Ontological Filtering for Sentiment Analysis

Francesco Colace¹, Massimo De Santo¹, Paolo Napoletano¹, C. Becchi², S.K. Chang³

¹Department of Electronic and Information Engineering, University of Salerno, 84084 Fisciano, Italy
{fcolace, desanto, pnapoletano1}@unisa.it
²77Agency, Dzirnavu 57A - 1, 1010, Riga, Latvija
c.becchi@77agency.com
³Computer Science Department – University of Pittsburgh, Sennott Square, Pittsburgh, PA 15260 USA
chang@cs.pitt.edu

Abstract

The rapid growth of the internet has increased the number of online reviews, opinions and sentiments toward products, services or topics. People appreciate this opportunity so that e-Commerce websites provide services for users to publish their reviews. Social networks, blogs and websites enable, thanks to the reviews, a social structure that provides benefits for the users and the firms that hosts electronic markets. Therefore, this huge quantity of information can confuse users and does not produce useful knowledge. In such a context, in fact, who says what and how they say it, matters. In this scenario a valuable contribute can be given by the sentiment analysis that is one of the hottest current research area. This paper presents a novel approach to the sentiment analysis which is based on the ontological filtering approach. The proposed approach shows how to automatically mine, from a corpus of documents, positive and negative sentiments. Experimental evaluations, on real dataset, show that the proposed approach is effective and furnishes interesting results.

1. Introduction

During a decision making process people have to consider many things but there is a moment when usually think: “what other people think”. Before the wide diffusion of the internet and web2.0 services people share opinions and recommendation by the use of traditional approaches: asking to friends, talking with experts and reading documents. The internet and the web have made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well known professional critics. The interest that individual users show in online opinions about products and services and the potential influence such opinions wield, is something that vendors of these items are paying more and more attention to. Companies are interested in what users are saying about their products as politicians are interested in how different news media are portraying them. Therefore there is a lot of information on the web that have to be properly utilized in order to provide vendors highly valuable network intelligence and social intelligence to facilitate the improvement of their business. In this scenario a very interesting approach is the sentiment analysis. More in general sentiment analysis is the computational study of opinions, sentiments and emotions expressed in text [1]. Sentiment classification is part of the opinion mining and refers to the identification of opinions and arguments in a text. Its main aim is the identification of the agreement or disagreement statements that deal with positive, negative or neutral in comments or reviews. There are many approaches to the sentiment analysis. A very broad overview of the existing work was presented in [2]. The authors describe in a very detailed way the main techniques and approaches for an opinion oriented information retrieval. Early work in this area was focused on determining the semantic orientation of documents. In particular some approaches attempt to learn a positive-negative classifier at the document level. In [3] three machine learning approaches (Naïve Bayes, Maximum Entropy and Support Vector Machines) has been adopted to label the polarity of a movie reviews datasets. A promising approach has been developed in [4] where a novel methodology has been obtained by the combination of rule based classification, supervised learning and machine learning. In literature some approaches are based on a computational approach to inferring the sentiment orientation of social media content and estimate sentiment orientations of a collection of documents as a text classification problem [5]. More in general according to these approaches sentiment related information can be encoded lexically within the actual words of the sentence syntactically and morphologically through changes in attitudinal shades of word meaning using suffixes [6]. This approach has been investigated in [7] where a lexicon for sentiment analysis has been obtained. Another interesting approach is in [8] where a probabilistic approach to sentiment mining has been adopted. In particular this approach adopt a probabilistic model called Sentiment Probabilistic Latent Semantic Analysis (S-PLSA) in which a review, and more in general a document, cab be considered as being generated under the influence of a number of hidden sentiment factors [9]. The S-PLSA is an extension of the PLSA where it is assumed that there are a set of hidden semantic factors or aspects in the documents that are related to documents and words
under a probabilistic framework. In [10] an approach combining the ontological formalism and a machine learning technique has been proposed. In particular the proposed system starting from a sentence uses domain ontology to extract the related concepts and attributes and then by the use of the Support Vector Machine (SVM) classifier for labelling it positive, negative or neutral. The last two papers are the starting point for the approach proposed in this paper. We propose a methodology for the sentiment analysis based on the combined use of probabilistic techniques (Latent Dirichlet Allocation and ontological formalism). In this way the document can be automatically filtered according their sentiment value.

The rest of the paper is organized as follows. Section 2 provides a brief introduction to the ontological filtering. The section 3 discusses the Latent Dirichlet Allocation and its use in the ontology building process. The section 4 shows the proposed sentiment analysis approach based on the use of the ontological filtering and the section 5 discusses the experimental results. The conclusions section concludes this paper.

2. Ontological Filtering

As previously said this section introduces the concept of ontological filtering. In order to introduce this concept an example can be showed. A user aims to make a search in a web repository that refers to a well-defined domain. Usually the user expresses his query by the use of keywords. In this way the search is conducted using the keywords and counting their occurrences (mutual or not) in the documents. This is a simple syntactic search. The ontological filter, besides, works in the following way: the user still search for by the use of keywords, we can navigate the domain through the topics that are in. In fact, we can assume that each keyword can be associated to various concepts. For example, in ontology related to the sport the keyword “ball” could be associated to various kinds of sports. In this way, by the use of keywords, we can navigate the domain through the topics that are in. In other words by the use of the keywords we can obtain a lightweight ontology which can be considered a sort of sieve for the searching of documents. In fact, in this way the system can give to the user only the documents that belong to certain concepts that are in this new ontology. In this case also documents that do not contain keywords but are related to the concepts that contain ones will be selected. This approach can be easily generalized to each kind of problem. For example in the case of a video surveillance system, each component can be part of some topics and can be identified by some keywords. In this way the request of the user can be easily mapped as a set of topics belonging to ontology and an answer to a user request. The ontology filtering approach introduces a new layer the interaction between users and data represented by the ontology. A more formal definition of the Ontology Filtering can be developed in the following way. First of all, ontology representing the domain of the knowledge involved in the problem has to be defined. This ontology can be obtained both by the support of experts both by the use of automatic methodologies able to infer the description of the domain by the analysis of related data, as we will show in the section 3 of this paper. In this way the \( O = \{C, A, H, R_T, R\} \) (where \( C \) is the set of concepts, \( A \) the set of the attributes, \( H \) the set of the hierarchical relationships, \( R_T \) the set of non-hierarchical relationships and \( R \) the set of semantic relationships) can represent the domain of interest and in particular the \( A \) attributes set contains both the set of keywords and the “items”. The user introduces its query by the use of keywords \( K_U = \{k_u\} \). In this way the following strategy could be followed in order to build the ontology:

Step 1: \( \forall k_{ui} \in K_U \) and \( \in A_{Ci} \) add in the set \( C' \) of the Ontology \( O'_{L+} = \{C', A', H', R'_T, R'\} \) the concept \( C_i \) belonging to an ontology \( O_{L+} = \{C, A, H, R_T, R\} \). The set \( A_{Ci} \) contains the attributes related to the concept \( C_i \) and is a subset of \( A \).

Step 2: \( \forall C_i \in C \) add in the set \( C' \) of the Ontology \( O'_{L+} = \{C', A', H', R'_T, R'\} \) all concepts \( C_j \) that share relations of kind \( H, R_T \) and \( R \) with the nodes \( C_i \). At the same time these relations have to be inserted in the sets \( H', R'_T \) and \( R' \) and all the attributes of the various \( C_j \).

At the end of these two steps the \( O' = \{C', A', H', R'_T, R'\} \) is the ontology that expresses the domain according to the requests of the user. In this way the ontological filter can be developed: in fact all the “objects” that are in a complete domain can be filtered by the use of the ontology \( O'_{L+} \) obtaining only the objects that user needs. This approach is able to answer the request of the user determining in an automatic way all the “items” that need for the resolution. Thanks to the relations, besides, that are among the various concepts that are in the ontology a use order can be inferred. The problem is how to infer in an automatic way the ontology of a domain. In the next section an approach for the unsupervised ontology learning by the use of the Latent Dirichlet Allocation will be introduced.

3. Ontology Extraction by the use of LDA

In statistics, latent Dirichlet allocation (LDA) is a generative model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA is an example of a topic model and was first presented as a graphical model for topic discovery in [10]. In general this approach is based on the concept of query expansion in a vector of features. The vector of features
A mixed Graph of Terms (mGT) is a hierarchical structure composed of two levels of information represented through a directed and an undirected sub-graph: the conceptual and word levels. We consider extracting it from a corpus \( \mathcal{D} = \{w_1, w_2, \ldots, w_M\} \) of \( M \) documents (that we call the training set), where each document is, following the Vector Space Model [16], a vector of feature weights \( w_j = (w_{1j}, \ldots, w_{T_j}) \), where \( \mathcal{T} = \{t_1, \ldots, t_T\} \) is the set of features that occur at least once in at least one document of \( \mathcal{D} \), and \( 0 \leq w_j \leq 1 \) represents how much the feature \( t_k \) contributes to the semantics of document \( w_j \). We choose to identify features with words, that is the bags of words assumption, and in this case \( t_k = v_k \), where \( v_k \) is one of the words of a vocabulary \( \mathcal{T} \). The word level is composed of a set of words \( v_i \) that specify through a directed weighted edge the concept \( c_i \) (see fig. 1(b), tab. 1 and fig. 2(a)), or better the centroid of such a set (group or cluster), which is, therefore, still lexically denoted as a word. The weight \( w_{ij} \) can measure how far a word is related to a concept, or how much we need such a word to specify that concept, and it can be considered as a probability: \( \frac{w_{ij}}{\|w_i\|} = \mathcal{R}(c_i | v_j) \). The resulting structure is a sub-graph rooted on \( c_i \). Alternatively, the conceptual level is composed of a set of interconnected, through undirected weighted edges, concepts \( c_i \) (see fig. 1), so forming a sub-graph of pairs of centroids. The weight \( \gamma_{ij} \) can be considered as the degree of semantic correlation between the two concepts and it can be considered as a probability: \( \mathcal{R}(c_i, c_j) \).

### 3.1.1. Graph drawing

A mGT is well determined through the learning of the weights, the Relation Learning, and through the learning of the three parameters, the Parameter Learning, that is \( \mathcal{D} = (\mathcal{H}, \mathcal{E}, \mathcal{X}) \) which specifies the shape of the graph. In more details, we have:

1. \( \mathcal{H} \): the number of concepts (namely the number of clusters) of the corpus \( \mathcal{D} \);
2. \( \gamma_{ij} \): the threshold that establishes for each concept the number of edges of the directed sub-graph, and so the number of concept/word pairs of the corpus \( \mathcal{D} \). An edge between the word \( s \) and the concept \( i \) can be saved if \( \gamma_{is} \leq \gamma_{ij} \). We consider, to simplify the formulation, \( \gamma_{ij} = \gamma_i \leq \gamma_{i} \); 
3. \( \gamma \): the threshold that establishes the number of edges of the undirected sub-graph, and so the number of concept/concept pairs of the corpus \( \mathcal{D} \). An edge between the concept \( i \) and concept \( j \) can be saved if \( \gamma_{ij} \leq \gamma \).

### 3.1.2. Relations Learning

Due to the fact that each concept is lexically represented by a word of the vocabulary, then we have
that $\mathcal{R}_i = \mathcal{R}(c_i | v_j) = \mathcal{R}(v_j | c_i)$, and $\mathcal{R}_j = \mathcal{R}(c_j, c_j) = \mathcal{R}(v_i, v_j)$.

As a result, we can obtain each possible relation by computing the joint probability $\mathcal{R}(v_i, v_j)$, which can be considered as a word association problem and so can be solved through a smoothed version of the generative model introduced in [10] called Latent Dirichlet allocation, which makes use of Gibbs sampling [12].

### 3.1.3. Parameters Learning

Given a corpus $\mathcal{D}$, once each $\mathcal{R}_i$ and $\mathcal{R}_j$ is known for $i, j, s$, letting the parameters assume a different set of values $\mathcal{R}_{it}$, we can observe a different graph $mGT_{it}$, where $t$ is representative of different parameter values.

A way of proving that a $mGT$ is the best possible for that set of documents is to demonstrate that it produces the maximum score attainable for each of the documents when the same graph is used as a knowledge base for querying in a set containing just those documents which have fed the $mGT$ builder.

Each graph $mGT_{it}$ can be represented, following again the Vector Space Model [16], as a vector of feature weights, that we call $\mathbf{q}_t$, and is defined as $\mathbf{q}_t = (w_1, ..., w_{|T_p|})$, where $|T_p|$ represents the total number of pairs. We have that each feature $t_k = (v_i, v_j)$, which is not the simple bags of words assumption, and $w_{tk}$ being the weight calculated thanks to the tf-idf model applied to the pairs represented through $t_k$, and with the addition of the boost $b_k$ which is the semantic relatedness between the words of each pair, at both the conceptual and the word level, namely $\mathcal{R}_i$ and $\mathcal{R}_j$.

You will recall that both $\mathcal{R}_i$ and $\mathcal{R}_j$ are real values (probabilities) of the interval $[0,1]$, and so to distinguish the relevance between the three cases, the traditional case ($b_i = 1$), the concept/word pair and the concept/concept pair, we have distributed such values with a wider interval. Specifically:

1. $b_i = 1$ being the lowest level of relatedness;
2. $b_i = 1$ with $\mathcal{R}_i, \mathcal{R}_j$ max and $\mathcal{R}_i = 1$;
3. $b_i = 1$ with $\mathcal{R}_j = 1$ and $\mathcal{R}_i = 1$.

In the experiments we have chosen $\mathcal{R}_i = 1$ and $\mathcal{R}_j = 3$ (see table 1).

<table>
<thead>
<tr>
<th>Conceptual Level</th>
<th>Concept $i$</th>
<th>Concept $j$</th>
<th>Relation Factor ($\mathcal{R}_{ij}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tank</td>
<td>tank</td>
<td>4.0</td>
<td></td>
</tr>
<tr>
<td>tank</td>
<td>water</td>
<td>3.72746</td>
<td></td>
</tr>
<tr>
<td>tank</td>
<td>liquid</td>
<td>3.13853</td>
<td></td>
</tr>
<tr>
<td>liquid</td>
<td>type</td>
<td>3.43828</td>
<td></td>
</tr>
<tr>
<td>liquid</td>
<td>pressur</td>
<td>3.07028</td>
<td></td>
</tr>
</tbody>
</table>

### Table 1. An example of a $mGT$ for the topic Storage Tank.

At this point, a document $\mathbf{w}_i$ can be viewed as a vector of weights in the space $|T_p|$, and so the general formula of each weight is:

$$w_{ij} = \frac{\text{tf-idf}(t_i, w_j) \cdot b_i}{\sqrt{\sum_{j=1}^{|T_p|} (\text{tf-idf}(t_i, w_j) \cdot b_j)^2}}$$

(1)

The score for each graph at time $t$, namely $\mathcal{S}_t$, can be computed following the cosine similarity model in the space $|T_p|$, and so we have:

$$\mathcal{S}_t(q_t, \mathbf{w}) = \frac{\sum_{i=1}^{|T_p|} w_{ij} w_{ji}}{\sqrt{\sum_{i=1}^{|T_p|} w_{ij}^2 \cdot \sum_{j=1}^{|T_p|} w_{ji}^2}}$$

(2)

Finally for the graph at time $t$ we have a score for each document, $\mathcal{S}_t = (\mathcal{S}(q_t, \mathbf{w}_1), ..., \mathcal{S}(q_t, \mathbf{w}_m))$.

As a result, to compute the optimum set of parameters $\mathcal{R}$, we can maximize the Fitness ($\mathcal{F}$),

$$\mathcal{F}(\mathcal{A}) = \arg \max_{\mathcal{A}} \{\mathcal{F}(\mathcal{A})\},$$

where $\mathcal{F}(\mathcal{A}) = E_m \left[ \mathcal{S}(q_t, \mathbf{w}_m) \right] - \sigma_m \left[ \mathcal{S}(q_t, \mathbf{w}_m) \right]$. $E_m$ is the mean value of all elements of $\mathcal{S}_t$ and $\mathcal{R}_m$ is the standard deviation. Since the space of possible solutions could grow exponentially, we have limited the space to $|T_p| < 150$, $\mathcal{A}$. Furthermore, we have reduced the remaining space of possible solutions by applying a clustering method, that is the K-means algorithm, to all $\mathcal{R}_i$ and $\mathcal{R}_j$ values, so that the optimum solution can be exactly obtained after the exploration of the entire space. This reduction allows us to compute a $mGT$ from a repository composed of a few documents in a reasonable time (e.g. for 3 documents it takes about 3 seconds with a Mac OS X based computer and a 2.66 GHz Intel Core i7 CPU and a 8GB RAM). It is important to make clear that the mixed Graph of Terms cannot be considered as a co-occurrence matrix. In fact, the core of the graph is the probability $\mathcal{R}(v_i, v_j)$, which we regard as a word association problem, which in the topic model is considered as a problem of prediction: given that a cue is presented, which new words might occur next in that context? It means that the model does not take into account the fact that two words occur in the same document, but that they occur in the same document...
when a specific topic (and so a context) is assigned to that document [12]. Furthermore, in the field of statistical learning, a similar structure has been introduced, with the name Hierarchical Mixture of Experts [17]. Such a structure is employed as a method for supervised learning and it is considered as a variant of the well-known tree-based methods. The similarity between such a structure and the proposed graph can be obtained by considering the "experts" as "concepts". Nevertheless, the mixed Graph of terms is not a tree structure, and more importantly is not rigid but is dynamically built depending on the optimization stage. Moreover, the Hierarchical Mixture of Experts does not consider relations between experts which are, on the other hand, largely employed in the mixed Graph of Terms. Notwithstanding this, we will explore further connections between these two structures in future works.

4. A Sentiment Analysis Approach based on the use of the ontological filtering

In this section the proposed approach will be introduced and described in details. As previously said the methodology aims to combine the LDA and the ontological filtering approaches in order to realize an automatic sentiment detector able to label the mood of a document or a collection of them. The proposed approach is composed by the following steps:

- the first step is composed by the building process of ontologies representing a set of documents labeled according their sentiment. In other words by the use of the LDA approach the ontologies representing the positive, negative and neutral sentiment documents. The process is depicted in figure 2. These ontologies, that will name mood ontologies, contain the concepts, the words and their combination representing positive/negative and neutral sentiments. Thanks to the LDA approach it is possible to achieve representative ontologies by the use of few documents.

Figure 2 The Ontology Mood Building tool

- the second step is based on the introduction of a tool that uses the obtained ontologies as ontological filters. In particular this tool accepts as input a set of documents (or a single document) and labels it according the detected sentiment (positive/negative/neutral) by the use of the ontological filtering (figure 3).

Figure 3 The Sentiment Analysis Tool

In particular the sentiment is detected according to the following formula:

\[ \text{Mood}_i = \begin{cases} 1 & \text{if } f(D_i, O_+) > |f(D_i, O_-)| \text{ and } f(D_i, O_+) > f(D_i, O_N) \\ -1 & \text{if } |f(D_i, O_-)| > f(D_i, O_+) \text{ and } |f(D_i, O_-)| > f(D_i, O_N) \\ 0 & \text{in the other cases} \end{cases} \]

where Mood$_i$ is the sentiment associated to the document D$_i$. The function f detects the sentiment of the document in this way:

1. set \( f = 0 \)
2. for each word that is in the document that matches a concept in the ontology increase of 2 the value
3. for each word that is in the document and in the ontology increase of 1 the value
4. for each word that is not in the ontology a research in a lexical dictionary, as Wordnet [18] for English language or IWN [19] for Italian language, has to be conducted. If there is a link between the word belonging to the document and a word or a concept increase of 1 the value of the function f

The overall mood is obtained in the following way.

\[ \text{Mood} = \sum_{i=1}^{n} \text{Mood}_i \]

If Mood is >0 the overall mood of the documents’ set is positive else if it is <0 the mood is negative. If the mood is 0 the overall sentiment is neutral. In order to assess the degree of positivity and negativity the mood can be normalized on the overall number of documents.

\[ SD = |\text{Mood}|/#\text{Documents} \]

SD can assume values in the range [0, 1]. Values close to 1 mean high degree of positive or negative sentiments while values close to 0 mean very low sentiment in the documents.

5. Experimental Results

To evaluate the proposed sentiment classification approach an experimental campaign on a real dataset has been conducted. The dataset has been organized
collecting post on the official blog of Samsung on the product Samsung Galaxy Tab. The dataset has been composed by 2,000 posts in Italian language and has been collected in three months (November 2011 – January 2012). The posts usually are a series of comments on the product and its evolution. An example of post is the following (that is one of the longest): “Un regalo meraviglioso...uno stupendo GALAXY TAB :-D ed è una vera figata......lo consiglio a tutti: semplice nell’uso, ottimo nelle prestazioni e mi permette di fare tutto quello che voglio senza le assurde limitazioni che mette la concorrenza... ;-))”. In this case the user is delighted by his phone. First of all the full dataset has been labeled by three experts splitting the posts in positive, negative and neutral. At this point a training test of 100 positive posts, a training set of 100 negative posts and a training set of 100 neutral posts have been obtained selecting casually posts from positive, negative and neutral datasets. In this way a test set of 459 positive posts, a training set of 384 negative posts and a training set of 857 neutral posts have been obtained. In order to test the approach two experimental campaigns has been conducted. The first aimed to classify the documents according their sentiment. By the use of the three training set the mood ontologies has been built (figure 4).

**Figure 4 An example of positive sentiment ontology**

We used them in order to label the various documents. In particular the full test set (positive test set, negative test set and neutral test set) has been labeled by the use of the various ontological filters. Introducing the concept of Precision defined as:

\[
\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}
\]

And Recall defined as

\[
\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}
\]

the obtained results are depicted in table 2

<table>
<thead>
<tr>
<th>Sentiment</th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
<th>Dataset 4</th>
<th>Dataset 5</th>
<th>Dataset 6</th>
<th>Dataset 7</th>
<th>Dataset 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>270</td>
<td>225</td>
<td>180</td>
<td>100</td>
<td>50</td>
<td>80</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Negative</td>
<td>20</td>
<td>50</td>
<td>80</td>
<td>100</td>
<td>50</td>
<td>180</td>
<td>225</td>
<td>270</td>
</tr>
<tr>
<td>Neutral</td>
<td>10</td>
<td>25</td>
<td>40</td>
<td>100</td>
<td>200</td>
<td>40</td>
<td>25</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 2 Obtained Results for the case of document sentiment classification

The second experimental campaign has aimed to evaluate the skill of the system in the sentiment classification of a set of documents. This feature of the system can be very interesting for the evaluation of a stream of posts as for example Twitter. So we created in a random way 8 datasets of 300 posts organized as described in table 3.

<table>
<thead>
<tr>
<th>Sentiment</th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
<th>Dataset 4</th>
<th>Dataset 5</th>
<th>Dataset 6</th>
<th>Dataset 7</th>
<th>Dataset 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trend</td>
<td>Positive</td>
<td>Positive</td>
<td>Positive</td>
<td>Positive</td>
<td>Negative</td>
<td>Negative</td>
<td>Negative</td>
<td>Negative</td>
</tr>
<tr>
<td>SD</td>
<td>0.83</td>
<td>0.65</td>
<td>0.57</td>
<td>0.52</td>
<td>0.54</td>
<td>0.61</td>
<td>0.69</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 3 Configuration of the various datasets

The various datasets has been evaluated by the sentiment classifier based on ontological filtering with the following results (table 3).

Also in this case the system offers good results

6. Conclusions

This paper proposes the combination of the LDA approach and ontological filtering for the enhancement of the sentiment classification. Thanks to the LDA approach an ontology representing a set of document with a well-defined sentiment can be obtained. These ontologies can be adopted as ontological filtering and so detect the sentiment in a document or in a set of...
documents. The first results on real datasets are quite interesting. Further development of this approach is still ongoing and a more detailed experimental campaign on standard datasets will be arranged.

References

[8] Xiaohui Yu, Yang Liu, Xiangji Huang, Aijun An, “Mining Online Reviews for Predicting Sales Performance: a Case Study in the Movie Domain”, IEEE Transactions on Knowledge and Data Engineering, 24-4, 2012
[16] P. R. Christopher D. Manning and H. Schtze, Introduc-

tion to Information Retrieval. Cambridge University, 2008.