpora are replaced by a stop word symbol but are not removed completely to avoid incorrect word co-occurrence. Porter stem algorithm is also applied to remove the common morphological and inflexional endings from words in the documents.

3.2 Automatically Keyword Extraction

Text mining operations are mainly based on the frequency of keywords. The goal of this step is to generate the best set of keywords that can distinguish news documents related to exchange rate from other news documents. To reduce the calculation complexity and increase the processing efficiency, the number of keywords is kept to the minimum amount but is still a good approximation of the original document set in its full space. There are two types of keyword frequencies used in this paper: term frequency and document frequency. The term frequency is calculated by the number
of times a term appears in the corpora. The document frequency is the number of the documents that contain this term in the corpora.

Keywords are not restricted to single words, but can be phrases. Therefore, the first step is to identify phrases in the target corpus. The phrases are extracted based on the assumption that two constituent words form a collocation if they co-occur a lot [22]. Once the phrases have been extracted, the key terms are extracted amongst the single words except stop words and the phrases in the target corpus. The generic corpus is the background filter. The distribution of terms in the target corpus and the generic corpus are compared. The terms in the target corpus that stand out are considered as the features of the corpus, indicating that these terms are domain-specific terminology. Dunning [23] suggested the log likelihood ratio (LLR) Chi-square statistic test is effective in determination of domain-specific terms. Vogel [24] also reported that LLR had a greater ability to differentiate the importance of a term in a domain than other methods such as information gain (IG) or mutual information (MI).

The likelihood ratio for a hypothesis is the ratio of the maximum value of the likelihood function over the subspace represented by the hypothesis to the maximum value of the likelihood function over the entire parameter space. In this case, the null hypothesis $H_0$ is formulated to test the distribution of a term is the same in the generic corpus and target corpus. $H_a$ measures the actual distribution of the term in the whole data set. The log likelihood ratio for this test is:

$$-2 \log \left( \frac{H_0(p; k_1, n_1, k_2, n_2)}{H_a(p_1, p_2; k_1, n_1, k_2, n_2)} \right)$$  \hspace{1cm} (4)

The binomial distribution of the log likelihood statistic is given by:

$$-2 \log \lambda = 2 \log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) - 2 \log L(p, k_1, n_1) - 2 \log L(p, k_2, n_2)$$  \hspace{1cm} (5)

where $\log L(p, n, k) = k \log p + (n - k) \log (1 - p)$

$k_1$ and $k_2$ are the document frequency of a term in the target corpus and generic corpus respectively,

$n_1$ and $n_2$ are the size of the target corpus and generic corpus respectively,

$$p_1 = \frac{k_1}{n_1}, \quad p_2 = \frac{k_2}{n_2}, \quad \text{and} \quad p = \frac{k_1 + k_2}{n_1 + n_2}.$$

The method scores the terms based on the difference in the percentage of documents containing the term in the target and generic corpus. It does not distinguish whether the difference is caused by the term occurring more or less in the target corpus. As in this research that only the terms significant in the target corpus are concerned, a simple condition is added to the ranking equation so the terms are significant in the generic corpus are filtered out:

$$\frac{p_1}{p_2} \geq 1$$  \hspace{1cm} (6)
3.3 News Relevance Classification

The news relevance classification is divided into two steps: the first step is to identify the news that has potential to cause movement in exchange rates; the second step is to identify the news that is Euro and/or US dollar related.

The exchange rate related news can be separated from other news based on the key terms extracted from the previous section, which often well represent the characteristics of the data set. In this case, a modified k-Means classification algorithm, which is particular suitable for this case, is chosen as being computationally simple and efficient. The centroid of the target corpus and the maximum Euclidean distance in the training data are calculated. The maximum distance is used as the threshold to determine if the data belongs to a target cluster.

News related to exchange rate may not be discussing Euro and US dollar currencies, which is further identified by using the frequency of the words of currency and country names it contains.

3.4 Positive and Negative News Classification

It is important to further classify the relevant news into “positive news” and “negative news” categories, as news in different groups have entirely different effects on the market behaviour.

Recent studies show that the effect of the news is the combined effect of market expectation and the news itself. A piece of news could have positive or negative impact to the market depending on the market expectation. Therefore, unlike some studies that define good and bad news by its immediately effect to the market, in this research, the news is defined to be good or bad according to its fundamental effect to the market. The market expectation is incorporated into the model in a later stage. For example, a news about US increased its interest rate is defined to be positive news to US dollars. The task of identifying “good” and “bad” news of exchange rate is not straight forward since both groups of news use similar set of keywords. For example, the following two pieces of news have exactly same set of words, but one is considered to be positive and the other is considered to be negative to the appreciation of US dollars:

1. The interest rate has gone up. The US dollar has gone down.
2. The interest rate has gone down. The US dollar has gone up.

The positive and negative news can use similar set of key terms, which causes great difficulties in the classification. However, the sequences of the key terms can represent the meaning of sentences better, which is well illustrated in the previous example. Therefore, a term is defined as the sequence of key terms in a sentence, which is used as the input features for the positive and negative news classification. The feature vectors of the above example can be represented as:
Table 1. Example of feature vector representation for “good”/“bad” news classification

<table>
<thead>
<tr>
<th>features</th>
<th>document 1</th>
<th>document 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>interest rate up</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>interest rate down</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>US dollar down</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>US dollar up</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

However, using key term sequence as classification features leads to a high dimensional vector space with sparsely distributed elements, which causes difficulty in separating instances into classes (subspaces). Therefore, the discriminant analysis is implemented to combine features of the original data in a way that most effectively discriminates between classes [11]. The discriminant analysis is to project the documents onto a lower dimensional subspace of the original vector space. After the projection, instances in the same class are tightly grouped, but well separated from the other clusters. Also, with a smaller set of input features, the complexity of the classification is reduced and the calculation efficiency can be greatly improved.

Again, the document collection with \( n \) documents and \( m \) features in cluster \( i \) represented by a term (key term sequence) frequency document matrix. In this application, there are two clusters - “good” news and “bad” news.

\[
A_i = [a_1, a_2, \ldots, a_m] \in \mathbb{R}^{n_{ni}}
\] (7)

An optimal linear transformation \( GT \) can be found such that the Euclidean distance between the clusters is maximised while the distance between instances within each cluster is minimised:

\[
G^T \in \mathbb{R}^{m \times n} : a_j \in \mathbb{R}^{m \times n_{1j}} \rightarrow y_j \in \mathbb{R}^{n_{ij}}, 1 \leq j \leq n_1 + n_2
\] (8)

To measure the cluster quality, scatter matrices are formulated based on the distance of each instance to the centroid. The scatter matrixes within cluster and between clusters are defined as the following equations:

\[
S_w = \sum_{i=1}^{2} \sum_{j \in n_i} (a_j - c)(a_j - c)^T
\] (9)

\[
S_b = \sum_{i=1}^{2} \sum_{j \in n_i} (c - c')(c - c')^T = \sum_{i=1}^{2} n(c - c')(c - c')^T
\] (10)

The scatter matrices of the transformed feature vectors are as follows:

\[
S_w^T = G^T S_w G, S_b^T = G^T S_b G
\] (11)

The closeness of the instances with the cluster and the separation between clusters can be calculated from the scatter matrices as \( \text{trace}(S_w^T) \) and \( \text{trace}(S_b^T) \) respectively.

The transformation matrix \( G^T \) is calculated by maximising the value of \( \text{trace}(S_b^T(S_w^T)^{-1}) \) that approximates the maximisation of \( \text{trace}(S_w^T) \) and minimisation of \( \text{trace}(S_b^T) \). The numerical algorithm for this optimisation problem presented in [25] is adapted.
After the $G^r$ being calculated, the k-Means classification algorithm can be applied to classify the transformed feature vectors $y_i$ into “good” and “bad” news categories.

### 3.5 The Econometric Model of Exchange Rate Responding to News and Economic Data

This research focuses on using text mining methods to incorporate the information in the news articles into a currency prediction model. As euro/dollar exchange rate will be used as the testing case, the empirical model presented by Galati and Ho to study the news effect on economic data particular for euro/dollar exchange rate is chosen [26]. In this work, the above model is modified to incorporate a news index ($I_{news}$), which reflects the news effect on exchange rate. The regression equation has the following form:

$$
\Delta \ln S(t) = \alpha_0 + \alpha_i x_i(t) + \beta I_{news} + \epsilon
$$

(12)

where $x_i$ represent the economic data variables which include: US non-farm payrolls, US unemployment rate, US employment cost index, US durable goods orders, NAPM manufacturing, NAPM non-manufacturing, US advance retail sales, US industrial production, US CPI, Ifo index, Germany unemployment rate, Germany industrial production, INSEE industrial trends, Germany CPI and EU 11 PPI.

### 3.6 Information summation

Ontology offers a way to share information between agents. A simple generic ontology using XML is built to integrate and present the data mining results of intelligent agents to the negotiation agents, which includes service ID, time stamp, estimate price, market price, and price explanation. In the future, a domain specific ontology will be built for each data mining agent to represent the key factors and their relationships with the estimate price. In this case, it is the relationships between economic data, the key factors discovered from the most relevant news and the estimate exchange rate. The ontology should be evolved when the factors that affect the exchange rate have been changed. With the domain specific ontology, the text mining agents are capable of answering a wide range of questions which can help the negotiation agents to generate the negotiation argumentation in a much flexible way.

### 4. Agent communication protocol

The curious negotiator as shown in figure 1 is a multi-agent system. A simple agent communication protocol is developed for. It is based on the ontology negotiation protocol between information agents for scientific archives, which well suites this particular application [27]. This protocol can be implemented using KQML format.

Figure 8 shows the high level structure of message passing scheme between negotiation agents and text mining agents.
Text mining agents and negotiation agents send and receive information through XML encoded messages. The ontology developed in section 3.6 provides a set of concepts, or meta-data, which can be queried and used to control the behaviour of agent cooperation. These concepts are marked using XML tags to underlie message interpretation. The structures and the semantics of the documents are represented by the corresponding DTDs and interpreters.

When a new text mining agent is added to the curious negotiator system, it advertises the service that it can provide. There are four scenarios at the current design. A negotiation agent initiates a service request to a text mining agent. The text mining agent can decline the request or fulfil the service. In the case that the negotiation agent receives the results of the requested service, it can request for additional information contained in the ontology. Query can only be handled after the service has been processed. The text mining agents keep the data of current service. The text mining agents record the query history. Therefore the information can only be query once.

With the domain specific ontology developed, the text mining agent can answer some sophisticated questions such as:

1. What is the most important factor affecting the current market?
2. How does the market change if a particular event happens i.e. interest rate goes up 0.5%?

These provide a broad knowledge base to negotiation agents and enable them to generate offers backed by arguments which summarise the reasons why the offers should be accepted. This type offer is more persuasive and thus places the negotiation agents in a better position to succeed.
5. Conclusions

This chapter presents a framework for providing related information to the negotiation agents in the electronic market environment. It includes a news extraction algorithm, a quantitative process model based on the extracted news information, which is exemplified by an exchange rate prediction model, and a communication protocol between data mining agents and negotiation agents. The news extraction algorithm utilises data mining techniques for automatically collection of relevant set of news articles from varied news sources on the web, regardless of the format and structure of the sources. A novel approach to building an exchange rate prediction model using the extracted news articles and economical data are also presented. Recent research suggests that the market daily movement is the result of the market reaction to daily news. However, this type of news is not included in most of existing models due to its non-quantitative nature. This chapter is the first attempt to apply text mining methods to incorporate the daily economic news as well as economic and political events into prediction models with economic data. This approach leads to a more accurate exchange rate model with better prediction rate.

The curious negotiator is still in its infancy. The goal is to blend ‘strategic negotiation sense’ with ‘strategic information sense’ as the negotiation unfolds. This requires a system capable of providing effective information to the negotiation agents. The information is critical for the negotiation agents to form their negotiation strategies. The smart data mining system presented in this chapter supports the negotiation agents that operate under time-constraints and over dynamically changing corpus of information. Future developments include incorporating the trust information into the system.

References


Informed Recommender: A Recommender System That Bases Recommendations on Consumer Product Reviews

Silvana Aciar¹, Debbie Zhang², Simeon Simoff² and John Debenham²

Abstract—Consumer reviews, opinions and shared experiences in the use of a product is a powerful source of information about consumer preferences that can be used in recommender systems. Despite the importance and value of such information, there is no comprehensive mechanism that formalizes the opinions selection and retrieval process and the utilization of retrieved opinions due to the difficulty of extracting information from text data.

In this paper, a new recommender system that is built on consumer product reviews is proposed. The review comments could come from chat rooms or online discussion forums. A prioritizing mechanism for producing the recommendation based on consumer level of expertise in using a product was developed. To make the recommendation process be able to utilize the textual information, an ontology was defined so the review comments can be represented in structured formats. Each piece of review comment should be mapped into the ontology as an instance so an automatic mapping process using text mining technique is proposed.

Therefore, a recommender system built on this information can overcome the problem of lacking initial information. The problem becomes even more challenging when the recommender system comes to deal with new products or the products have not been evaluated by consumers. Our work also addresses this problem based on comparison of product specification.

The proposed approach is demonstrated using a case study using digital camera reviews.

Index Terms—Ontology, Reviews Acquisition, Recommender Systems, Cold Start problem, Text Mining.

1. INTRODUCTION

Recommender systems are programs which attempt to predict items that a user may be interested in, given some information about the user's and items’ profiles. Most existing recommender systems use content-based methods, collaborative filtering methods, or hybrid recommender methods that combine both techniques. Content-based methods use information about the item itself to make suggestions, rather than information about the preferences of other consumers. Such recommender systems emulate the behaviour of a consumer recommending a product to her friend, because she has used the product and know the preferences of her friend in terms of product features (content-based). Such recommender systems uniquely characterise each consumer without having to match her/his interests to other consumers. They can provide a list of content features that explain why an item has been recommended. Such a list can strengthen consumer confidence in the recommendation and provide reflection of consumer own preferences. In the content-based approach consumers can provide some initial information about the product to assist the system.

Collaborative filtering makes recommendations about the preferences of a user (filtering) based on collected taste information of other users’ (collaborative). The underlying assumption of collaborative filtering methods is that those who agreed in the past also tend to agree again in the future [D. Billsus and M. Pazzani (1998). Learning collaborative information filters. In Proceedings of the 15-th International Conference on Machine Learning, 24-27 July, 1998, Madison, Wisconsin, Morgan Kaufmann, San Francisco, CA, pp 46-54]. In other words, such recommender system emulates the behaviour of a consumer recommending a product to her friend, because some other consumers that she knows and believes that have similar tastes to her friend, like the product. Technically such recommender system operates similar to a case-based reasoning system, without the adaptation step. It maintains a case base of the preferences of individual consumers, for a given patron finds other consumers whose known preferences correlate significantly with the patron, and recommends to a person other items enjoyed by her/his matched patrons. The system can provide the list of some of these patrons relating them to their other purchases in order to provide consumer with some explanation and confidence in the recommendation. The approach requires sufficient number of consumer ratings. These systems use a collection of historical rating data of m users on n products as input, which are collected by asking users to input the rating of the products on some rating scales [1]. The collection of such ratings requires consumer’s response (e.g. spending some time) and the actual values may not necessarily provide reliable estimates of consumer preferences. Another important issue is the

¹ University of Girona, Campus Montilivi, Building P4, 17071 Girona, Spain. email: saciar@eia.udg.es
² Faculty of IT, University of Technology, Sydney, POBox 123 Broadway 2007, Australia email: {debbie, simeon, debenham}@it.uts.edu.au.

recommendation of products that have not been rated by a sufficient number of consumers. Traditional collaborative filtering approach does not provide effective recommendation strategies.

Rather than completing forms with rating values, many consumers prefer to use natural language and express their opinions about the product in a free text form, similar to a conversation with a friend. In the online world, there are several popular ways for consumers to exchange their experiences with a product - product review forums, virtual community logs, product discussion boards and e-commerce sites. There is growing evidence that such forums informal and influence consumers’ purchase decisions [5, 6]. Despite the importance and value of such information, there is no comprehensive mechanism that formalizes:

- The process of selection and retrieval of opinions, and
- The utilization of retrieved opinions.

Part of the problem resides in the complexity of extracting information from text data and converting it into product recommendations. Adomavicius provided an overview of recent developments in recommender systems [7]. According to this review, the recommender systems that utilize review comments using text mining techniques are yet to be developed. Ricci [8, 9] proposed to utilize review comments for product description and user behaviour study. He believed the review comments could be widely used in recommender systems and result in better recommendations. Ricci and Wietsma [8, 9] so far seem to be the only recommender system that integrates reviews in the recommendation process. The authors use product reviews in the product selection decision process for a mobile recommender system. They employ social-filtering algorithms [10] to extract knowledge from the reviews. The main aim of their system is to improve the explanation of the recommendation providing the relevant reviews of users with similar tastes. The reviews are used to give explanations of the recommendations, but they are not used to make recommendations.

This paper addresses the problem of utilisation of consumer opinion about products, expressed online in a free text form in order to generate product recommendations. Figure 1 shows the overall process structure of the proposed recommender system.

The realisation of this process structure requires the completion of several tasks, including:

- The development of an information representation structure using the quality/feature ontology.
- The implementation of a text mining algorithm for mapping automatically the information from the reviews into the in-formation structure of the ontology.
- The development of a ranking mechanism that computes the rating of a product using the information from the consumer reviews stored in the ontology.
- The development of a recommender mechanism, which computes recommendations in response to a user request.

The process collects relevant product consumer reviews and builds a collection of relevant reviews. Technically, the procedure for collection of product reviews follows the algorithms for auto-mated news extraction from news sites developed in [15]. Once the product opinions mining base is populated, we employ text mining techniques to extract useful information from review comments. In order to make reviews information useful for the recommendation process, it has to be translated into a structured form and communicated to the recommender process in a form suitable for generating recommendations. We have developed and employed an ontology to translate opinions’ quality and content into a form suitable for utilisation by the recommender process. The ontology contains two main parts: Opinion Quality and Product Quality, which summarise the consumer skill level and the consumer experience with the product in the review, respectively. The text mining process maps the review comments into the ontology. A ranking mechanism operates with over the data stored in the ontology. It prioritises that information with respect to the consumer level of expertise in using the product in consideration. The recommendation is made based on the data in the ontology. Therefore, the recommendation quality depends on the accurate mapping of the proper knowledge from the semantic features in the review comments into the ontology structure.

The major contribution of the paper is the overall framework for automating the utilisation of consumer reviews, and its individual components. Where possible it utilises existing algorithms (for example, in the text mining process), as the goal of the reported work is to demonstrate the strengths of the overall approach. The structure of the paper is as follows. In Section 2 we describe the translation ontology that provides the “filing cabinet” for the structured representation of reviews. Section 3 presents the details of the mapping process, which uses text mining techniques. The set of measures used to calculate a rating of a product based on the reviews is briefly explained in Section 4. Section 5 presents our approach towards the problem of cold-start, using comparison of product specifications. Section 6 presents a case study that demonstrates the effectiveness of proposed approach. The case study is in the popular product domain of digital cameras. Finally, the conclusions, the limitations of the work and the directions for future research are discussed in
2. Representation of Consumer Reviews

A typical review comment could be like:

*Canon PowerShot A530* is a very good camera with very good image quality. 5 MP is enough for very sharp pictures in almost every condition. Only negative is plastic body but considering the price it is by far one of the most valuable cameras to buy. 170€ for 4x zoom, 5MP and very good handling! Many useful features for great pictures... After trying different higher price cameras I was impressed by the speed of the AF and the typical Canon menu and functions. This is not the "smallest ever 3x zoom 8 MP Camera" but very good thing to work with. ISO 400 and 800 does not look really good.

The reviews chosen for this example is from the Digital Photography Review web page (www.dpreview.com). The goal of this step is to find a suitable tool for extracting the information contained in the text and converting it into structured data, such as a form depicted in Figure 2.

Section 7.1 presents the ontology developed for digital cameras reviews that are used in the study cases of this paper.

3. Mapping Review Comments into Ontology Instances

Ontology provides a controlled vocabulary and relationship to describe the consumer skill level and the consumer experience with the product in the review comment in the system. The classes and relationships in the ontology have been defined in our previous work manually. They are only required to be defined once and can be used until the products have new features. Each review comment is represented as an ontology instance. For an online service agent, manually mapping the ontology instances is a tedious and time consuming job. This section describes a methodology developed for the agent to create ontology instances automatically using text mining techniques. As the ontology has been defined, the mapping process includes the identification of both the classes involved in the instance and their attributes. The mapping process is composed by two steps:

1. Sentence selection and classification: This step identifies the class attributes. In the user valuation from the text data, each feature from the comment is assigned either “Good” or “Bad”. Therefore, the sentences in the review are selected and classified into three categories: “Good” comments, “Bad” comments and “Quality”. “Quality” category contains the sentences that indicate the opinion quality.

2. Concept identification: Once the relevant sentences are selected, this step identifies the classes that the selected sentences belong to. The concepts which implicated in the sentences determine the classes in the ontology, are identified by related words used as the product features, which is highly domain specific.
The next sections detail both steps of the mapping process.

### 3.1. Sentence Selection and Classification

Under the text mining paradigm, each sentence is treated as a document in this application. To group review sentences into “Good”, “Bad” and “Quality”, shallow parser was firstly considered as an analyzer tool. However, most of parsers give complicate and incorrect results. Furthermore, each document is very short. Classification algorithms based on term frequencies do not provide satisfaction results either. Therefore, rule based classification techniques are employed. As described in previous section, three categories have been defined to classify the sentences: “Quality”, “Good” and “Bad”. “Quality” category groups those sentences that contain information about the skill of the consumer. “Good” category groups those sentences that contain information about some features that consumer has valuated as the strengths of the product. “Bad” category groups those sentences that contain information about some features that the consumer considers as weaknesses of the product.

At this stage, the work has been focused on providing the over-all concept of utilizing text mining for automatic mapping of re-view comments into ontology instances. Hence, we employed an off-the-shelf text mining kits. The Text-Miner Software Kit (TMSK) and the Rule Induction Kit for Text (RIKTEXT) have been used to obtain the classification rule sets [13]. TMSK generates a dictionary from a set of documents (sentences in our case) and converts a set of sentences into sparse vectors based on the dictionary. The dictionary and the vectors representing each category are used by RIKTEXT for learning a classifier. RIKTEXT is a complete software package for learning decision rules from document collections. The rules are induced automatically from data. The output is a rule set of classification of “Good” “Bad” and “Quality” category from training data. Figure 4 shows the inputs and the output of the selection and classification process. Opinions of 68 reviews about the digital camera: Canon PowerShot SD500 (Digital IXUS 700) from www.dpreview.com web have been used to create the training data set. Each sentence of each review is treated as a document. 195 sentences have been obtained for the “Good” category, 127 sentences for “Bad” category and 47 sentences for “Quality” category. The available data have been spited into training and tests portions. Test cases are selected randomly in RIKTEXT and we specified how many cases should be used for testing. We choose two-thirds of the available cases for training and the rest for testing. The results are presented in Table 1. As can be seen, it displays a number of rule sets.

<table>
<thead>
<tr>
<th>Table 1 RULE SET TO CLASSIFY SENTENCES INTO GOOD CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RSet</strong></td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2**</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6**</td>
</tr>
</tbody>
</table>

Additional Statistics (Training Cases):
- precision: 71.6049
- recall: 89.2308
- f-measure: 79.4521

Additional Statistics (Test Cases):
- precision: 67.5676
- recall: 76.9231
- f-measure: 71.9424

Each rule set is numbered under the column “RSet”. A “**” delineates the rule set with the minimum error rate. A “*” indicates the best rule set according to the error rate and
simplicity. “Rules” is the number of rules in the rule set. “Vars” indicates the total number of conjuncts in the left-hand-side of the rules. The column “Train Err” gives the error-rate of the rule sets on the training data. “Test Err” is an error-rate estimate and Test SD is the standard deviation of the estimate. “Mean Var” is the average number of variables of the resampled rule set that approximates in size the rule set for the full data. “Err/Var” gives an indication of the quality of the solution.

The chosen rules are those that have minimum error rate or are very close to the minimum but may be simpler than the minimum (**). Precision, recall and f-measure obtained from training and test cases are shown at the end of the table. Tables 2 and 3 show the rule sets obtained from classification of “Bad” and “Quality” categories, respectively.

### TABLE 2
RULE SET TO CLASSIFY SENTENCES INTO BAD CATEGORY

<table>
<thead>
<tr>
<th>RSet</th>
<th>Rules</th>
<th>Vars</th>
<th>Mean Var</th>
<th>Err/Var</th>
<th>Err</th>
<th>Test Err</th>
<th>Test SD</th>
<th>Mean Var</th>
<th>Err/Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>28</td>
<td>0.1731</td>
<td>0.3710</td>
<td>0.00</td>
<td>0.00</td>
<td>3710</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>27</td>
<td>0.1755</td>
<td>0.3710</td>
<td>0.00</td>
<td>0.00</td>
<td>3710</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td>17.79</td>
<td>0.3466</td>
<td>0.00</td>
<td>1.50</td>
<td>0.00</td>
<td>0.3466</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>20</td>
<td>0.2043</td>
<td>0.3605</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3605</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>14</td>
<td>0.2150</td>
<td>0.2742</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2742</td>
<td>0.00</td>
<td>0.38</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>12</td>
<td>0.2740</td>
<td>0.2661</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2661</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>10</td>
<td>0.3029</td>
<td>0.2661</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2661</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>8**</td>
<td>9</td>
<td>9</td>
<td>0.2221</td>
<td>0.2881</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2881</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>6</td>
<td>0.2436</td>
<td>0.2742</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2742</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>7</td>
<td>0.3678</td>
<td>0.2923</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2923</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0.0000</td>
<td>0.6332</td>
<td>0.00</td>
<td>0.00</td>
<td>0.6332</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Additional Statistics (Training Cases): precision: 70.3704 recall: 44.1860 f-measure: 54.2857

<table>
<thead>
<tr>
<th>RSet</th>
<th>Rules</th>
<th>Vars</th>
<th>Mean Var</th>
<th>Err/Var</th>
<th>Err</th>
<th>Test Err</th>
<th>Test SD</th>
<th>Mean Var</th>
<th>Err/Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>14</td>
<td>0.1779</td>
<td>0.3466</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3466</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>20</td>
<td>0.2043</td>
<td>0.3605</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3605</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>24</td>
<td>25</td>
<td>0.1755</td>
<td>0.3710</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3710</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td>27</td>
<td>0.1755</td>
<td>0.3710</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3710</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>28</td>
<td>28</td>
<td>0.1848</td>
<td>0.3242</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3242</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>34</td>
<td>34</td>
<td>0.1901</td>
<td>0.3303</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3303</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>40</td>
<td>0.2063</td>
<td>0.3466</td>
<td>0.00</td>
<td>0.00</td>
<td>0.3466</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>8**</td>
<td>9</td>
<td>9</td>
<td>0.2221</td>
<td>0.2881</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2881</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>6</td>
<td>0.2436</td>
<td>0.2742</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2742</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>7</td>
<td>0.3678</td>
<td>0.2923</td>
<td>0.00</td>
<td>0.00</td>
<td>0.2923</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0.0000</td>
<td>0.6332</td>
<td>0.00</td>
<td>0.00</td>
<td>0.6332</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Additional Statistics (Test Cases): precision: 70.3704 recall: 44.1860 f-measure: 54.2857

### 3.2. Concept Identification

Once the sentences have been classified into one of the categories, the concept (class) in the ontology implicated in the sentence is needed to be identified. Each concept in the ontology contains a label name and a related word list. A related word list of a concept contains vocabulary (a set of keywords) through which the concept can be matched with one sentence in the comments. The related words generated are listed in Table 4. For example, related word for the concept “Comparison” found in reviews can be “compare, compared, equal, same, etc”.

### 4. RECOMMENDATION USING CONSUMER’S REVIEWS

The review comments are firstly mapped into ontology to make the ranking calculations possible. Since it has been explained in the Section 2, the ontology contains two main parts: Opinion Quality, which summarise the consumer skill level and the consumer experience with the product in the review, respectively. A set of measures: Opinion Quality (OQ), Feature Quality (FQ), Overall Feature Quality (OFQ) and Overall Assessment (OA) are computed based on the data in the ontology. Opinion Quality (OQ) is defined to evaluate the
weighting value of opinions according to the opinion provider’s expertise. Overall Feature Quality (OFQ) is the global valuation of the feature from all reviews, which is calculated from the Feature Quality (FQ) value of individual comment. Overall Assessment (OA) provides a final score of the product based on the valuation of each feature. The recommendation in response to a user request is given based on these measurements. The recommendation is made based on the review comments that are summarised by an Overall Feature Quality (OFQ) value for each feature. In the next sections are detailed the calculation of these measures.

4.1. Rating the Consumer Skill Level

The review comments were given by people with diverse experience and skill levels. In general, people who have longer history of using the product can provide more professional opinions. Therefore, these diverse opinions should not be treated equally. The opinions from more experienced people should be taken into account to a greater extent than those from people with little knowledge of the product. Opinion Quality (OQ) is defined to evaluate the weighting value of opinions according to the opinion providers' expertise.

**Definition 1.** Opinion Quality (OQ) is the sum of the weight \( w_j \) given for each variable representing the skills and experiences of consumer \( i \) divided by the number of variables representing the information about consumer’s skill and expertise provided in the ontology.

\[
OQ_i = \frac{\sum w_j}{n} \quad (1)
\]

The Opinion Quality is calculated by the values stored in the corresponding part of the ontology. An Opinion Quality value is calculated for each piece of comment.

4.2. Product Quality Ranking

The product is ranked according to the consumer comments for each feature. Due to the difficulties of quantification of user valuation from texture data, each feature from the comment can only be assigned either “Good” or “Bad”, which is calculated as “1” or “-1” respectively.

For each feature, a Feature Quality is calculated, which is a function of consumer valuation and Opinion Quality.

**Definition 2.** Feature Quality (FQ): The quality value for each feature of the product in a review is the rating multiply by the Opinion Quality value of the consumer

\[
FQ_j = r * OQ_i \quad (2)
\]

4.3. Selection of the Relevant Opinion and Making Recommendations in Response to a User Request

When a user requests the evaluation of a particular product based on certain features, the Overall Feature Quality is calculated from the reviews that contain the valuation of this feature.

**Definition 3.** Overall Feature Quality (OFQ) is the global valuation of the feature from all reviews, which is calculated by the average value of Feature Quality.

\[
OFQ_j = \frac{\sum (Scalingfactor * FQ)}{NumberOfOpinions} \quad (3)
\]

Here Scaling Factor is used to do the minor adjustment of the user valuation, which can be set to:

\[
Scalingfactor = \frac{1}{n} \quad (4)
\]

\( n \) is the number of all the features rated by the consumer. Each review rated different number of features so \( n \) could be different.

To provide the user with a comprehensive valuation of the product quality in related to the requested features, an Overall Assessment score is defined.

**Definition 4.** Overall Assessment (OA) provides a final score of the product based on the valuation of each feature. It is calculated as the sum of all OFQ (calculated by equation (3)) multiplied by the Importance Index.

\[
OA = \sum OFQ * Importance Index \quad (5)
\]

The Importance Index measures the different influence of the features to consumer’s decision making, which can be assigned in two ways: according to the importance of the feature expressed in the user request or by the frequency that the features have been rated in the consumer reviews.

5. HANDLING MISSING INFORMATION IN THE SYSTEM

Recommender systems suffer seriously from the cold start problem [14]. At the early stage of use of these systems, there is no or little initial consumer information available to make recommendations.

The cold start problem can be classified into two types of problems: the new-system problem and the new-item/user problem. The new-system problem is where there is no or little initial information for the system to perform recommendation. As the recommendation system discussed in this paper is based on user review comments from the internet, this problem can
be largely eliminated thanks to large amount of review comments available from the internet nowadays. However, as the review comments are provided by consumer voluntary, not every product has been evaluated. The problem of recommending items without evaluation data in this case is similar to the new-item/user cold start problem, where the system has been running for a while and a new item/user arrives, the system has no information for making recommendation. Most recommender systems perform poorly in this situation [15].

This paper addresses the new-item cold start problem based on the assumption that there are linear relationship between product specifications and the degrees of consumer satisfaction, which are described by Overall Feature Quality (OFQ) value of each feature in this system. The missing Overall Feature Quality data are estimated using linear extrapolation and interpolation of the product specification. The specification of features can only be in one of three forms: numerical value, Boolean value or a set of properties. For each different form, the distance of a feature between two products is calculated as:

- The Euclidean distance for numerical specifications.
- Cosine distance, if the specification contains a set of properties.
- Either be 0 or infinity for Boolean values, depending on whether the specifications are the same or different.

Due to the fact that the quality can vary from different manufacturers even though the specification is the same, the estimation should be performed using the data from the same manufacturer. In the case when the data of the same manufacturer are not available, data from other manufacturers can be applied.

6. CASE STUDY

In this section, we present the different steps that must be considered in order to offer a recommendation about digital camera in response to a user request. Case study was conducted using digital cameras. Data from the Digital Photography Review (www.dpreview.com) were chosen. Each day consumers visit this page to rate and add opinions about different digital cameras. Firstly, we explain how the ontology has been defined and the reviews is mapping into the ontology. Afterwards, the recommendation calculation and the solution to deal with the cold start problem are presented.

6.1. Representation of the Consumer Reviews - Digital Camera Ontology

First of all, we define an ontology. In computing ontology is a specification of an abstract, simplified view of the world that is wise to represent for some purpose [11]. Therefore, ontology defines a set of representational terms called concepts. Interrelationships among these concepts describe a target world. In this research, an ontology has been developed for digital camera domain, which is shown in Figure 5.

Each concept in the ontology was obtained analyzing the reviews from the consumers of different digital cameras from www.dpreview.com. Consumers can choose any digital camera and rate it on a scale of half start to four starts. They can also write free form text reviews about the camera. For the construction of digital camera reviews ontology, first was made a list of all possible objects necessary to cover given cameras reviews. This possible list should include different digital cameras such as Canon, Sony, etc. Furthermore, different cameras can be qualified by features such as size, zoom, lens, quality picture, etc. This information is represented by the concept “Features”. And the different consumer’s reviews can be qualified by opinions from beginners, professionals and by the level of expertise using digital cameras. This information is represented in the ontology by the concept “Opinion Quality”.

6.2. Mapping a review comment into an ontology

Once the ontology has been defined, it is necessary to match the information of the review with the ontology. We now show a new mapping to map the information into the ontology. Case study was conducted using a review add by a consumer in www.dpreview.com web. This review is:

**Canon PowerShot SD630 (nzjs, 16 Apr 06):**

You cant take a photo unless you are carrying a camera and this is a good one to carry all the time!

I have owned most IXUS cameras since the first - Just traded my Ixus 700 on this one (an Ixus 65 but still have an Ixus 40). The huge screen is excellent for framing shots and showing them off. I like the new low shutter speed showing on the screen and the grid lines. The screen is quite scratch resistant (unlike the 40 or 700 which scratch easily). Images are “good” (not stunning) better handheld in low light than the 700 but still blurry for distant features in landscapes. Gives excellent 6x4 prints on Canon CP330. The smaller 5-way selector button is touch sensitive and shows fancy icons on the screen that get bigger when you touch (not push) that side of the button - nice! The Flash is undersized but works well with closer subjects and has good balance for backlit shots. The bigger strap eye on this camera allows you to fit the strap of your choice. I use a black early Ixus strap that works better with my Jenova black leather Ixus belt case. Interesting to see other camera brands copying the Canon zoom button around the shutter release – works well on the Ixus 65/SD630 with the new low profile buttons. The slim mode switch on top is now needs the left hand. This is my “main” camera - and while there are times when I want a longer or wider lens I would not be draging along the big camera that they are attached to. Recommended - if there was a better camera this size I would have bought it! The Images are not that sharp but better than the 700. No manual ASA select and still no Battery.
power meter. Not so much room for the right thumb to grip the
Fig. 6 Mapped Ontology from a Consumer Review Comment

<table>
<thead>
<tr>
<th>Consumer Information</th>
<th>John</th>
<th>Karen</th>
<th>James</th>
<th>Laura</th>
<th>Andy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Opinion Quality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Consumer Skill</strong></td>
<td>Beginner</td>
<td>Professional</td>
<td>Beginner</td>
<td>Beginner</td>
<td>Professional</td>
</tr>
<tr>
<td><strong>Time to Use this Camera</strong></td>
<td>2 months</td>
<td>1 year</td>
<td>2 weeks</td>
<td>3 months</td>
<td>2 months</td>
</tr>
<tr>
<td><strong>Time to Use Digital Camera</strong></td>
<td>4 months</td>
<td>1 year</td>
<td>3 weeks</td>
<td>5 months</td>
<td>2 years</td>
</tr>
<tr>
<td><strong>Number of Different Cameras</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consumer Information</th>
<th>Bernat</th>
<th>Amadeus</th>
<th>Micky</th>
<th>Carmen</th>
<th>Benn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Opinion Quality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Consumer Skill</strong></td>
<td>Beginner</td>
<td>Beginner</td>
<td>Professional</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Time to Use this Camera</strong></td>
<td>2 months</td>
<td>1 months</td>
<td>2 weeks</td>
<td>3 moths</td>
<td>2 months</td>
</tr>
<tr>
<td><strong>Time to Use Digital Camera</strong></td>
<td>4 months</td>
<td>moths</td>
<td>3 weeks</td>
<td>8 moths</td>
<td></td>
</tr>
<tr>
<td><strong>Number of Different Cameras</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

### TABLE 5
**Ontology Instances Mapped from Consumer Reviews**

<table>
<thead>
<tr>
<th>Consumer Information</th>
<th>Camera</th>
<th>Opinion Quality</th>
<th>Product Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Camera</strong></td>
<td></td>
<td><strong>Features</strong></td>
<td><strong>Product Quality</strong></td>
</tr>
<tr>
<td><strong>Features</strong></td>
<td>Size: good</td>
<td>Zoom: good</td>
<td><strong>Size</strong>: good</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>bad</td>
<td><strong>weight</strong>: good</td>
<td><strong>Size</strong>: good</td>
</tr>
<tr>
<td><strong>Doc.</strong></td>
<td>good</td>
<td><strong>Material</strong>: bad</td>
<td><strong>Doc.</strong>: good</td>
</tr>
<tr>
<td><strong>Zoom</strong></td>
<td>good</td>
<td><strong>battery</strong>: bad</td>
<td><strong>Soft</strong>: bad</td>
</tr>
<tr>
<td><strong>battery</strong></td>
<td>good</td>
<td><strong>Software</strong>: good</td>
<td><strong>Start Up</strong>: good</td>
</tr>
</tbody>
</table>

back but practicing a different finger arrangement holding the camera in the fore-finger and thumb with other fingers clamped on the strap. The Auto-Rotate is such crap - who needs to see a smaller image rather than turn the camera? Mode switch in this position changes easily when pulling the camera out of it's case.

The next sections describe the classification process applied to the new review.

6.2.1 Classify each Sentence into One of “Good”, “Bad” and “Quality” Category

The set of rules obtained in the previous section is applied to each sentence of the new review to classify it into one category. For example the first sentence has been classified into the Good category. The second sentence has been classified into the Quality category as it is illustrated next:

Sentence 1: You can't take a photo unless you are carrying a camera and this is a good one to carry all the time!
Goods rules: rule 26
Bad rules: none
Quality rules: none
Classification: GOOD

Sentence 2: I have owned most IXUS cameras since the first - Just traded my Ixus 700 on this one (an Ixus 65 but still have an Ixus 40)
Goods rules: none
Bad rules: none
Quality rules: rule 5
Classification: QUALITY

Twenty sentences have been classified of which eight have been classified into the Good category, three into the Bad category, one into the Quality category, five that are irrelevant (those that nothing of the rules has been applied).

6.2.2 Finding the Concept Represented in the Sentence

For each sentence that is in the “Good” or “Bad” categories, the mapping feature of this sentence is found by searching the keywords that are in the related word list. For example in the first sentence, that has been classified as a good opinion. Also, a word has been found, which is related to the concept “size”. We suppose that the size of the camera is good for a customer so is assigned the value good for the feature size in the ontology. Follow are show some cases of the concept identification made in the case study.

Sentence 1: You can't take a photo unless you are carrying a camera and this is a good one to carry all the time!

Classification: GOOD
Ontology Concepts: size (related word: carry)

Sentence 14: This is my “main” camera - and while there are times when I want a longer or wider lens I would not be dragging along the big camera that they are attached to.
Classification: BAD
Ontology Concepts: lens (related word: lens)

Figure 6 shows the mapping of the new review into the predefined ontology.

6.3 Computing the recommendation

In this section, detail calculations of the recommendation in response to user request are given. Table 5 shows 10 ontology in-stances mapped from review comments, which are used for the recommendation calculations in the case study.

6.3.1 Obtaining the Opinion Quality (OQ) and Feature Quality (FQ)

The Opinion Quality is calculated by equation (1) in Section 4.2. Table 6 presents the weighting value of each variable defined in the equation.

<table>
<thead>
<tr>
<th>VARIABLES</th>
<th>Value</th>
<th>weight (wi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer Skill</td>
<td>Beginner</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Advanced</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>Professional</td>
<td>0.9</td>
</tr>
<tr>
<td>Consumer Experience</td>
<td>Day</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Week</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Month</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>Year</td>
<td>0.9</td>
</tr>
<tr>
<td>Time to use Digital Camera</td>
<td>Day</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Week</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Month</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>Year</td>
<td>0.9</td>
</tr>
<tr>
<td>Number of different Cameras</td>
<td>One</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Two</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Three</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>(+) Three</td>
<td>0.9</td>
</tr>
</tbody>
</table>

The OQ values for each consumer in Table 5 are:

\[
OQ_{Beginner} = \frac{0.5 + 0.7 + 0.7 + 0.5}{4} = 0.6 \\
OQ_{Advanced} = \frac{0.9 + 0.9 + 0.9 + 0.3}{4} = 0.75 \\
OQ_{Professional} = \frac{0.5 + 0.5 + 0.3}{3} = 0.43 \\
OQ_{User} = \frac{0.5 + 0.7 + 0.7 + 0.5}{4} = 0.6 \\
\]
With the calculated values, the best opinion came from Andy: note that Andy is a professional photographer; he has used digital cameras for longer period of time than the other consumers in the sample and he has used three different cameras. This information leads to the assumption that Andy’s opinion is the most valuable opinion within the sample.

Feature Quality (FQ) value for each feature rated by the consumers is also calculated. For example as shown in Table 7, John gave the value “good” or “bad” for each feature of the digital camera Sonny361 and his OQ value is 0.6.

As described in previous sections, by assigning the value 1 for “good” and -1 for “bad” in equation 2, the Feature Quality for each feature in John’s opinion are calculated, as shown in Table 8.

The same process has been applied to all consumers. The OQ and FQ for each review comment are calculated offline to achieve quick response to the user requests. The recommender system requires from the user to input the model of the camera he (she) is interested and selects the features that he (she) is most concern. The features in the selection panel are the same set of features that is covered by the ontology.

For example, a user request “I would like to know if Sony361 is a good camera, specifically its interface and battery consumption” is presented. Three keywords (Sony361, interface and battery) can be identified. Firstly, only the opinions for Sony361 are selected. Observing in Table 5, there are three opinions about Sony361’s cameras: John’s opinion, Karen’s opinion and James’s opinion. Then the OFQ of each feature is calculated using equation (3).

The Overall Assessment for the digital camera Sony361 based on the two features requested is obtained using equation (5). The Importance Index was calculated in two ways. For the case of using the importance index from the user request where the user has expressed that the interface is more important than the battery, so the value of 1 is assigned for interface and 0.5 for battery. Using these values the OA for Sony361 camera is:

\[ OA = -0.165 \times 1 + 0.18 \times 0.5 = -0.075 \]

In the case of no user preference is given, the importance index are calculated based on the frequency of the feature being re-viewed:

\[ Importance\ Index = \frac{n}{N} \quad (6) \]

Where n is the number of time that the feature appears in the re-views and N is the total number of reviews. Using equation (6), the OA for Sony361 camera is:

\[ Importance\ Index \text{ Int erface} = \frac{6}{10} = 0.6 \]
\[ Importance\ Index \text{ Battery} = \frac{2}{10} = 0.2 \]
\[ OA = -0.165 \times 0.6 + 0.18 \times 0.2 = -0.063 \]

Assigned the value “Good” for OA and OFQ > 0 and “Bad” for OA and OFQ< 0 the Sony361 camera is “Bad” according to consumers’ opinions. The response for the user request is shown in Figure 7.

The best camera with the features the user concern is also recommended. The same process is applied to all other cameras re-view. CannonTW45 is recommended considering...
this information. The complete recommendation is show in Figure 8.

![Figure 7 Recommendation in Response for a User Request from the Consumers' Opinions](image)

![Figure 8 Final Recommendation Answer to the User Request Generated from the Consumer's Opinions about Digital Cameras](image)

### 6.4. Handling the cold start problems in the system

This section illustrates the detail calculations of approximating the missing review data for the features with different data types used in the specification by a case study.

The evaluation of each feature of the products given by consumers is represented by the Overall Feature Quality, which is a function of the consumer skill level and the consumer experience with the product expressed in the review. Table 9 shows the consumer evaluation of four features of five digital cameras: Sony361, OlympusZ25, Olympus810, Olympus700 and CanonTW45.

<table>
<thead>
<tr>
<th>TABLE 9</th>
<th>OVERALL FEATURE QUALITY OF THE PRODUCTS CALCULATED FROM CONSUMER REVIEWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective pixels</td>
<td>Sony 361</td>
</tr>
<tr>
<td>Auto Focus</td>
<td>0.8</td>
</tr>
<tr>
<td>Flash modes</td>
<td>N/A</td>
</tr>
</tbody>
</table>

As consumers can evaluate only products they have experienced, so they commonly rate only a very small subset of all available items. Furthermore, for the products like cameras that have so many features, a review comment generally only provides opinions in a few features. In Table 9, the table cells that contain N/A represent no re-view comment on that feature is available. Therefore, the Overall Feature Quality of the feature is unable to be computed in a normal way. Table 10 presents the specification of these features of the cameras. By comparing the similarity of the specification, the missing evaluation can be estimated.

### 6.4.1. Estimating the rating for specifications with numerical values

Numerical data is the most common data type in the specifications, for example “Effective pixels” in Table 10. For this type of specifications, the Euclidean Distance is used to calculate the similarity of the specification between the products. The Euclidean Distance of two objects can be computed as:

$$S(X, Y) = ED(X, Y) = \sqrt{(X - Y)^2} = |X - Y|$$  \(7\)

According to equation (7), the similarity of Effective Pixels specification between cameras CanonTW45 and Sony361 is:

$$S_{\text{EffectivePixels}}(\text{CanonTW45},\text{Sony361}) = |6.0 - 4.0| = 0.2$$

The direction of the changes can be estimated by the following equation based on the assumption that each attribute in the feature has equal weight:
For CannonTW45 and Sony361, the direction of changes is:

\[ \text{Sign}_{\text{EffectivePixels}}(\text{CanonTW45}, \text{Sony361}) = \text{sign}(6.0 - 4.0) = 1 \]

The direction-weighted distance between models can then be calculated by multiplying the distance given in equation (7) by the direction of changes given in equation (8).

The similarities between CanonTW45 for the Effective Pixels feature with other cameras are shown in Table 11. The result is 0 for the same specifications and other value for different specifications.

**TABLE 11**

<table>
<thead>
<tr>
<th>Model</th>
<th>Effective Pixels (million)</th>
<th>Overall Feature Quality</th>
<th>Direction-weighted Distance from CanonTW45</th>
</tr>
</thead>
<tbody>
<tr>
<td>CanonTW45</td>
<td>6.0</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>Sony361</td>
<td>4.0</td>
<td>-0.3</td>
<td>2.0</td>
</tr>
<tr>
<td>Olympus225</td>
<td>6.0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>Olympus700</td>
<td>8.0</td>
<td>0.8</td>
<td>-2.0</td>
</tr>
<tr>
<td>Olympus810</td>
<td>8.2</td>
<td>0.7</td>
<td>-2.2</td>
</tr>
</tbody>
</table>

Since there is no other data from Canon, the overall feature quality of effective pixels for CanonTW45 is estimated using the complete data set from different manufacturers. This can be done by using linear regression based on effective pixels and overall feature quality:

\[ \text{OFQ}_{\text{EffectivePixels}, \text{CanonTW45}} = 0.52 \]

### 6.4.2. Estimating the rating for specifications with a list of properties

Some camera specifications are composed by a list of properties, for example the “Flash modes” has values such as Auto, On, Off, Slow, Manual and Red-Eye reduction. In this work, cosine distance is used to calculate the similarity between these types of specifications of two digital cameras.

The specification of two cameras represented by vector and that contain property sets and respectively. The cosine similarity function between these specifications treats the set of properties as components of a multi-dimensional vector, and the similarity is the cosine of the angle between these vectors. The similarity is given by the following expression:

\[ S(X, Y) = \cos(X, Y) = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}} \] (9)

As the specifications become more similar, the \( \cos(X, Y) \) come close to 1 and with different specifications the \( \cos(X, Y) \) come close to 0.

In the case study the specification “Flash modes” has the following set of properties:

\[ \{ \text{Auto, On, Of f, Slow, Manual, RedEye Reduction} \} \]

The vector representing the value for the specification is composed by binary values: 1 if this property appears in the specification and 0 for the property that does not appear in the specification.

According Table 10, Flash modes for OlympusZ25 has the values: Auto, Red-Eye reduction and off. The vector representing this information is:

\[ X = \{1, 0, 1, 0, 0, 1\} \]

The vector \( Y \) representing the Flash modes specifications for camera Sony361 is:

\[ Y = \{1, 1, 0, 0, 1\} \]

The similarity between the specifications of these two cameras is:

\[ S(\text{OlympusZ25, Sony361}) = \cos(X, Y) = \frac{(1*1)+(0*1)+(0*1)+(0*0)+(0*0)+(1*1)}{\sqrt{3} \sqrt{3}} = 0.867 \]

The camera that has the highest similarity with the target camera is chosen. As shown in Table 12, CanonTW45 has exactly the same specification with Sony361.

**TABLE 12**

<table>
<thead>
<tr>
<th>Model</th>
<th>Flash modes</th>
<th>Overall Feature Quality</th>
<th>Distance with Sony361</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sony361</td>
<td>Auto, On, Off, Red-Eye reduction</td>
<td>N/A</td>
<td>1</td>
</tr>
<tr>
<td>Olympus810</td>
<td>Auto, Red-Eye reduction, Off</td>
<td>0.72</td>
<td>0.86</td>
</tr>
<tr>
<td>OlympusZ25</td>
<td>Auto, Red-Eye reduction, Off</td>
<td>0.68</td>
<td>0.86</td>
</tr>
<tr>
<td>Olympus700</td>
<td>Auto, Red-Eye reduction, Off</td>
<td>0.74</td>
<td>0.86</td>
</tr>
<tr>
<td>CanonTW45</td>
<td>Auto, On, Off, Red-Eye reduction</td>
<td>0.85</td>
<td>1</td>
</tr>
</tbody>
</table>

Therefore, the Overall feature quality of flash modes for Sony361 is estimated by CanonTW45:
recommender system based on review comments in free form. To the best of our knowledge, this is the first attempt to build a recommender system that utilizes reviews in a recommender system is still in its infancy. The research work using online consumer review comments. The research work using reviews in a recommender system is still in its infancy. To the best of our knowledge, this is the first attempt to build a recommender system based on review comments in free form text. In [8,9] the authors use the reviews to give some explanation about the recommendation of a product. We have proposed a potentially and novel approach for the retrieval of review's information.

\[
\text{OFQ}_{\text{Flash Mode, Sony361}} = \frac{\text{OFQ}_{\text{Canon TW 45}}} {S(\text{Canon TW 45, Sony361})} = 0.85
\]

6.4.3. Estimating the rating for specification with Boolean values

Some specifications have Boolean values, for example if the camera has Digital Zoom or if it has Tripod Mount. These specifications have values either “Yes” or “No”. The distance of these features can either be 0 or infinity depends on the specification are the same or different respectively.

Table 13 shows the similarity between OlympusZ25 with other cameras. Since OlympusZ25 has auto focus, any other cameras that also have auto focus have 0 distances from Olympus25; otherwise, they have infinity distance.

**TABLE 13**

<table>
<thead>
<tr>
<th>Model</th>
<th>Auto Focus</th>
<th>Overall Feature Quality</th>
<th>Distance with Sony361</th>
</tr>
</thead>
<tbody>
<tr>
<td>OlympusZ25</td>
<td>Yes</td>
<td>N/A</td>
<td>0</td>
</tr>
<tr>
<td>Sony361</td>
<td>Yes</td>
<td>0.8</td>
<td>0</td>
</tr>
<tr>
<td>Canon TW45</td>
<td>No</td>
<td>-0.2</td>
<td>infinity</td>
</tr>
<tr>
<td>Olympus810</td>
<td>Yes</td>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>Olympus700</td>
<td>No</td>
<td>-0.5</td>
<td>infinity</td>
</tr>
</tbody>
</table>

The calculation is estimated by the average value of the products that have 0 distances with the target product. In this case, Sony361 and Olympus810 can be used. However, as stated in Section 3, the estimation is preferred to use the data from the same manufacturer. Therefore, only Olympus810 is used for the calculation. The Overall feature quality of OlympusZ25 is estimated as:

\[
\text{OFQ}_{\text{Auto Focus, Olympus Z25}} = 0.6
\]

7. CONCLUSIONS

This paper proposed a novel approach to creating recommendations in recommender systems, which utilizes online consumer re-view comments. The research work using reviews in a recommender system is still in its infancy. To the best of our knowledge, this is the first attempt to build a recommender system based on review comments in free form text. In [8,9] the authors use the reviews to give some explanation about the recommendation of a product. We have proposed a potentially and novel approach for the retrieval of review’s information.

This paper makes three mayor contributions:

- An ontology to translate the information from the reviews into structured form that is suitable for processing by the recommender system.
- An automatic ontology mapping process using text mining techniques at a sentence level.
- A ranking mechanism for prioritizing the product quality with respect to the consumer level of expertise and the rating given to some features of the product has been developed. A set of measures such as Opinion Quality (OQ), Feature Quality (FQ), Overall Feature Quality (OFQ) and Overall Assessment (OA) have been defined to select the relevant reviews and provide the best recommendation in response to a user request. The recommendation is given based on these measurements.

This paper also addresses the cold start problem for the recommender system that based on consumer product reviews. Solutions are provided for each type of feature specifications, namely numerical data, property list and Boolean property.

In the case study, an ontology has been defined for the domain of digital camera reviews and has been used for demonstration of the work with some examples. The case study shows that the information extracted from unstructured data contained in the re-views can be mapped into predefined ontology using text mining techniques based on decision rules.

The implementation of this method allows a recommender system to use valuable textual information for recommendation.

The analysis of the experimental results was carried out with 195 “Good” comments, 127 “Bad” comments and 47 “Quality” comments from 68 user reviews of digital cameras. The following conclusions about the mapping process can be drawn.

- The comments used in the case study are all for one model of camera (Canon Power-Shot SD500). The recall and precision measures can be further improved in the classification process by using multiple models.
- “Good” category contains more training data than other categories so it achieved the best results amongst.
- There are some long and complicated sentences that cannot be classified into any category. These sentences should be broken into several short sentences before the classification.

Despite of these issues, the results obtained are considered good as we can accurately map a large portion of a review into the predefined ontology.

Case studies also have been developed to illustrate the detailed calculations of solution for the cold start problem. Although the algorithms were developed specifically for the recommender system based on consumer product reviews, this approach can be generalized to other collaborative filtering recommender systems.

In conclusion, in this paper, we have provided a new application of reviews to recommender systems. The experimental study shows that it is a promising approach. In our
future work, we intend to evaluate our approach with the intended consumer groups.

REFERENCES


