
Sentiment-Based Semantic Rule Learning for Improved Product Recommendations

Dandibhotla Teja Santosh and
Bulusu Vishnu Vardhan

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.72514>

Abstract

Crucial data like product features and opinions that are obtained from consumer online reviews are annotated with the concepts of product review opinion ontology (PROO). The ontology with instance data serves as background knowledge to learn rule-based sentiments that are expressed on product features. These semantic rules are learned on both taxonomical and nontaxonomical relations available in PROO ontology. These rule-based sentiments provide important information of utilizing the relationship among the product features 'as-a-unit' to improve the sentiments of the parent features. These parent features are present at the higher level near the root of the ontology. The sentiments of the related product features are also improved. This approach improves the sentiments of the parent features and the related features that eventually improve the aggregated sentiment of the product. The result is either the change in the position of the product in the list of similar products recommended or appears in the recommended list. This helps the user to make correct purchase decisions.

Keywords: recommender system, product feature, feature sentiment, ontology, rule-based sentiments, purchased decision

1. Introduction

Traditional machine learning algorithms experience the data and learn the hypothesis. Tree and rule-based algorithms learn the hypothesis using the attribute-value pairs from the input data. Machine cannot go beyond the task of identifying features and opinions from the reviews as it never possess prior knowledge to understand the relationships among the attributes and context specific constraints that are available among the product features and opinions.

Semantic web ontology helps to overcome this problem. Ontology [1] encodes the relationships among the concepts of features and opinions with inequality constraints, semantic

characteristics, and cardinality restrictions. This ontology is used as background knowledge on the product reviews. The knowledge mined from the ontology is expressed in the form of semantic rules. These semantic rules emphasize the target sentiment expressed on the product feature. Machines are able to classify the product reviews automatically with exact sentiments learned on the product feature.

Sentiment analysis [2] plays a vital role in understanding the opinions from online reviews. It helps to understand the views of the people on the product, to take quick purchase decisions on the product, and to improve the availability of the product in the market. Online reviews affect the emotion of the readers. Measuring the effect of the sentiment on the semantic rules in the form of knowledge spread is performed to understand whether positive reviews of the product spread faster than negative reviews. The kind of emotions that are more representative on various e-commerce sites about the product is also well identified. Furthermore, the type of sentiment expressed in reviews based on temporal changes on the features of the product is determined in a proper manner.

2. Literature survey

The recommender systems (RS) are the information filtering systems which deal with the large amount of information that is dynamically generated based on user's preferences, interests, and observed behaviors. These traditional recommender systems fall into three categories. They are collaborative filtering-based RS, content-based RS, and knowledge-based RS.

The collaborative recommender systems are the most popular and widely implemented systems. These systems aggregate ratings from the set of users on the item and recommend it. It also identifies the users who are similar with the user from whom recommendations are to be provided. Resnick et al. developed [3] a system called GroupLens to help people to find articles they are most interested in. Stavrianou and Brun developed [4] an application to recommend products based on the opinions and suggestions written in the online product reviews.

The content-based recommender systems learn the user profile based on the product feature where the user has targeted. Lang developed [5] a system called NewsWeeder which uses the words of the text as the features. Zhou and Luo developed [6] a content-based recommender system that views customer shopping history to recommend the similar products based on the similarity between the product features.

The knowledge-based recommender systems provide the entity suggestions based on the deductions from user's needs and preferences. These systems have the knowledge about how a particular product meets the customer requirement based on the factual data. The user profile is also required to provide good product recommendations to the user. Case-based reasoning (CBR) is a kind of knowledge-based recommender system. Kolodner used [7] CBR to recommend the restaurants based on the user's choice of features. Burke used [8] the FindMe system to recommend the online products. Stefan et al. worked [9] on user log data to mine the product preferences based on the like or dislike information available in the log.

Sentiment-based product recommendations have gained research importance in the recent times. The knowledge discovered in terms of product features and opinions from online product

reviews among the category of products are useful to the customer in personalized recommendations. These feature-level sentiments are aggregated to form the product sentiment. Chen and Wang proposed [10] a novel explanation interface that fuses the feature sentiment information into the recommendation content. They also provided the support for multiple products comparison with respect to similarity using the common feature sentiments. Gurini et al. proposed [11] friends recommendation technique in Twitter using a novel weighting function which is called sentiment-volume-objectivity (SVO) that considers both the user interests and sentiments. Xiu et al. proposed [12] a recommender system that recognizes the sentiment expressions from the reviews, quantified with the sentiment strength and appropriately recommend products according to customer needs. Recently, Dong et al. developed [13] a product recommendation strategy that combines both similarity and sentiments to suggest products.

The utilization of ontologies for better product recommendations is an emerging research area. Uzun and Christian developed [14] a semantic extension to FOKUS recommender system. This extension is capable of integrating contextual and semantic information in the recommendations. Hadi Khosravi and Mohamad Ali introduced [15] a semantic recommendation procedure using ontology on online products based on the usage patterns of the customers.

The works on ontology-based recommender systems [14, 15] was neither concentrated on utilizing the depth information of the domain feature nodes from the ontology tree nor on height of the ontology tree. These properties act as supervised weights in improving the sentiment of the feature and thereby help in improving the recommendations.

3. Improving product recommendations using semantic sentiments

The recommender system proposed in this work is a knowledge-based recommender system that encapsulates the product catalog knowledge in the form of classes in the ontology and product functional knowledge in the form of facts in the ontology. The user profile is created as and when the user navigates the web pages for the products. The user profile is indexed with the product information from the ontology.

The principal objective of recommending products using sentiments learned from the ontology is to utilize the taxonomical and non-taxonomical constraints mined from ontology for sentiments. The detailed procedures expressed in algorithmic form for learning taxonomical constraints and non-taxonomical constraints are presented below.

Input: PROO {Ontology}

Output: machine interpretable rule $\{A \rightarrow B\}$

EXTRACT_TAXONOMICAL_CONSTRAINT (Ontology)

```
{  
  for each concept with hierarchy from Ontology  
  {  
    contentconstraint = false;
```

```

    if(parent_of(superconcept, subconcept))
        contentconstraint = true;
        write(parent_of(superconcept, subconcept) →
target_class(subconcept));
        else if(parent_of(superconcept1, subconcept1) ^
parent_of(superconcept2, subconcept2))
            subconcept1 ←superconcept2;
            contentconstraint = true;
            write(parent_of(superconcept1,subconcept2) ^
datatype_property(superconcept1,rel(int)) →
target_class(subconcept2));
    }}

```

Algorithm for extracting taxonomical constraints from ontology

Algorithm for extracting taxonomical constraints runs as follows: given the PROO ontology, all the super concept and sub concept hierarchies are identified. Super concept node is called parent and sub concept node is called child. The rules are then obtained as of the predicate on the hierarchy as the relation between parent and child concepts leading to target class. Content constraint is initialized to the false value in the beginning. It is then changed to true value after taxonomical constraints are obtained. The Algorithm also tests for descendant child nodes in the hierarchy. A descendant node is a node which is derived from the ancestor node. An ancestor node is the parent node in the given hierarchy. All the child nodes for a given parent are known as descendant nodes. The intermediate parent node is devised as another child node to satisfy the descendant property. At this level, the content constraint value is changed to true. The rules are then obtained as of the predicate on the hierarchy as the relation between parent, newly created child and the datatype property leading to target class.

Input: PROO {Ontology}

Output: machine interpretable rule {A→B}

EXTRACT_NONTAXONOMICAL_CONSTRAINT (Ontology)

```

{
for each node in Ontology
    {
        contentconstraint = false;
        if(object_property(nodei, nodej))
            contentconstraint = true;
            write (object_property(nodei, nodej) →
target_class(nodej));
    }
}

```

```

else if(object_property(noder, nodei) ^
[datatype_property(noder,rel(int)) v
datatype_property (noder,rel(int))])
    contentconstraint = true;

write(object_property(noder,nodei)^datatype_property(noder,rel(int))
→ target_class(nodei));

write(object_property(noder,nodei)^datatype_property(noder,rel(int))
→ target_class(nodei);    }}

```

Algorithm for extracting nontaxonomical constraints from ontology

Algorithm for extracting nontaxonomical constraints runs as follows: given the ontology, all the related class nodes that are bound with the object properties are identified. Content constraint is initialized to the false value in the beginning. It is then changed to true value once related class nodes are obtained. The rules are then obtained with the object property as the relation between the related classes leading to the target class. The Algorithm also tests the relation between related classes and datatype properties. The related class node and the datatype property are associated using the conditions that are imposed on the ontology. This is identified by the algorithm. The content constraint value is changed to true. The rules are then generated from the relation between object property and the datatype property leading to the target class.

The PROO ontology concepts namely Opinion and Feature and the properties namely ObjectPart and ObjectPartFeature are used in generating the machine interpretable rules on target sentiment class. These articulate that they are the features to acquire positively oriented sentiment when the opinion strength on these features has a value greater than or equal to 2.5. The corresponding class hierarchies and the related classes of the PROO ontology are presented in **Figure 1**.

The sentiments of the product features that are present near to the root of the ontology are to be improved. The features located at the higher level near the root of the ontology are considered to be more important as compared with the lower level features [16]. The product features that are present near the ontology root are the parent features obtained from the taxonomical constraints and the other features present at the same level as the parent features. Other features are obtained from the non-taxonomical constraints. In order to achieve this goal, a framework is presented in **Figure 2**.

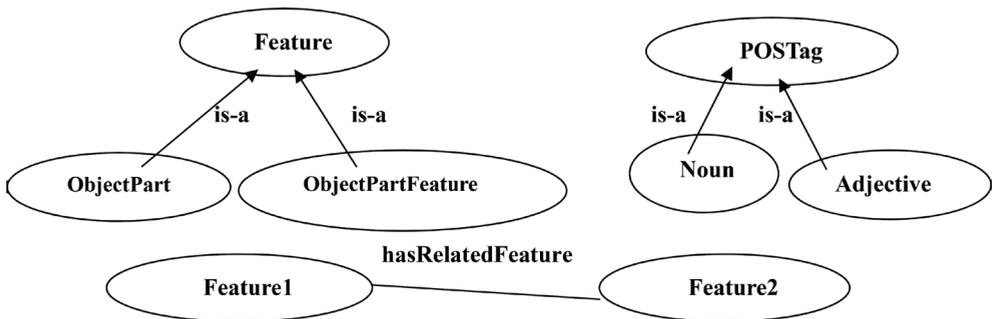


Figure 1. Class hierarchies and related classes in PROO.

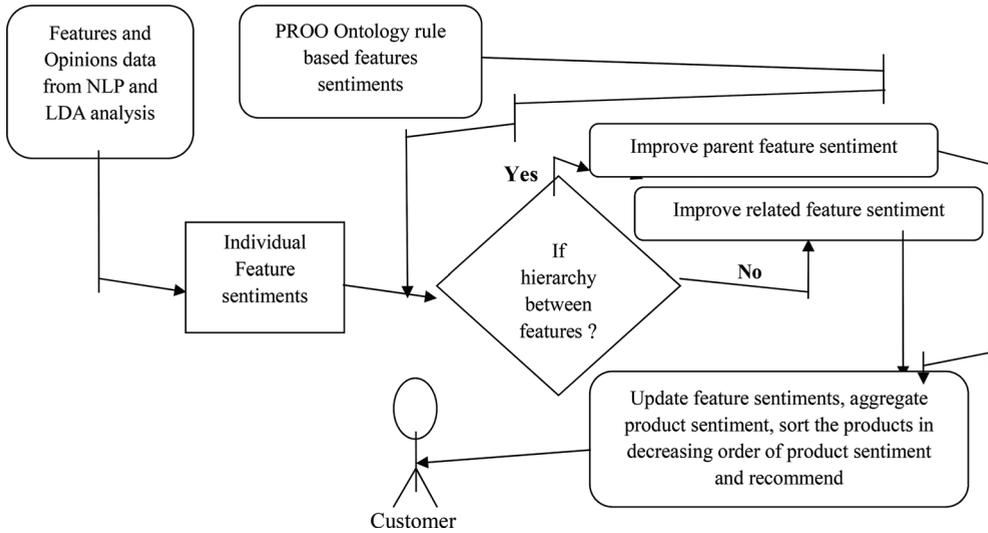


Figure 2. Model for improving the sentiments of the product features.

The framework is composed of main component. The improvement of sentiments of the product features using the knowledge mined from the PROO ontology for improved product recommendations is shown in a diagram as under. The first two modules, i.e., the development of PROO ontology and semantic data mining the PROO ontology, were already carried out by the researchers in their work in [17]. The proposed main component of improving the sentiments of the product features using the knowledge mined from the PROO ontology for better product recommendations is described below with the algorithm pseudo-code.

3.1. Improving the product recommendations using rule-based sentiments from ontology

The rule-based sentiments mined from the PROO ontology specify the relations between the parent feature and the child feature. It also reveals the relations among the related product features. The opinion strength of the feature for which the sentiment is to be determined by the machine also carries its importance in the rule. Also the sentiments calculated for each of the product features after extracting from the reviews are stored separately for further mapping. The detailed procedure for improving product recommendations is expressed in step-by-step form below. The symbols used in the steps are as follows: O is the PROO ontology. P_i is the product and $i = 1, 2, 3, \dots$. The sentiment of the product feature F_j of the product P_i is represented as $Sentiment(F_j, P_i)$ where $j = 1, 2, 3, \dots$. The $Pos(F_j, P_i)$, $Neg(F_j, P_i)$, and $Neu(F_j, P_i)$ are the positive, negative, and neutral product features. The $count()$ is the number of occurrences of polarity kind. $Parentof(F_{jkparent_node}, F_{jkchild_node})$ is the feature hierarchy in the ontology. $Objectproperty(node_a, node_b)$ is the fact about related product features. $Strength(node, rel(int))$ is the opinion strength of the feature which is present in the review. $Depth$ of the node

in the ontology and the height of the ontology are the ontology tree measures. The asterisk '*' in the steps represents the multiplication operator.

1. Retrieve the similar products from the ontology based on the user-searched product. The common features of retrieved products and the searched product are called as 'k-common features.'
2. For each of the k-common features, calculate the sentiment using the count of positive mentions and count of negative mentions on the features as:

$$\text{Sentiment}(F_i, P_i) = \frac{\text{count}(\text{Pos}(F_i, P_i)) - \text{count}(\text{Neg}(F_i, P_i))}{\text{count}(\text{Pos}(F_i, P_i)) + \text{count}(\text{Neg}(F_i, P_i)) + \text{count}(\text{Neu}(F_i, P_i))}$$

3. Retrieve taxonomical and non-taxonomical sentiment rules on the product features from ontology.
4. Map Rule_Positive_Sentiment = [0.001 ... 1] and Rule_Negative_Sentiment = [-1 ... -0.001].
5. For each k-common feature among all the similar products in ontology,

```

if (parentof(Fjkparent_node' Fjkchild_node') == true)
{
    if (Sentiment(Fjkparent_node' Pi) < Sentiment(Fjkchild_node' Pi))
    {
        Sentiment(Fjkparent_node' Pi) = Sentiment(Fjkparent_node' Pi) +
[Sentiment(Fjkchild_node' Pi) * depth of the Fjkchild_node'];
        New_Sentiment(Fjkparent_node' Pi) = Sentiment(Fjkparent_node' Pi);
    }
    if (Sentiment(Fjkchild_node' Pi) == Sentiment(Fjkparent_node' Pi))
        Continue;
}
else if (objectproperty(nodea, nodeb) ^ strength(node, rel(int)) == true)
{
    if (Sentiment(Fjknodea' Pi) <= 0)
    {
        Sentiment(Fjknodea' Pi) = Sentiment(Fjknodea' Pi) + height of the
ontology/100; /*Since to have small change in the score*/
        New_Sentiment(Fjknodea' Pi) = Sentiment(Fjknodea' Pi);
    }
}

```

```

    }
    if(Sentiment(Fjknodeb' Pi) <= 0)
    {
        Sentiment(Fjknodeb' Pi) = Sentiment(Fjknodeb' Pi) + height of the
ontology/100; /*Since to have small change in the score*/
        New_Sentiment(Fjknodeb' Pi) = Sentiment(Fjknodeb' Pi);

```

6. Sort the products in the descending order based on the enhanced sentiments of the k-common features.
7. Recommend products.

The explanation of the steps is as follows: given the product to be searched by the end user in the E-Commerce site, all the similar products are recommended. Initially, the algorithm retrieves all the similar products data from the ontology based with respect to the user-searched product. The common product features of retrieved products and the searched product are called as 'k-common features'. Next for each of the k-common features, the corresponding sentiment is calculated by using the number of positive mentions and number of negative mentions on the features. Whenever a neutral mention is identified, it is also counted and used in the sentiment calculation. Then the taxonomical and non-taxonomical sentiment rules on the product features are retrieved from the ontology. The target sentiment instances Positive and Negative are mapped to the minimum and maximum sentiment scores of the product features to create a sentiment range. Following discussions are the examples to clarify how the improved product recommendations are returned to the customer when a search for the product takes place. The dataset details for which the examples discussed were presented in **Table 1** which was presented in section V.

The product 'Samsung Galaxy j7 prime' has one of the taxonomical features as battery and battery life respectively. The number of positive mentions and negative mentions on the battery are 6 and 0. There are no neutral mentions. The number of positive and negative mentions on the battery life is 1 and 0. There are no neutral mentions. The sentiment scores obtained after calculation for battery and battery life are 1 and 1 respectively. The opinion strengths for battery and battery life obtained from review dataset are 3 and 3. By applying these features as instances in the taxonomical sentiment rule, the semantic sentiment learned is positive. The sentiment scores of battery and battery life are now mapped to Positive sentiment label.

Document attributes	Values
Number of review documents	300
Minimum sentences per review	9
Maximum sentences per review	15

Table 1. Reviews dataset details.

The product 'Samsung Galaxy j7 prime' has one of the non-taxonomical features as RAM and performance respectively. The number of positive and negative mentions on the RAM is 6 and 5. There are no neutral mentions. The number of positive and negative mentions on the performance is 6 and 2. There are no neutral mentions. The sentiment scores obtained after calculation for RAM and performance are 0.09 and 0.1 respectively. The opinion strengths for RAM and performance obtained from review dataset are 2.5 and 2.5. By applying these features and opinion strength values as instances in the non-taxonomical sentiment rule, the semantic sentiment learned is positive. The sentiment scores of RAM and performance are now mapped to Positive sentiment label.

The similar products are retrieved from the ontology by querying on the 'similarTo' object property for the corresponding instance values for the customer-searched product. Now for each k-common feature among all the retrieved products in ontology, whenever there exists any taxonomical constraints and when the sentiment of the parent feature node in the ontology is less than the sentiment of the child feature node then the sentiment of the parent feature node is updated by adding the weighted sentiment of the child feature node. The weight is the depth of the child feature node present in the ontology. This kind of analysis is possible as specified by [6], who say that the importance of the feature is determined by the depth of the feature in the ontology. This analysis views the taxonomical features 'as-a-unit.' Whenever the sentiment of the parent feature node is equal to the sentiment of the child feature node, then no update is carried out on these nodes.

Once all the taxonomical constraints are analyzed, the non-taxonomical constraints are also analyzed. The non-taxonomical constraints are analyzed to learn the related features and the contribution to their sentiment values. When the sentiments of the related nodes are less than or equal to zero, the sentiments of the related nodes are updated by adding the ratio. The ratio is 1/100th of the height of the ontology to make the score present in the sentiment range. The height of the ontology is added to the existing sentiment score as the related nodes are present at any level in the ontology other than the root.

The product 'Samsung Galaxy j7 prime' has sentiment scores obtained after calculation for battery and battery life is 1 and 1 respectively. There is no update in the sentiment value for either of the features. This is because the sentiment values for parent feature (battery) and child feature (battery life) which fall under taxonomical constraints are equal.

The product 'Samsung Galaxy j7 prime' has sentiment scores obtained after calculation for screen and display is 1 and 0 respectively. There is an update in the sentiment value for feature 'display'. This is because the sentiment value of display is equal to zero. The updated sentiment value for the feature 'display' is 0.03. The product features screen and display fall under non-taxonomical constraints.

Finally, the products are sorted in the descending order of the enhanced sentiments. The sorted list is provided as the product recommendations to the customer.

4. Design decisions in the implementation of ontology

The description logic (DL) is used in reasoning the instances of ontology. DL is the math behind the constructs of the ontology. The engineered PROO ontology has DL expressivity

level $ALCIN(D)$. $ALCIN(D)$ is attribute logic with complement, role inverse, unqualified number restriction and datatype. This ontology is robustly scalable and the rules learned from it are computationally solvable in polynomial running time, i.e., PTIME. The target sentiment which is learned as the rule consequent on the object properties of PROO ontology is decidable as the rules are deductible in the PTIME. Also the learned rules are DL-safe as these rules are restricted to known instances of the ontology.

There were some issues encountered at the time of PROO ontology development. This PROO ontology development was based on design decisions taken at two stages. The two stages were namely the design decisions made before the ontology development and, the decisions made at the time of ontology development.

The first design decision before the development of ontology was on the scope of the ontology to represent the appropriate knowledge for conceptualization. In the product reviews domain, the PROO ontology was intended to support the new customers in retrieving the object information from a large number of reviews by reasoning on object property ontology path. The second design decision was on adhering to the development of a formal ontology so as to reason the ontology for making meaningful conclusions. The PROO ontology was developed using the formal Web Ontology Language (OWL) constructs. The third design decision was whether to annotate the product features and opinions extracted from the reviews as instances to the concepts of the ontology or not.

The design decision taken during the development of ontology was to choose the required superclass-subclass taxonomies in the ontology. The taxonomies created in the development of PROO ontology were the hierarchy of the product features and the PoS word class tags. For some queries on PROO ontology, it was observed that the information retrieved is incorrect. The same instance that was used in analyzing the different product reviews has led to the former mentioned problem.

5. Evaluation of results

The datasets that were used in the feature specific sentiment classification and knowledge-based product recommendations were the collection of electronic device reviews from Amazon. The electronic devices were Iphone 6 s plus, oppo f1 plus and Samsung galaxy j7 prime smartphones. These products were named as P1, P2 and P3, respectively. The selection of reviews was considered in such a way as each review contains the mention of the product features. **Table 1** presents the details of the datasets used for this experiment.

The reviews preprocessing was carried out by eliminating stop words and non-English words. The negation words which were present by the adjective in review sentences were handled with care. For such review sentences, the sentiment orientation of the word was determined by flipping the actual sentiment. The product features and opinions extracted on the considered mobile phone reviews using NLP-based language model and LDA-based language model are collected. PROO ontology is engineered and annotated with the

collected product features and opinions. Only one product type for the rule-based sentiments analysis as the PROO ontology is developed for a class of mobile phones of different manufacturers.

ILP rules are also extracted from PROO ontology. The rule predecessor is learned by forming a conjunction of PROO ontology classes and the relevant properties which relate to these classes. The class instances and the property values are reasoned for extracting the target sentiment class instance which is the rule consequent. The generated rules cover the positive instances of the product feature. The assessment of the generated rules is envisioned with area under receiver operating characteristic curve (AUC).

The AUC is a measure to showcase the reviews covered in either of the two sentiment groups (good/bad) available from the dataset. The parameters of the receiver operating characteristic (ROC) curve are the target class label and the ranking attribute. The target instance considered is good for the sentiment class and the ranking attribute is considered as opinion strength. An accuracy of 86.7% of ROC area coverage is obtained. The k-common features identified after the customer searched for Iphone 6 s plus are tabulated in **Table 2**. The value of k found is 17. The similar products are Oppo f1 plus and Samsung galaxy j7 prime.

k-Common features

Phone
ROM
Battery
Performance
OS
Brand
Network connectivity
Camera
Price
Build quality
Touch
Screen
Battery life
Camera quality
Appearance
Display
RAM

Table 2. List of k-common features.

The algorithm calculates sentiments for all the three cellular products on 17 features. Now the algorithm gets all the taxonomical and nontaxonomical constraints for learning feature sentiments from the ontology in the form of rules. In this work, the height of the PROO ontology is 3 and the depth of the child feature node in the ontology tree for taxonomical sentiments is 2. In order to evaluate the sentiments of the k-common features for recommending products, similarity metrics namely cosine similarity [18] and Better [19] are considered.

The small number for k-common features restricts the ability to compare the products during retrieval. This leads to a problem called ‘sparsity problem’. This problem is common in collaborative filtering systems.

An empirical analysis is carried out to understand the impact of small k values on product recommendations. The scatter plot for the percentage of products with different k values with the searched product is presented in **Figure 3**.

It is observed from the above figure that at k value of 1, the product recommendations are not possible as all the products have same similarity value. It is also observed that a single product is not recommended with the available k features as the products are competing with respect to the sentiments on these k features. From k = 2 through 17, the product recommendations has started.

In order to know the product recommendations for small k values, the cosine similarity values for k values 1, 2, and 3 without and with ontology are tabulated in **Table 3**.

An important observation is made **Table 3**. It is that the cosine similarity values without and with ontology for small values of k are same. The influence of taxonomical and nontaxonomical constraints on the product recommendations is reflected from the k value of 4.

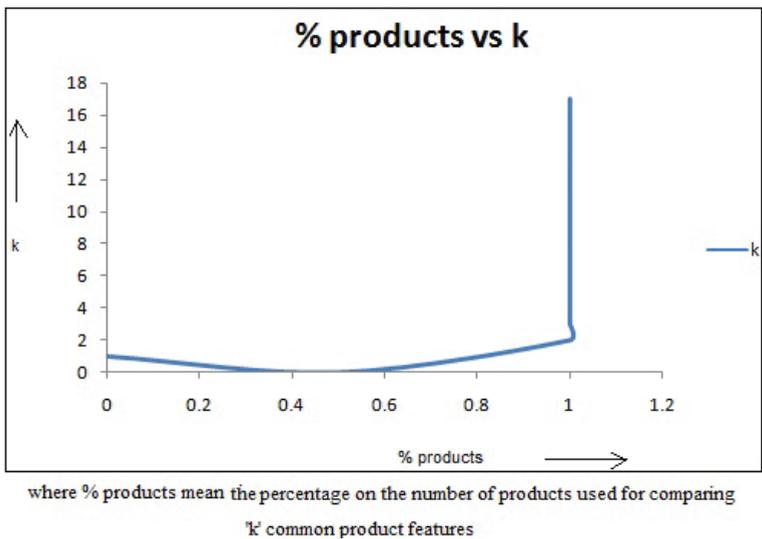


Figure 3. Scatter plot for the percentage of products with different k values.

Without ontology			With ontology	
k	Cosine(P1,P2)	Cosine(P1,P3)	Cosine(P1,P2)	Cosine(P1,P3)
1	1	1	1	1
2	0.86	0.89	0.86	0.89
3	0.69	0.94	0.69	0.94

Table 3. Cosine similarity values for small k.

Different values for ‘k’ provide the useful understanding about the products comparison for eventual recommendations. The variations in the number of k-common features on the similar products using sentiments without ontology and with ontology are tabulated in **Table 4**.

The higher better values in relative comparison with the search product specify that the product is on the top of the recommendation list. The lower cosine values in relative comparison with the search product specify that the product is on the top of the recommendation list. The sentiments of k-common features on the three products in the absence of ontology are displayed in **Figure 4**.

The product similarity with the sentiment data on the similar products without the support of ontology is displayed in **Figure 5**.

The sentiments of k-common features on the three products in the presence of ontology are displayed in **Figure 6**.

The product similarity with the sentiment data on the similar products with the support of ontology is displayed in **Figure 7**.

The product recommendations based on the Cosine similarity measures with and without ontology support for different ‘k’ values are specified in **Table 5**.

From the results in **Table 5**, it is observed that without the support of ontology for different values of ‘k’ (4,8,12) the cosine similarity returned the similar products as recommendations in the same order (product P2 comes first in the list and then the product P3) by using the sentiments on k-features. The product with higher cosine value between two similar products is

k	Without ontology		With ontology					
	Better (P1,P2)	Cosine (P1,P2)	Better (P1,P3)	Cosine (P1,P3)	Better (P1,P2)	Cosine (P1,P2)	Better (P1,P3)	Cosine (P1,P3)
4	-0.0275	0.87	-0.0075	0.79	-0.0275	0.75	-0.0075	0.95
8	-0.0006	0.61	-0.08938	0.45	-0.00063	0.33	-0.08938	0.52
12	0.0370	0.54	0.044583	0.51	0.037083	0.54	0.025874	0.51
17	0.0997	0.29	0.058235	0.48	0.099705	0.29	0.035866	0.49

Table 4. Better and cosine similarity measures statistics for analyzing similarities between products.

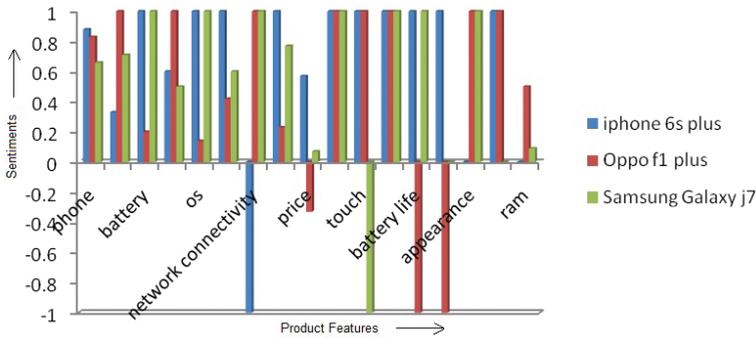


Figure 4. Sentiments of k-common features of similar products in the absence of ontology.

shown as first product in the recommendations list. For k value of 17, the order in the product recommendations is changed. This is because the product P3 has higher cosine value and P2 has lower cosine value when compared with the searched product.

When ontological knowledge is utilized in the product recommendations analysis, the sentiments of the taxonomical features [(battery, battery life) and (camera, camera quality)] are not changed as the sentiments of the parent features are greater than the sentiments of the child features in the taxonomy. The sentiments of the non-taxonomical features [in the work the related features are (RAM, mobile performance), (brand, price), and (screen, display)] are improved in the similar products of k-common features by using the recommendation algorithm. It is observed that the order of product recommendations after improving the sentiments of the related features is changed for two k values (for values 4 and 8). This is because the related sentiments of product P3 have improved so they show higher cosine value than the product P2. This shows the improvement in the product recommendations.

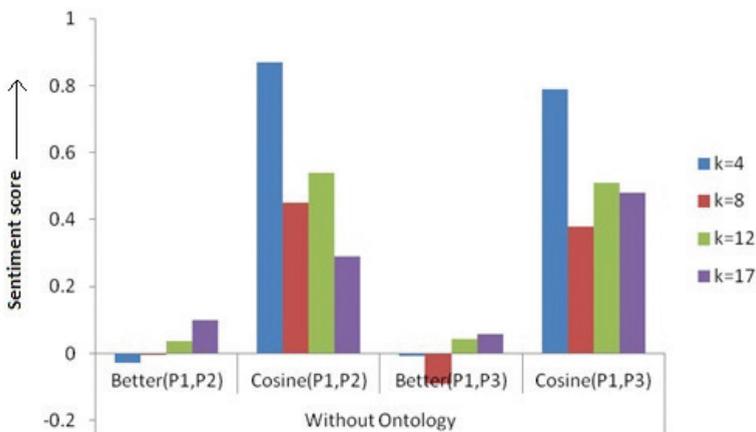


Figure 5. Products comparison with the searched product in the absence of ontology.

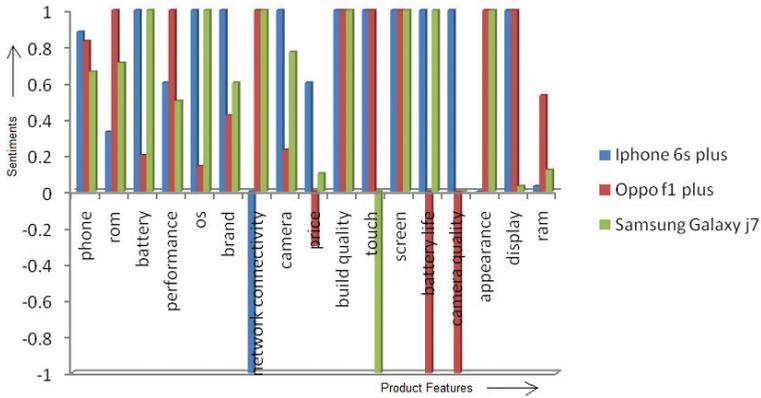


Figure 6. Sentiments of k-common features of similar products in the presence of ontology.

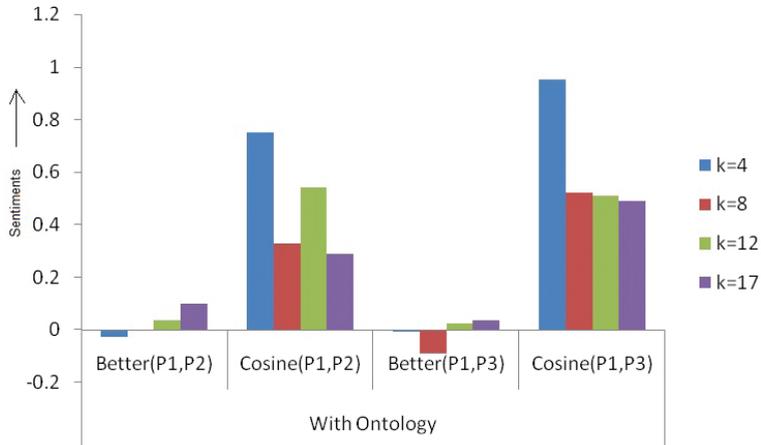


Figure 7. Products comparison with the searched product in the presence of ontology.

Searched product in E-Commerce site: Iphone 6 s plus		
k (No. of common product features)	Product recommendations order–without ontology (product1, product2)	Product recommendations order–with ontology (product1, product2)
4	Oppo f1 plus, Samsung Galaxy j7	Samsung Galaxy j7, Oppo f1 plus
8	Oppo f1 plus, Samsung Galaxy j7	Samsung Galaxy j7, Oppo f1 plus
12	Oppo f1 plus, Samsung Galaxy j7	Oppo f1 plus, Samsung Galaxy j7
17	Samsung Galaxy j7, Oppo f1 plus	Samsung Galaxy j7, Oppo f1 plus

Table 5. Product recommendations.

6. Conclusions and future work

The sentiment-based semantic rule learning for improved product recommendations is presented. The role of semantic rules in sentiment learning is discussed. The influence of sentiments on semantic rules is also discussed. The algorithms for learning taxonomical and non-taxonomical constraints are explained and results are tabulated. Also the algorithm for improving product sentiments using the learned taxonomical and nontaxonomical constraints for product recommendations is explained and results are tabulated. The design decisions in the implementation of PROO ontology are discussed. Several observations from the experiment are also discussed.

Future scope of work is in the lines of learning the intentions of the reviewers using the advanced machine learning algorithms and bigger datasets. The influence of the intentions on new customers and on the product manufacturers by quantifying the effect of intention on information diffusion in social media are to be investigated. The classification performance of the machine learning model on the intentions is to be examined for discovering the actual intention of the reviewer.

Author details

Dandibhotla Teja Santosh^{1*} and Bulusu Vishnu Vardhan²

*Address all correspondence to: tejasantoshd@gmail.com

1 School of Technology, GITAM, Rudraram, Patancheru, Sangareddy, Hyderabad Campus, India

2 JNTUHCEM, Manthani, India

References

- [1] Horrocks I. 2009. Ontologies and the Semantic Web. ACM. 2009
- [2] Liu B. Sentiment analysis and subjectivity. Handbook of Natural Language Processing. 2010;2:627-666
- [3] Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of CSCW '94, Chapel Hill, NC. 1994
- [4] Stavrianou A, Brun C. Expert recommendations based on opinion mining of user-generated product reviews. Computational Intelligence. 2015;31(1):165-183
- [5] Lang K. Newsweeder: Learning to filter netnews. Proceedings of the 12th International Conference on Machine Learning; 1995

- [6] Zhou J, T Luo, F Cheng. Modeling learners and contents in academic-oriented recommendation framework. Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on. IEEE, 2011
- [7] Kolodner J. Case-Based Reasoning. Morgan Kaufmann; 2014
- [8] Burke RD, Hammond KJ, Yound BC. The FindMe approach to assisted browsing. IEEE Expert. 1997;**12**(4):32-40
- [9] Holland S, Ester M, Kießling W. Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications. European Conference on Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer; 2003
- [10] Chen G, Chen L. Recommendation based on contextual opinions. International Conference on User Modeling, Adaptation, and Personalization. Springer International Publishing; 2014
- [11] Gurini DF et al. Analysis of sentiment communities in online networks. Proceedings of the International Workshop on Social Personalization & Search, co-located at SIGIR; 2015
- [12] Li X, Wang H, Yan X. Accurate recommendation based on opinion mining. Genetic and Evolutionary Computing. Springer International Publishing; 2015. 399-408
- [13] Dong R et al. Combining similarity and sentiment in opinion mining for product recommendation. Journal of Intelligent Information Systems. 2016;**46**(2):285-312
- [14] Uzun A, Christian R. Competence Center FAME. Exploiting Ontologies for better Recommendations. GI Jahrestagung. 2010;(1)
- [15] Farsani HK, Nematbakhsh MA. Designing a catalog management system—An ontology approach. Malaysian Journal of Computer Science. 2007;**20**(1):35
- [16] Agarwal B et al. Sentiment analysis using common-sense and context information. Computational Intelligence and Neuroscience 2015;**2015**:30
- [17] Teja SD, Vardhan BV. PROO ontology development for learning feature specific sentiment relationship rules on reviews categorisation: A semantic data mining approach. International Journal of Metadata, Semantics and Ontologies. 2016;**11**(1):29-38
- [18] Huang A. Similarity measures for text document clustering. Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand; 2008
- [19] Dong R et al. Opinionated product recommendation. International Conference on Case-Based Reasoning. Berlin Heidelberg: Springer; 2013

