

Article

OpinionML—Opinion Markup Language for Sentiment Representation

Malik Muhammad Saad Missen ¹, Mickaël Coustaty ², Gyu Sang Choi ^{3,*},
Fahd Saleh Alotaibi ⁴, Nadeem Akhtar ¹, Muhammad Zeeshan Jhandir ¹,
V. B. Surya Prasath ⁵, Nadeem Salamat ⁶ and Mujtaba Husnain ¹

¹ Department of Computer Science & IT, The Islamia University, Bahawalpur 63100, Pakistan; saad.missen@iub.edu.pk (M.M.S.M.); nadeem.akhtar@iub.edu.pk (N.A.); zeeshan.jhandir@iub.edu.pk (M.Z.J.); mujtaba.husnain@iub.edu.pk (M.H.)

² L3i Lab, Université de La Rochelle Av. Michel Crépeau, 17000 La Rochelle, France; mickael.coustaty@univ-lr.fr

³ Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 712-749, Korea

⁴ Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21577, Saudi Arabia; fsalotaibi@kau.edu.sa

⁵ Department of Pediatrics Cincinnati, Children's Hospital Medical Center, Cincinnati, OH 45229 USA; prasatsa@uc.edu

⁶ Department of Mathematics, Khawaja Fareed University of Engineering and Technology, Rahim Yar Khan 64200, Pakistan; nadeem.salamat@kfueit.edu.pk

* Correspondence: castchoi@ynu.ac.kr

Received: 14 March 2019; Accepted: 11 April 2019; Published: 15 April 2019



Abstract: It is the age of the social web, where people express themselves by giving their opinions about various issues, from their personal life to the world's political issues. This process generates a lot of opinion data on the web that can be processed for valuable information, and therefore, semantic annotation of opinions becomes an important task. Unfortunately, existing opinion annotation schemes have failed to satisfy annotation challenges and cannot even adhere to the basic definition of opinion. Opinion holders, topical features and temporal expressions are major components of an opinion that remain ignored in existing annotation schemes. In this work, we propose OpinionML, a new Markup Language, that aims to compensate for the issues that existing typical opinion markup languages fail to resolve. We present a detailed discussion about existing annotation schemes and their associated problems. We argue that OpinionML is more robust, flexible and easier for annotating opinion data. Its modular approach while implementing a logical model provides us with a flexible and easier model of annotation. OpinionML can be considered a step towards "information symmetry". It is an effort for consistent sentiment annotations across the research community. We perform experiments to prove robustness of the proposed OpinionML and the results demonstrate its capability of retrieving significant components of opinion segments. We also propose OpinionML ontology in an effort to make OpinionML more inter-operable. The ontology proposed is more complete than existing opinion ontologies like Marl and Onyx. A comprehensive comparison of the proposed ontology with existing sentiment ontologies Marl and Onyx proves its worth.

Keywords: sentiment analysis; opinion mining; opinion representation; feature selection; application interoperability

1. Introduction

Opinions play a significant role in our daily lives. These opinions help a person to evaluate a situation from various viewpoints in order to make an appropriate and suitable conclusion. Furthermore, the opinions of one individual may effect the opinion of other individuals and hence the concept of *public opinion* is generated. It is observed that public opinion is given equal importance in almost every domain, especially while designing administrative policies and decisions in democratic societies and government offices. Similarly, the manufacturers of commercial products concentrate on the opinions of the general public in order to make rules to enhance their productivity. However, the clients additionally ask about the items that the others have effectively used to enable them to choose which item to purchase. Another example can be an individual who is intending to spend his holidays at a touristic place after observing others opinions about choosing a decent spot to visit.

The world wide web (WWW) has made things easier for people looking for others' opinions. Especially Online Social Networking Services have transformed into huge warehouses of opinions because they have attracted millions of users since their emergence on the web [1]. The Pew Internet and American Life Project surveyed in 2013 and found that almost 81% of the U.S. adult population had access to the internet, and of this ratio, 73% were using social media. Among these social media, the most popular were Facebook (www.facebook.com), Twitter (www.twitter.com), myspace (www.myspace.com), and blogger [2] (www.blogger.com).

With this huge collection of opinion data online, an automatic method of opinion extraction is needed. In the literature, this automatic extraction of opinions is known as opinion mining. A formal definition is that *Opinion mining extracts and analyzes peoples' opinion about an entity* [3]. Furthermore, it is also known as *opinion finding*, *opinion extraction* or *opinion detection* in the related literature. Besides these, terms like *sentiment* and *emotion* have also been used interchangeably for opinions but more in the context of indicating the polarities of the opinions (i.e., negative, positive or neutral). *Polarity Detection* or *Sentiment Detection* or *Sentiment Classification* or *Sentiment Analysis* is one of the several tasks of opinion mining.

Motivation and Contribution

Semantic annotation is the process of labeling the important terms with some tags in a document to explain their semantic orientation in text fragments. This process makes the documents processable not only by humans but also by automated agents [4]. A complete semantic annotation will enable computers to process all or most of the real world situations [5] by annotating all such situations. Existing opinion annotations schemes (i.e., OpinionMining-ML, EmotionML and SentiML) fail to deal with many situations which, if annotated well, could be influential for developing better opinion mining systems. Problems like contextual ambiguities [6,7], lack of semantics interpretation on sentence level, tackling temporal expressions [8,9], identification of opinion holders [10–12], opinion aggregation and their comparison [13,14] remain unanswered by these annotations. Each of the opinion annotation schemes have positive and negative features associated with them but there is a need to have a strong opinion annotation which combines positive features of existing schemes (like flexible emotion vocabulary choice in EmotionML, feature-level processing of OpinionMining-ML, etc.) and focuses on dealing with problems of opinion mining. Our contributions for this paper are listed below:

- We propose OpinionML—a new sentiment annotation markup language,
- We also propose OpinionML ontology supporting OpinionML markup structure,
- We provide a comparison of existing sentiment annotation markup languages,
- We present state-of-art comparison of proposed OpinionML ontology with existing sentiment ontologies,
- We perform a series of experiments in order to highlight the efficacy of proposed OpinionML markup language.

The proposed OpinionML aims to satisfy the objectives described above and can be considered an extension to our previous work [15]. OpinionML utilizes the positives of existing schemes while equipping itself with ontological and linked-data support. The aspects that have been top priority while proposing OpinionML are:

- It provides support for major challenges of opinion mining i.e., contextual ambiguities, temporal expressions, opinion holders, etc.,
- It is easy to understand and modify,
- It has a strong ontological support,
- It provides support to external resources like linked-data and knowledge-bases.

OpinionML makes it possible by annotating each and every necessary component of the opinion model [16], by adopting a modular structure, by providing semantics to the annotations with a proposed opinion ontology and by providing strong support for external semantic web resources. In addition to the OpinionML, we propose OpinionML ontology with built-in support for OpinionML.

We organize the rest of the paper as follows. Section 2 starts with illustrating why opinion mining is difficult, and compares opinion annotation schemes of the past, namely the SentiML, OpinionMining-ML, and EmotionML. Section 3 details our OpinionML in terms of conceptual, logical, and ontological aspects. We also differentiate OpinionML with Marl and Onyx. Finally, Section 5 concludes the paper.

2. Opinion Mining and Previous Annotation Schemes

Opinion mining is a more complex task than the ad-hoc information retrieval. In the following sub-section, we will look at a few examples highlighting how opinion mining is a more complicated task.

2.1. Why Opinion Mining is Difficult

For most of us, identifying semantic orientation of a text (i.e., to judge whether it is positive or negative) could be very simple and could be done just by counting positive or negative words in the text. Consider the following text segment www.religiondispatches.org/let-us-now-praise-wealthy-men-structural-poverty-religiously-reconsidered:

While most of my friends agree that big money needs to be kept out of politics, far fewer of them realize how the power of money shapes us all the way to the core-shapes the way we think, shapes our innermost hopes and dreams, shapes even our faith. That is where the ultimate power of money rests, and that is where an awakened resistance has yet to develop.

In the given text segment, the author has used many positive words like *hopes, dreams, faith* etc. but a careful analysis would reveal that the overall polarity of the text seems to be negative because the author criticizes his friends for not realizing the actual power of money. Therefore, the notion of counting the number of positive and negative words in a text to conclude its semantic orientation is nullified.

The foremost goal of opinion mining is to distinguish the opinions from the facts, which is itself a difficult task. Sometimes it is really hard to differentiate between a fact and an opinion even for a human being [17,18]. For example, given a statement www.dawn.com/news/1138137/pakistan-india-peace-a-good-idea-that-nobody-wants:

Indians and Pakistanis are brothers who have separated. Let them live in different homes and continue to remain brothers.

The issue that may arise is whether the first sentence (*Indians and Pakistanis are brothers who have separated*) in the given text depicts a fact or an opinion. Its syntactical analysis suggests that it is a fact and probably a computer will label it as a fact, which is not correct because it is more likely to be an opinion with a metaphorical use.

The existence of sarcastic or ironic statements in a text cannot be identified with high accuracy and leads to wrong results [19]. For example, the Figure 1 shows a sarcastic product review on Amazon www.amazon.com. In this product review, the author hardly gave a direct opinion of the product but it is the sarcastic language which is indicating his true thoughts about the product (apparently negative). If a human has to judge then it is bit easier to evaluate the polarity of such reviews but the process becomes more complex when computers are to judge such reviews [20]. Turney and Littman [21] highlight the importance of context when processing sarcastic or ironic text for opinion mining.



Figure 1. Sarcastic Amazon Review of a Movie Disc.

Just like the sarcastic language, the mining of colloquial language and emoticons is also a non-trivial task [22]. Such text cannot be ignored because of its very important role in estimating and contributing polarity to the opinionated text. For example, the following statement was found on an online discussion board, where the author of the question replied to someone who suggested the author search on Google www.google.com for his problem:

Thank you so much, this really helped me ☺

The statement seems to have very positive semantics if we keep ignoring the emoticon given at its end, which in-fact reverses the semantics of the sentence to negative.

Another issue which makes the process of opinion mining more complicated is to deal with the contextual polarities [23]. Contextual polarity (the semantic orientation of an expression within a given context) is different from prior polarity (the semantic orientation of an expression without any context). Some of the factors like modality, word sense, the syntactic role of a word in the sentence, or the perspective of the person who is expressing the sentiment, may effect the contextual polarity. In opinion mining, the context is generally defined by negations, topical and cultural aspects.

The simplest case of contextual ambiguity is caused by the negations that, apparently, can be evaluated just by reversing the polarity of the words associated with them, but things are not as easy as perceived since the negation words such as *not*, *nor* etc., aka syntactic negation are not the only parameter and criterion for calculating the negative orientation of a sentence. There are also some other linguistic patterns such as—prefixes (e.g., *un-*, *dis-*, etc.) or suffixes (e.g., *-less*) that change the context to negation in textual data [24]. Similarly, the word intensifiers such as adverbs, adjectives or adverbial phrases; and diminishers (contextual valence shifter) also change the polarity of sentiments [25]. The second case of contextual ambiguities arises when the same word gives different semantics for different domains. For example, the word *unpredictable* could be good as far as plot of a movie is concerned in a movie review, while the same word would be considered as negative if it is used for usability of a product or talent of someone [26,27].

Language is not independent of cultural beliefs, it is part of them [28]. The same words in different cultures could be interpreted with different or even opposite semantics and this is what causes yet another case of contextual ambiguity. For example, the verb *to table* means *to propose* in the UK, but *to set aside* in the US <http://english.stackexchange.com/questions/1999/words-with-opposite-meanings-in-different-regions>. This cultural context of a word's semantics is very hard to deal with and hardly any research work has been done in this regard as far as opinion mining is concerned.

Two more technical challenges that the researchers face are the association of opinions to the given topic (for e.g., to features in product review mining) and aggregation of opinions in a given text. A document could contain opinions about many topics and hence find only those opinions which are linked to the topic; what we are interested in is quite a difficult task. Similarly, one can express different opinions about different aspects of a topic and it is hard to have a global idea about the topic [29]. Therefore, it becomes interesting as well as challenging to aggregate all opinions about a particular topic and present one opinion about it.

Besides these, many other issues like the order of opinion expressions (example *The work is hard, but the salary VS The salary is high, but the work is hard*), measuring the intensity of emotions in opinions (like *very happy VS happy*) etc., further complicate the task of opinion mining.

2.2. Previous Opinion Annotation Schemes

In order to evaluate the opinion mining approaches using machine learning models, semantic annotations play an important role in preparing data. Furthermore, this activity also helps in the automatic extraction of opinions. Unfortunately, there have been hardly any serious attempts for the proposal of annotation schema until recently when SentiML [30], OpinionMining-ML [31] and EmotionML [32] were proposed. In this paper, we propose *OpinionML*, an opinion annotation model addressing challenges of the domain of opinion mining. It can be considered as an effort towards a model providing maximum information by dealing with contexts of the opinions and that will make the process of manual annotation easier by providing an opportunity for semi-automatic annotations.

In this section, we discuss about *SentiML*, *EmotionML* and *OpinionMining-ML* and compare them in the context to represent opinions within text documents.

2.2.1. SentiML

SentiML annotation schema [30] is based on the conventional sentiment annotation style in which a reader can easily find a target tag for the opinionated object. Similarly, some other tags such as a modifier tag are also identified quite easily. Furthermore, another tag is also added in the vocabulary called as appraisal type tag that not only combines the other related two tags but also gives the information about the eventual polarity of the combined expression. SentiML follows an Appraisal Framework (AF) that is one of the strong linguistically-grounded theories. AF assists in finding the appraisal type of information like affect, judgments and appreciation, etc. within the modifier tag. This feature improves the efficacy of SentiML. Being very simple in its annotation scheme, SentiML attracts the researchers working in this domain to adapt it without hesitation because they do not need to learn about anything new to adapt this annotation scheme. Still, there are some issues that can be raised about SentiML style.

SentiML Example

In the example annotations of SentiML, also used in our work [15] (please see Listing 1), one sentence is given as below:

"The US State Department on Tuesday (KST) rated the human rights situation in North Korea "poor" in its annual human rights report, casting dark clouds on the already tense relationship between Pyongyang and Washington."

Relevant annotations are given below:

Listing 1: SentiML example

```

<APPRAISALGROUP id="A0" fromID="T0" fromText="situation" toID="M0" toText="poor"
orientation="negative"/>

<APPRAISALGROUP id="A1" fromID="M1" fromText="dark" toID="T1" toText="clouds"
orientation="negative"/>
<APPRAISALGROUP id="A2" fromID="M2" fromText="tense" toID="T2" toText="relationship"
orientation="negative"/>

<MODIFIER id="M0" start="201" end="205" text="poor" attitude="appreciation"
orientation="negative" force="normal" polarity="unmarked"/>

<MODIFIER id="M1" start="250" end="254" text="dark" attitude="appreciation"
orientation="negative" force="normal" polarity="unmarked"/>

<MODIFIER id="M2" start="277" end="282" text="tense" attitude="appreciation"
orientation="negative" force="normal" polarity="unmarked"/>

<TARGET id="T0" start="175" end="184" text="situation" type="thing"
orientation="neutral"/>

<TARGET id="T1" start="255" end="261" text="clouds" type="thing"
orientation="ambiguous"/>

<TARGET id="T2" start="283" end="295" text="relationship" type="thing"
orientation="neutral"/>

```

There are three different phrases involved in the above-mentioned annotated sentence. Each of the phrases is identified separately by SentiML notation e.g., The terms like *situation*, *clouds* and *relationship* are categorized as targets. The identification of target objects is performed at phrase level. However, in the whole sentence, the main target of the opinion i.e., “North Korea” has not been identified which is one of the main elements of an opinion as defined by Bing Liu [16].

2.2.2. OpinionMining-ML

OpinionMining-ML is an XML-based platform that helps in automatic tagging of attitude expressions associated with features or objects in a given text. According to the literature review of notable work in opinion mining, we can categorize the set of approaches that are used in the extraction of the expressions having feature-based opinions but this categorization is limited to proposing an annotation schema. Our objective is to mine the precise features from the text and then build an ontology of these features. Some user-defined tags like <FEATURE-OF> and <SERVEDAT> are defined in order to build relations among these extracted features. Furthermore, some meta-tags like <APPRAISAL>, <OBSERVATIONS>, etc. are also used to define these features. This tagging will assist in developing a modular approach in order to build an annotation scheme. However, it could be a rigorous activity for an annotator to add too much meta-information about an expressive statement (ES). Furthermore, the overall structure proposed for this scheme seems difficult to understand.

OpinionMining-ML Example

We present the annotation of the previous example [15] using OpinionMining-ML syntax (please see Listing 2):

Listing 2: OpinionMining-ML example

```

<COMMENT id="1" ontologyreference="1">
<FRAGMENT id="1"> The U.S. State Department on Tuesday (KST) rated the human rights
  situation in North Korea "poor" </FRAGMENT>
<FRAGMENT id="2"> in its annual human rights report, casting dark clouds on the
  already tense relationship between Pyongyang and Washington.</FRAGMENT>

<APPRAISAL polarity="negative" intensity="medium">
<FACETREFERENCE>1</FACETREFERENCE>
<FRAGMENTREFERENCE>1</FRAGMENTREFERENCE>
<FRAGMENTREFERENCE>2</FRAGMENTREFERENCE>
</APPRAISAL>

<APPRAISAL polarity="negative" intensity="medium">
<FACETREFERENCE>2</FACETREFERENCE>
<FRAGMENTREFERENCE>1</FRAGMENTREFERENCE>
<FRAGMENTREFERENCE>2</FRAGMENTREFERENCE>
</APPRAISAL>

</COMMENT>
\label{lst:omml}

```

Note: The annotation using OpinionMining-ML requires a domain ontology for meta-tagging. This is why meta-tags like <ONTOFACET> and <FACET> have not been defined here in this example.

2.2.3. EmotionML

EmotionML [32] helps in developing concepts from existing frameworks relating to emotions. It is noteworthy that being informed by the affective sciences [33], EmotionML admits the fact that there exists neither a single standard representation of affective states, nor of the thesauruses capable of defining these terms to use. Therefore, four different types of assertions can be used to describe an emotional state (*emotion*) namely (*category*); (*dimension*); (*appraisal*) and (*actiontendency*). Furthermore, these assertions can be used in identification of the vocabulary.

EmotionML is intended to work at three different use cases [34]: 1. Manual annotation of emotion-related data; 2. Recognition of emotion in real time and 3. The evaluation of emotional system behavior. Furthermore, EmotionML is conceived as a ‘plug-in’ language in order make it suitable for all three domains that may be used in different contexts. As indicated by the W3C draft of EmotionML, one must mention at least one vocabulary to be utilized for speaking to affective and sentiment states. Due to discrepancies among the community, it is unfortunate that EmotionML specifications are not developed efficiently and as a result it failed to include a default vocabulary set. Therefore, it was left to the discretion of users. “It is impossible to impose one single emotion vocabulary,” Marc Schroeder, editor of the EmotionML standard said. “EmotionML is striking a balance, providing a carefully selected set of ‘recommended’ vocabularies” and documenting how to use them.

Things get more critical, too. Besides considering emotional states, “emotion dimension sets” are considered that qualify emotions with attributes such as intensity regardless of its appearance in the text as positive or negative.

EmotionML Example

We present the annotation previous example using EmotionML syntax (please see Listing 3):

Listing 3: EmotionML example

```

<emotionml xmlns="http://www.w3.org/2009/10/emotionml" xmlns:meta="http://www.example.com/metadata" category-set="http://www.w3.org/TR/emotion-voc/xml#everyday-categories">
<info>
<meta:doc>Example taken from annotation of SentiML
</meta:doc>
</info>
The U.S. State Department on Tuesday (KST)
<emotion>
< category name="" value=""/>
rated the human rights situation in North Korea "poor"
</emotion>
in its annual human rights report
<emotion>
< category name="" value=""/>
casting dark clouds
</emotion>
on the already
<emotion>
< category name="" value=""/>
tense relationship between Pyongyang and Washington.
</emotion>
</emotionML>

```

We next give comparison of annotations schemes from different perspectives:

- **Scope:** EmotionML is a markup language and also a W3C standard that covers the entire range of emotion features in all concerning fields. On the other hand, OpinionMining-ML and SentiML are limited to the domains of IR and NLP. While talking about SentiML and OpinionMining-ML, SentiML has a larger scope than OpinionMining-ML that is limited only to feature-based sentiment analysis.
- **Complexity:** EmotionML is considered multifaceted and less user-friendly because of its larger scope than the rest of the two annotation schemes. While the vocabulary of SentiML annotation scheme is easier to use than that of OpinionMining-ML.
- **Vocabulary:** The annotations in EmotionML, in general, are extended to include new and broader vocabulary. On the other hand, SentiML and OpinionMining-ML languages are more specific. The role of SentiML is limited around collecting the concept of modifier and targets of the sentiments while OpinionMining-ML is equipped with meta-tags and is mainly concerned with extraction of sentiment relevant features of the objects.
- **Structure:** The structure of all three annotation schemes is based on XML. However, in OpinionMining-ML, the granularities are well defined at feature level.
- **Contextual Ambiguities:** Like the other two schemas, SentiML also define semantics of the *affective* expressions like appreciation, suggestion, etc. Furthermore, SentiML also assists in identifying the contextual ambiguities, which is considered as a major research issue nowadays in the field of opinion mining.
- **Completeness:** Completeness [5] is one of the main characteristic features of the annotation. This property deals with the fact of whether an annotation covers all or most of the real-world scenarios. For SentiML, EmotionML and OpinionMining-ML, in light of this particular characteristic, it is observed that none of these annotation schemes seem to satisfy it. SentiML and EmotionML both apply annotations to sentimental expressions while OpinionMining-ML targets the corresponding features of objects. Unfortunately it failed to focus on contextual aspects of opinions that are also considered as challenging issues in opinion mining.
- **Flexibility:** EmotionML provides an independent way of selecting the vocabulary of emotional states that suits a particular domain. OpinionMining-ML solely depends on the procedure

of EmotionML and opens ways to build a domain-specific ontology as per requirements, while SentiML lacks such flexibility.

From the above comparison (also summarized in Table 1), it can be deduced that our proposed SentiML bears a larger scope. Besides, it is also furnished with a more affordable and precise thesaurus as compared to previous work of the same kind [35–37]. It is the only annotation scheme dealing with the contextual ambiguities challenge within the scope of its structure, and use of a standard Appraisal Framework makes it strongly theoretically grounded. In addition, it provides an easier environment for annotation of opinions within text documents. Further, we see the opinion annotation process as a set of many tasks such as entity extraction (holder and target entities), subjective word extractions (where positive and negative words are identified), temporal expressions extraction, topic modeling and feature extraction [38]. Hence, we model our proposed annotation schema the same way. Representing all extracted information as sets and linking them together to find answers to required information is the basic idea of the proposed model.

Table 1. Comparison between Existing Sentiment Annotations.

Criteria	SentiML	OpinionMining-ML	EmotionML
Scope	Limited to the domains of IR and NLP	Limited to the domains of IR and NLP	Covers wide range of emotion features
Complexity	Simpler and more semantic	Complex and less user-understandable syntax	Multifaceted and less user-friendly
Vocabulary	Limited around the concepts of modifier and targets of the sentiments	Equipped with meta-tags and feature-based sentiment extraction vocabulary.	Rich in vocabulary which is further extendable
Structure	XML-based structure	XML-based structure	XML-based structure
Contextual Ambiguities	Offers support	No Support	No Support
Completeness	No	No	No
Flexibility	Lacks flexibility	Flexibility depends on ontology used	Flexible in defining emotion states

3. OpinionML—A Conceptual, Logical, Ontology Model

3.1. Conceptual Model

A classic work is reported in [16] in which Liu proposed a comprehensive and precise definition of sentiment analysis task stating that an opinion is described as a quintuple, $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$, where e_i is the real name of an entity, a_{ij} is one of the aspects of the entity e_i , oo_{ijkl} represents the semantic orientation of the opinion about given aspect a_{ij} of entity e_i , h_k depicts the opinion holder, and t_l indicates the time when the opinion is expressed by h_k . The semantic orientation, oo_{ijkl} , about the opinion may be either positive, negative or neutral; or may be expressed using different strength/intensity levels. When an opinion is depicting the entity itself as a major unit, we denote it by using the special aspect “GENERAL”.

We believe that we can adapt this opinion quintuple model to deal with many other complex problems while keeping the opinion model simple. To deal with further opinion problems, we propose to expand this opinion definition both horizontally and vertically (i.e., increasing the number of elements and by adding more details for each element). Besides this, many problems can be dealt with by modeling some operations on this opinion definition. In this section, we extend the opinion abstraction proposed by Liu [16] to enable it to entertain more challenging problems of opinion

mining as mentioned in the previous section. OpinionML mainly deals with problems like contextual ambiguities, opinion aggregation and processing of cultural or informal language within opinions.

Liu's abstraction of the opinion [16] is one of the most prominent works in this field but this abstraction does not deal with some major problems of the domain like contextual ambiguities. In addition, it does not formalize the aggregation of opinions on feature level. In this paper, we propose OpinionML that is capable of dealing with the contextual ambiguities created by topics and cultural aspects. Furthermore, Bing Liu does not provide any logical model for this abstraction while OpinionML comes with a very strong and flexible logical model. As described above, we see the opinion mining process as a collection of several sub-tasks which includes the extraction of entities, subjective words, cultural phrases, temporal expressions and relations between all of them. All of these sub-tasks provide us a collection of sets of different entities and concepts defined as below:

Definition (Opinion Elements):

The opinion recognition system is based on the following elements:

- $F = \{f_1, f_2, \dots, f_{N_F}\}$ is a set of N_F fragments (sentences or sub-sentences) of the given text.
- $H = \{h_1, h_2, \dots, h_{N_H}\}$ is a set of N_H unique holders (i.e., the entity giving opinion about some target) identified in the given text.
 $h \in H$ can be in different fragments.
- $S = \{s_1, s_2, \dots, s_{N_S}\}$ is a set of N_S unique targets (i.e., the entity about which opinion has been expressed in the text) identified in the given text.
where $s_i \in S$ is defined by a set of features X_i .
- $X_i = \{x_{i1}, x_{i2}, \dots, x_{iN_{X_i}}\}$ is a set of N_{X_i} different attributes or features or properties (for example, screen size of a smart phone) about the target s_i

$$X = \bigcup_i X_i \quad (1)$$

where X represents all features in the text.

- $M = \{m_1, m_2, \dots, m_{N_M}\}$ is a set of N_M different modifiers (i.e., subjective words like excellent, destructive, etc.).
 $m \in M$ can be in different fragments.
- $V = \{v_1, v_2, \dots, v_{N_V}\}$ is a set of N_V cultural expressions or informal phrases (for example, "bite the bullet, piece of cake" etc.) as found in the given text.
 $v_i \in V$ can be in different fragments.
- $T = \{t_1, t_2, \dots, t_{N_T}\}$ is a set of N_T different temporal expressions (for example, "day after tomorrow") found in the text
- $Y = \{y_1, y_2, \dots, y_{N_Y}\}$ is a set of N_Y different topics in the given text (for example, "human rights, sports" etc.).
 $y \in Y$ can be in different fragments.
- $R = \{r_1, r_2, \dots, r_{N_R}\}$ is a set of N_R references identified in the text (e.g., source of the text, creator).

OpinionML models add two dimensions to the semantic orientation of opinions. One is to simply compute the polarity of the given text as positive, negative or neutral and the other is to express the emotion attached (if any) to the given text. This emotion modeling of opinion is done according to emotion models proposed recently [32,39].

Definition (Opinion Context):

We define opinion context (\mathbf{c}) as a representation of all elements necessary to represent an opinion. It is defined by

$$\mathbf{c} = (f, h, s, m, y, x, te, t) \quad (2)$$

where $f \in F, h \in H, s \in S, m \in M, y \in Y, x \in X, te \in T$ and t is the opinion publication date.

Definition (Context Polarity function):

Let l is a polarity name (positive, negative and neutral), v the polarity value and w is a confidence method ($w \in [0, 1]$).

We define the polarity function of the context \mathbf{c} by:

$$(l, v, w) = \phi(\mathbf{c}) \quad (3)$$

Definition (Context Emotion function):

Let $L = \{l_1, l_2, \dots, l_{N_L}\}$ is a set of N_L different emotion names (e.g., category, dimension, appraisal, action-tendency). We define the emotion function of the context \mathbf{c} by:

$$(\mathbf{l}, \mathbf{v}, \mathbf{w}) = \Phi(\mathbf{c}) \quad (4)$$

This function can return different emotions of the used emotion vocabulary:

$$|\mathbf{l}| = |\mathbf{v}| = |\mathbf{w}| \leq |L| = N_L \quad (5)$$

and (l_i, v_i, w_i) represent respectively the emotion name, the intensity value and the confidence method for the same emotion.

Definition (Opinion Entity):

Let $\phi(\mathbf{c})$ and $\Phi(\mathbf{c})$ respectively the polarity function and emotion function of the context \mathbf{c} . The opinion o is defined by

$$o = (\mathbf{c}, \phi(\mathbf{c}), \Phi(\mathbf{c})) \quad (6)$$

Definition (Opinion Connection function):

Sometimes, more than one opinion exists in a single textual unit joined by some connections like (and, or, yet, so, etc.). In such cases it becomes mandatory to detect the global polarity of this textual unit. Therefore, we define an opinion connection operation which takes two opinions joined by one of the connections defined in a separate list and computes a global polarity for the textual unit.

Let Z is a set of the connection operators (“for”, “and”, “nor”, “but”, “or”, “yet”, “so”, ...)

Let $O = \{o_1, o_2, \dots, o_{N_O}\}$ is a set of N_O different opinions.

$$\mathcal{C}(o_i, o_j, z) = \begin{cases} (l, v, w) = \phi(o_i, o_j, z) & (\text{polarity}) \\ (\mathbf{l}, \mathbf{v}, \mathbf{w}) = \Phi(o_i, o_j, z) & (\text{emotion}) \end{cases} \quad (7)$$

where $o_i, o_j \in O, z \in Z, l \in$ (positive, negative and neutral), $v \in [0, 1]$ the polarity value, $w \in [0, 1]$ is a polarity confidence and the matrix $(\mathbf{l}, \mathbf{v}, \mathbf{w})$ is the different emotions with their values and confidences.

Definition (Opinion Aggregation function):

Opinions about an object can be found in many opinionated fragments and if we want to know an overall opinion about an object, we need to aggregate all the related opinions about that object. Therefore, we define an opinion Aggregation operation which is described below: An aggregate function is notated by $\mathcal{G}_{\mathcal{F}(X_i)}$. The subscript $\mathcal{F}(X_i)$ represents the specific aggregate function \mathcal{F} to be applied, along with the attribute X_i on which to perform the function:

$$\mathcal{G}_{\mathcal{F}(X_i)}(s_i) = \begin{cases} (l, v, w) = \mathcal{P}_{\mathcal{F}(X_i)}(s_i) & (\text{polarity}) \\ (\mathbf{l}, \mathbf{v}, \mathbf{w}) = \mathcal{E}_{\mathcal{F}(X_i)}(s_i) & (\text{emotion}) \end{cases} \quad (8)$$

where $\mathcal{P}_{\mathcal{F}(X_i)}(s_i)$ and $\mathcal{E}_{\mathcal{F}(X_i)}(s_i)$ are respectively the polarity aggregate function and the emotion aggregate function, where $s_i \in S$ represents any target, and of course, X_i is a set of attributes (all or part) about target s_i .

- A frequent aggregate function parameter is the attribute in $s_i \in S$ over which the aggregate function is to be applied. For example, in an opinion containing hotels and their rooms, the global quality of the room would be written as:

$$\mathcal{G}_{\text{Quality}(\text{room})}(\text{Ibis hotel}) \quad (9)$$

- The result of $\mathcal{G}_{\mathcal{F}(X_i)}(s_i)$ is a relation with a single attribute, containing two components:
 - the first component is defined by three tuples: label (neutral, positive, negative) polarity value and polarity confidence.
 - the second component is defined by a matrix of the emotions with their intensities and confidences.

that holds the result of the polarity aggregate function.

General Form of the Aggregation operation

- Typically, groups are specified as a list of attributes $G_{i1}, G_{i2}, \dots, G_{iN_{X_i}}$, meaning that, for a given relation, tuples should first be clustered into partitions such that these tuples share the same values for $G_{i1}, G_{i2}, \dots, G_{iN_{X_i}}$. The aggregate function should then be applied to each cluster:

$$G_{i1}, G_{i2}, \dots, G_{iN_{X_i}} \mathcal{G}_{\mathcal{F}(X_i)}(s_i) \quad (10)$$

- With groups, the resulting aggregate relation would have as many tuples as there were groups in the original, partitioned relation.
- As a final extension, we note that we can perform more than one aggregate function $\mathcal{F}_k(X_{ik})$ over the groups in a relational algebra expression. Thus, the general form of the aggregation operation is:

$$G_{i1}, G_{i2}, \dots, G_{iN_{X_i}} \mathcal{G}_{\mathcal{F}_1(X_{i1}), \mathcal{F}_2(X_{i2}), \dots, \mathcal{F}_m(X_{im})}(s_i) \quad (11)$$

To explain the usage of these functions, we give an example where we want to know a global polarity of an object which has been the target of several opinions within a given text. In this case, one function can represent all the positive opinions about this object times their confidence values and averaged by total number of positive opinions. Similarly, functions could be defined for negative and neutral polarities. The global polarity orientation of the object remains the one with the highest score with confidence value of difference with other values.

3.2. Logical Model

The present section describes the logical model of OpinionML. The syntax of OpinionML along with the thesauruses are described in detail. The thesauruses are designed to describe and differentiate the opinions in terms of categories, contextual dimensions, appraisals and/or action tendencies. This section presents a detailed discussion on a number of relevant aspects of opinion. This section also specifies Opinion Markup Language (OpinionML) 1.0, a markup language designed to be applied as a helpful tool in a number of technological contexts. Furthermore, it also helps in making standards for

description of sentiments in a given opinionated text using a finite set of fixed descriptors. We observed that the required vocabulary is dependent on the context of its usage. The formal and practical way of defining OpinionML is to define the possible structural elements along with their valid child elements and attributes and then allow the users to “plug in” the vocabularies that they consider appropriate for their work. The proposed OpinionML vocabulary is capable to serve as a starting point. If the listed vocabularies seem inappropriate, users can create their custom vocabularies. We have used hyperModel software tool to generate the UML diagrams (see Figures A1 and A2) of OpinionML XML Schema's.

3.3. Structure of OpinionML

The modular structure of OpinionML makes it very flexible to modifications and easier to read and annotate. Listing details can be found in Appendix B.

3.3.1. Profiling Section

It is the heart of the logical model of OpinionML which acts as a database of all elements of OpinionML. Details about each element are added by extracting information from the given text. Only unique entities of each type are kept and information is not repeated on same entities. This information acts as a profile of an entity whether it is of type holder, target or modifier, etc. Each element is given a unique ID which can be referenced from other sections to leverage this element. In short, the profiling section is like a section where variables of a program are defined using a programming language.

3.3.2. Opinion Section

It is more an operational part of OpinionML where all the elements defined in the profiling section are combined together to give semantics to the text being annotated. The <Opinion> elements are defined here by integrating its elements from the profiling section.

3.3.3. Vocabulary Section

The emotional states are needed to annotate text with the opinions. It is proven that different emotional states are needed for different kinds of situations and domains, therefore, their use cannot be forced by defining a fixed set of states. Inspired by the EmotionML strategy, OpinionML provides a flexible use of sets of emotional states depending on the situation. In EmotionML, an emotion is described in terms of descriptors: category, dimension, appraisal and action-tendency that are defined with a name and an optional intensity value (except dimension for which it is mandatory). OpinionML defines all vocabularies to be used for annotation in this section (please see Listing 4).

These vocabularies can be cited in <Opinion> elements (please see Listing 5).

If vocabularies are not part of the OpinionML document as defined in the above listing then they can be referenced in <OpinionML> or <Opinion> element (please see Listing 6).

The elements of OpinionML along with their syntax can be consulted in Appendix A.

3.4. OpinionML Example

The profile section of OpinionML contains details about different elements of OpinionML (extracted implicitly or with the help of annotator) which then can be referenced in annotation of the text. In the following example, we annotate a sentence “The U.S. State Department on Tuesday (KST) rated the human rights situation in North Korea poor in its annual human rights report casting dark clouds on the already tense relationship between Pyongyang and Washington”. This little example is enough to point out some of the major characteristics of OpinionML. The structure of OpinionML is very simple because of its modular approach. Anyone who is familiar with the structure of OpinionML will easily understand it. If value of any attribute cannot be qualified because of missing information in the given text, it is left as it is with null "" value.

In this example, we opt to create two fragments of the sentence while keeping the link between both fragments by giving reference of the sentence. This reference is very important when a piece of information is missing in any of the fragments of a sentence. For example, for the second fragment it is hard to point out the holder of the fragment but this value can still be estimated by other fragments of the sentence. Please see Listing 19.

The way opinions are modeled in OpinionML allows us to perform very complicated operations like connection and aggregation of opinions in a very easy and flexible manner.

- **Connection:** A connection function takes two opinions (let say *OP01* and *OP02*) and the connection operator (for example OR, AND, etc) linking those opinion segments as its input. Polarity and emotions associated to these opinion segments can be combined by exploiting their <polarity> and <emotion> elements.
- **Aggregation:** Aggregating opinions is one of the hardest tasks in opinion mining. OpinionML makes it easier for us by giving us an opportunity to aggregate opinions on two levels: document level (i.e., aggregating all opinion elements to have one global polarity or emotion score for the given document) and object level (i.e., aggregating all opinion elements talking about different features of the same object to have one global polarity or emotion score for a given object).

3.5. OpinionML Ontology

We also propose an opinion ontology (called OpinionML ontology). OpinionML Ontology aims to complement the Marl Ontology [40] and Onyx Ontology [41] by providing a simple means to describe opinion analysis processes and results using semantic technologies. Figures 2 and 3 depict the simpler versions of our proposed OpinionML ontology. Figures 4 and 5 shown (Part 1) and (Part 2) Class diagram of OpinionML ontology respectively. A more recent work on sentiment analysis using ontology called OntoSenticNet [42] is also worth noting. However, OntoSenticNet is a commonsense ontology which does not as such relate to sentiment ontology and hence can be ignored for comparison with OpinionML. The goals of the OpinionML ontology to achieve as a data schema are:

- To make it enable in order to publish the raw data about opinionated text in user-generated content,
- To construct a model that helps to compare the opinions automatically, coming from different sources having different polarity, topics, features etc.,
- To link the opinions using contextual information expressed in concepts with other popular specialized ontologies.

The detail about the terms used in OpinionML ontology are given below. These terms are classified by class (concepts) and by property (relationships, attributes).

- Classes: AggregatedOpinion, Opinion, Polarity,
- Properties: aggregatesOpinion, algorithmConfidence, describesFeature, describesObject, describesObjectPart, extractedFrom, hasOpinion, hasPolarity, maxPolarityValue, minPolarityValue, polarityValue,
- Instances: Negative, Neutral, Positive.

The OpinionML ontology class diagram (see Figures 4 and 5) shows connections between classes and properties used for describing opinions. We used protege <http://protege.stanford.edu> for creating ontology and Graffoo [43] for its visualization.

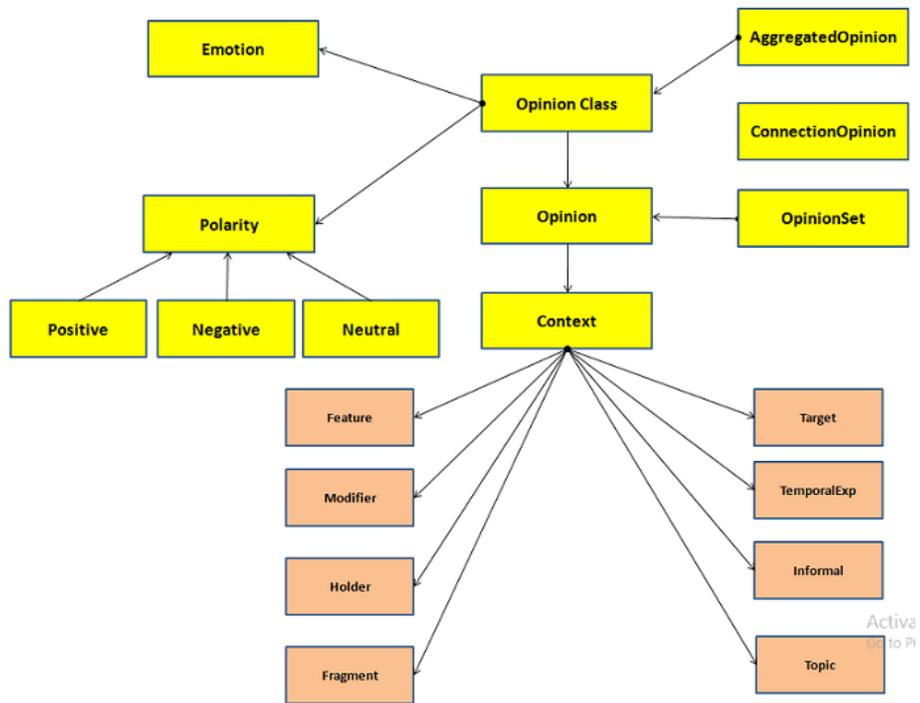


Figure 2. (Part 1) Simpler Version of Class Diagram in Figure 4.

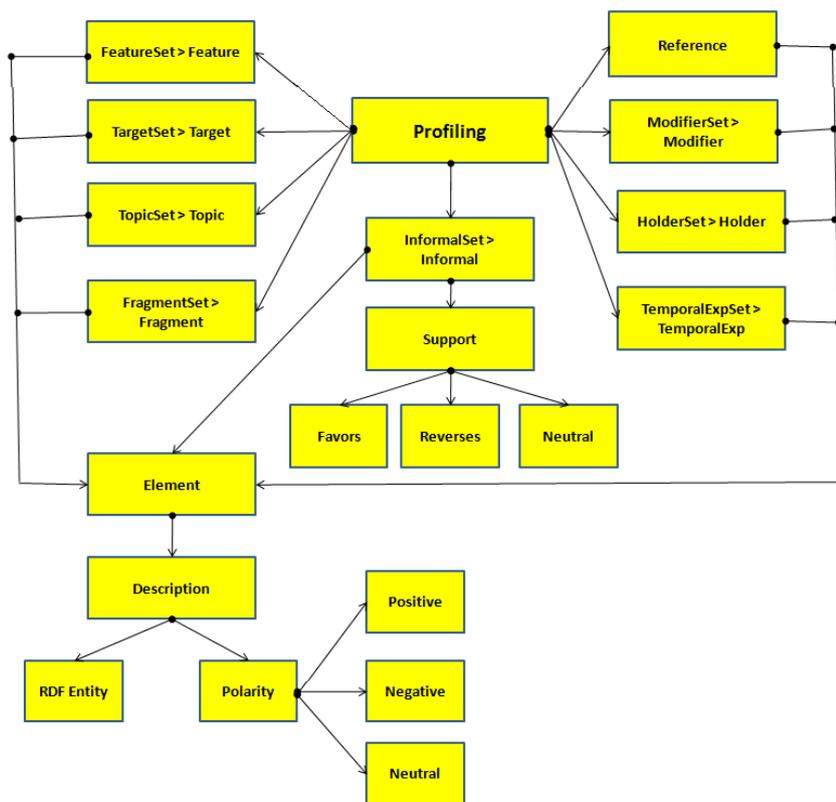


Figure 3. (Part 2) Simpler Version of Class Diagram in Figure 5.

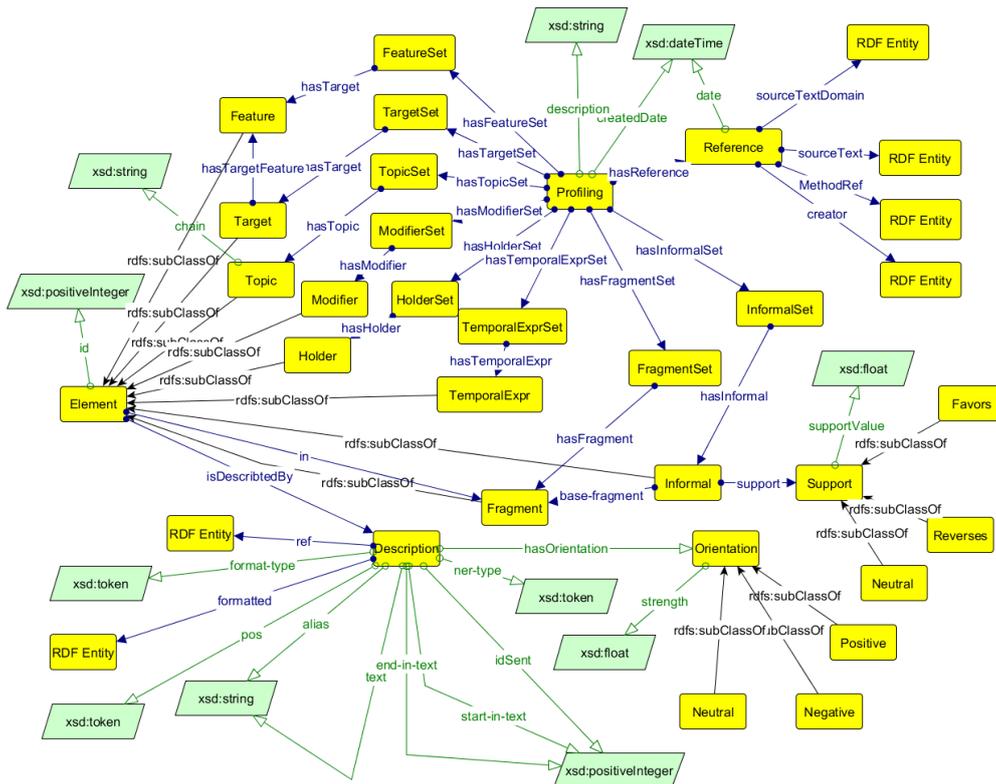


Figure 4. (Part 1) Class diagram of OpinionML ontology.

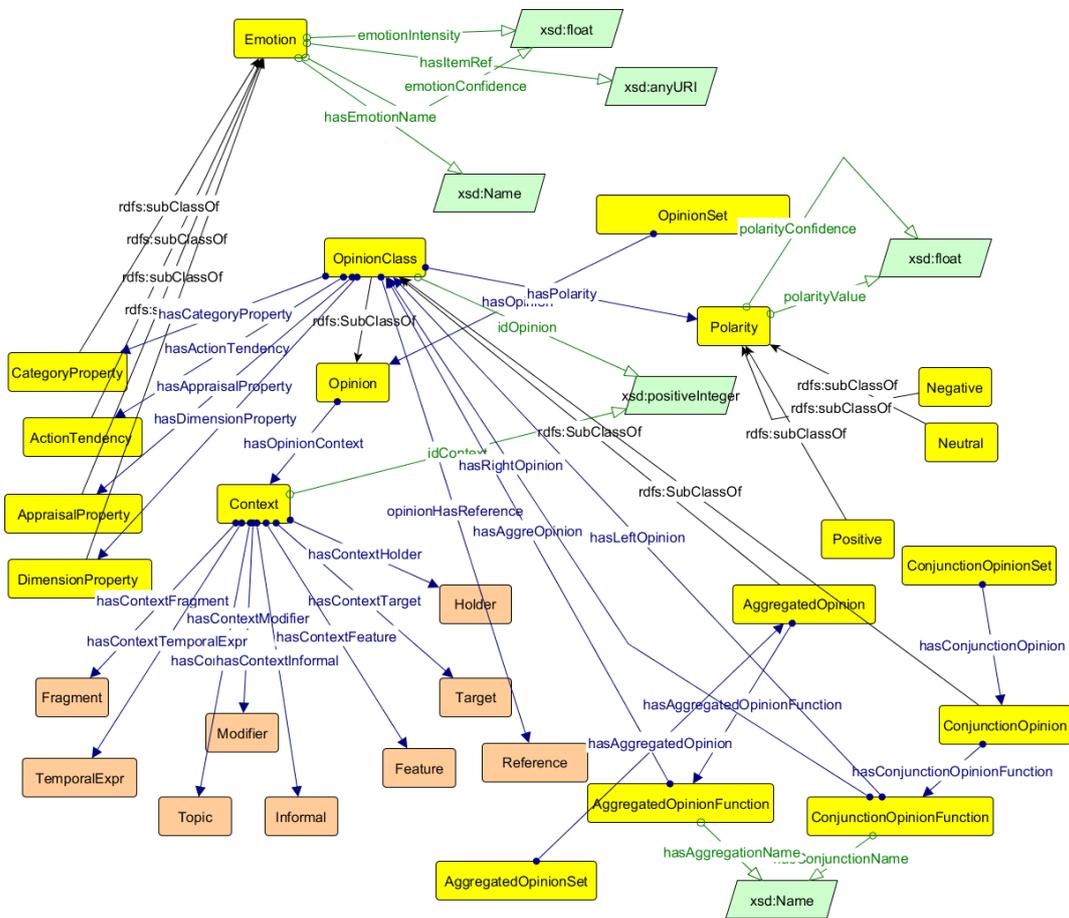


Figure 5. (Part 2) Class diagram of OpinionML ontology.

3.6. Comparison with Marl and Onyx

Marl ontology is a semantic resource that helps in annotating the opinions. Furthermore, this ontology mainly focuses on polarity extraction and is not capable of representing emotions. However, Onyx, on the other hand, provides a number of tools to perform various tasks of sentiment analysis at different granularity levels, including advanced Emotion Analysis. However, both Marl and Onyx fail to represent opinions in a complete manner i.e., both of them lack many concepts (like holder, target, temporal entities, etc.) that are very important for opinion mining. While Onyx goes far ahead to represent emotions but lacks the tools OpinionML ontology is equipped with. The depth of OpinionML ontology can be estimated by looking at the fact that it envelops semantic orientation from word level (i.e., polarity of modifiers) to a textual segment containing opinion (i.e., <opinion>) and then to aggregating them on object and document level. OpinionML combines polarity and emotion components in a coherent way by representing both in a similar fashion. On comparing all three ontologies, it seems that Onyx resembles OpinionML in terms of its dealings with emotions but lacks on complete representation of opinions. We detail the advantages of OpinionML as follows.

- **Linked Data and Knowledge Base Support:**
One of the biggest advantage of using OpinionML is its support for linked data. All the entities and concepts are identified with Universal Resource Identifiers (URIs) that can be used to link to external resources like knowledge base DBpedia, etc. It will help enrich more information about entities where information is lacking. Besides this, it seeks support of a knowledge base which models complicated relation between features and targets including nested relationships. In the annotated example given above, we have two different targets like 'North Korea' and 'Pyongyang' in two fragments. If North Korea is linked to the popular knowledge-base using its URI then we can easily know the relationship between both of these targets where 'Pyongyang' is represented as the largest city (*dbpprop : largestCity*) of North Korea.
- **Standard XML Format:**
The logical model of OpinionML uses XML format for defining its elements. XML format supports interoperability between different platforms and is a tested data exchange format. One of the biggest advantages of using XML is that lot of tools and technologies already exist for XML processing and one does not need to develop new tools.
- **Ontology Support:**
OpinionML takes support of domain ontologies for associating the instances to their relevant concepts. Marl ontology <http://www.gi2mo.org/marl/0.1/ns.html> is one of the ontologies which defines concepts and relations relevant to opinion mining. OpinionML satisfies almost all the concepts of Marl but Marl ontology still lacks many concepts that still need to be added (like time, informal, etc). Therefore, design and creation of another detailed opinion ontology is part of our future work.
- **Ontology-Based Indexation:**
The support for our proposed opinion ontology facilitates the ontology-based indexation of documents which will definitely ensure the efficient processing of OpinionML documents.
- **Domain Independent Support:**
OpinionML is not limited to textual data annotation of a certain domain but it holds the capability of supporting data from all domains i.e., product reviews, news data, tourism, social network data, etc.
- **Granularity Independence:**
Annotation schemes generally restrict their users to limit themselves to annotation of one granularity level like document, sentence, sub-sentences or passages but OpinionML gives liberty to its users to choose a granularity level of their own choice by identifying them as fragments. A fragment is supposed to be a basic semantic unit having a semantic orientation.

- **Semi-Automatic Annotation:**
Entity orientation perspective of the problem where opinion mining is defined in terms of several sub-tasks makes the opinion annotation much easier using OpinionML graphical annotator. Using the semi-automatic approach rather than a 100 percent manual approach makes it possible to annotate big data collections in less time. However, in the presence of an opinion algorithm it will be a 100% automatic annotation.
- **Problem Oriented Annotation:** The annotation approaches can be arranged in a three-dimensional space, c.f. [44,45]:
 1. effort of the annotation,
 2. completeness of the result (i.e., how well does it capture the real-world situation) and
 3. (ontological or social) commitment to the result (i.e., how many commit to this model of the world and understand it).

OpinionML is an effort towards achieving completeness of the results where it models most of the real-world situations. For example, it takes into account the contextual ambiguity problem and the processing of cultural expressions or informal language. It is to be noted that no annotation model has dealt with these problems until now. Most of the existing annotation models limit themselves to annotation of subjective expressions while ignoring the rest of the problems. The logical model of OpinionML also helps to aggregate opinions on feature as well as higher granularity level (i.e., sentence, passage, and document).

- **Adaptability:**
Although OpinionML is flexible enough to be adopted for any domain but of course it supports the addition of extra information in its model as per requirements. This is made possible by the use of references that are part of the model. These references can be used to model different aspects like user profiles, geo-locations or any other object according to the requirements.
- **Problem Oriented Vocabulary:**
One of the advantages of using OpinionML is its vocabulary conformance with the existing work. The usage of terms like holder, target, etc. makes it easier to understand OpinionML.
- **Standard Based Support:**
The structure of OpinionML is inspired from the W3C standard EmotionML and therefore is equipped with the flexibility and interoperability that EmotionML provides. OpinionML uses the notion of flexible vocabularies exactly the way EmotionML does it.
- **Compensating Missing Information:**
OpinionML compensates for the missing information by using the links between its elements. For example, in the given example, holder information for the fragment FRAG01 is missing but it can still be extracted using its sibling fragment i.e., FRAG0 because both of them make part of a single sentence.
- **Redundancy Reduction:**
The logical model of OpinionML helps to reduce redundancy by keeping profiles of opinion elements separately from actual annotation. Hence, annotation only keeps references of those opinion elements. This model facilitates the modifications in annotations and improves the comprehension of the annotations.
- **Modular Approach:**
OpinionML elements are equipped with rich information about the text to be annotated. This information is split into different modules to save readability and comprehension of the annotation. References are created among several elements of the OpinionML to create a link between several modules. The < *Profiling* > module acts like a database of all OpinionML elements where information about all elements is kept while in < *Opinion* >, a collection of several elements are combined to give some semantics of the text. Vocabularies and other

knowledge-based resources are kept as a separate module that provides the flexibility of the OpinionML.

- **Rich in Information:**
The logical model of OpinionML enriches the opinion elements with information that could be very helpful. Ontologies are used for obtaining this information about different elements. The structure of the logical model helps to get a benefit from this information.
- **Representations of opinion:** Opinions may be depicted using the four types of descriptors derived from the scientific literature like *emotions* is used in EmotionML [32]: <category>, <dimension>, <appraisal>, and <actiontendency>.
An <opinion> element may consist of one or more of these descriptors. Each descriptor must be labeled with a name and a value attribute depicting its intensity. For <dimension> descriptor, it is mandatory to assign some points to the value attribute since it describes the opinion strength on one or more scales. For the other descriptors, there is an option to ignore the value since it is possible to make a binary decision about the presence of a given category in the opinionated text. The example below demonstrates various numbers of representing the possible uses of the core opinion representations.

4. OpinionML Evaluation

After a deep analysis of all annotation schemes discussed above, it can be concluded that the proposed OpinionML adheres to SentiML in its structure and objectives while OpinionMining-ML and EmotionML are a little bit different in their nature as far as sentiment annotation is concerned. For example, OpinionMining-ML more focuses on aspect-based opinion mining while EmotionML focus remains mostly on emotion processing. Therefore, it is preferable to compare performance of OpinionML with SentiML, which can also be considered the latest development on sentiment annotation.

4.1. Data Collection

For experimental purpose, we use original data collection developed by SentiML researchers and annotate it with OpinionML. The original SentiML data collection [30] consists of several text documents (with 7000 words) taken from the following online sources:

- **Online Political Speeches:** This part of the data collection consists of text documents describing political speeches of US presidents.
- **TED Talks:** Second source of the data collection is supposed to be Technology, Entertainment, Design (TED) talks on several topics revolving around the theme “ideas worth spreading”.
- **MPQA Human Rights:** Third source for the used data collection is an extraction from popular MPQA opinion data collection [46]. The “human rights” part of the mentioned data collection is used for this purpose. It is to be mentioned that MPQA original data collections consist of texts with opinion and other emotional annotations of opinions.

The preparations of the data collection involve annotating the chosen data collection with OpinionML.

4.2. OpinionML Annotation

For data collection to be annotated using OpinionML, the following operations are to be performed:

- identifying opinionated statements within a document,
- identifying (possible) phrases within a statement,
- identifying opinion holders in the sentence,
- identifying the object about which the opinion was (or target) made in a sentence,

- identifying different aspects of target within extracted phrases,
- extracting sentiment of the target from the opinion expressed in a sentence,
- assigning an appropriate title of the given sentence,
- identifying opinion holder at phrase level,
- identifying the possible expressions at temporal level within sentences.

For annotating data collection with OpinionML, we hire five persons having a computer science background in their major studies and having some experience with data annotations.

4.3. OpinionML Annotation Guidelines

The persons hired for annotations are given a briefing on purpose of annotations along with demonstrations. Each hired individual is given the following set of annotation guidelines, mostly focusing on cases with possible conflicts:

- **Identification of Polarity of a Sentence:** Annotating individuals are supposed to label a sentence as positive (or negative) if it shows a positive (or negative) orientation for a given target. The sentence is labeled as “neutral” if you cannot decide about its polarity.
- **Labeling Opinion Holders or Opinion Targets** Generally, the entity giving opinion (i.e., Holder of the opinion if exists) can be tracked in subject(s) while the entity which is being opinionated (i.e., target of the opinion) can be checked in object of a sentence. If any individual fails or finds difficulty in identification of either or both of holders or targets, he should mark nothing in holder and target elements. The same process can be repeated for phrase level identification of holders and targets.
- **Identification of Sentence Topic** A web directory like DMoz can be consulted for topic identification of a sentence. While using a web directory, one should be careful about the selected topic and must choose the nearest chain.
- **Identify Informal Expressions** An idiomatic resource can be used to identify idioms or informal expressions present in the text. It is a very necessary phase because words present in an idiom do not necessarily represent their actual semantics.

We use Fleiss Kappa [47] measure to assess degree of agreement among annotators. This measure is used when computing agreement between a fixed number of raters while assigning ratings to a number of items or classifying items.

$$\kappa = \frac{P - \bar{P}_e}{1 - \bar{P}_e} \quad (12)$$

The factor $1 - \bar{P}_e$ depicts the attainable degree of annotation performed by OpinionML and, $P - \bar{P}_e$ is the calculation of the degree of actual agreement achieved above chance. In case of ideal agreement i.e., agreement without conflict, then $\kappa = 1$. However, the value of $\kappa \leq 0$ if there is no agreement. The scores for Fleiss kappa agreement in order to perform different tasks are shown in Table 2.

Table 2. Fleiss Kappa Agreement Results for Different Tasks.

Task	Kappa Agreement Score
Sentence Polarity Agreement	0.87
Holder Recognition Agreement	0.93
Target Recognition Agreement	0.94
Topic Identification Agreement	0.79
Informal Expression Agreement	0.81
Overall Annotation Agreement	0.80

4.4. Evaluation

Fifteen queries are generated for comparison of SentiML [15] and OpinionML performance from the original data collection (see Table 3). All generated queries are in the form of questions asked using natural language. Queries are processed and executed against corresponding annotations of SentiML and OpinionML and results are shown in Table 4.

Table 3. Set of Natural Language Queries Generated (also used in our previous related work [15]).

Query I	Who blamed North Korean Authorities?
Query II	What is report about?
Query III	How many times the report said something negative about North Korea?
Query IV	Who runs the greatest anti-poverty program?
Query V	What IFRC says about human rights situation in Iran?
Query VI	Which entity named the US report a “groundless plot”?
Query VII	On what issue president Kim made a very positive statement?
Query VIII	Who are detained arbitrarily?
Query IX	What causes detention of citizens?
Query X	Who said that “Terrorists believe that anything goes in the name of their cause”.
Query XI	In IHRC declaration who was target of injustice?
Query XII	Which organization called on Israel to stop bombing?
Query XIII	Analysts have warned about what?
Query XIV	What is the purpose of Mary Robinson visit to China?
Query XV	What President Obama had to say about violence?

Table 4. Evaluation Results of SentiML and OpinionML with R-Precision.

Query	SentiML	OpinionML
Query I	0	1
Query II	0	1
Query III	0	1
Query IV	1	1
Query V	0	1
Query VI	0	1
Query VII	0	1
Query VIII	1	1
Query IX	0	1
Query X	0	1
Query XI	1	1
Query XII	0	1
Query XIII	1	1
Query XIV	0	1
Query XV	0	1
Mean	0.26	1

We perform experiments on the generated set of queries by choosing a famous measure of evaluation. Selecting a suitable evaluation measure is very important during IR experiments and sometimes it can be very tricky. In cases like ours where system is supposed to bring only one document and it must be relevant, measures like Precision and Recall behaviours may be quite similar. The nature of our problem demands that our proposed system should behave ideally since we have access to only one relevant corpus having the target text segment that must be retrieved. Therefore, we decide to choose R-Precision [48] for experimental purposes. R-Precision is supposed to provide 1 as a result for an ideal annotation and zero for the annotation failing to perform well. The equation for computing R-Precision is given below:

$$R - Precision = \frac{r}{|R|} \quad (13)$$

where r depicts the number of relevant segments retrieved from top $|R|$ documents and $|R|$ is the total number of relevant documents within the corpus. The results (see Table 4) reveal that OpinionML significantly outperforms SentiML (with p -value < 0.05). We can conclude from the results that the proposed approach i.e., OpinionML retrieves answers to natural language queries with a mean R-Precision score of 1.00 in comparison to mean R-precision of 0.26 for SentiML. As discussed already, SentiML inventory is missing some important elements like opinion holders and sentence-based orientations, making it hard to answer such queries. Therefore, It becomes difficult for SentiML to answer these queries without help of NLP techniques while on the other hand, OpinionML's structure helps in providing answers to queries.

4.5. OpinionML Annotation Tool

In the previous sub-section, we described the mathematical model for our proposed opinion annotation model which takes into account various challenges of opinion mining including contextual aspects of opinions. In this section, we highlight the practical view of OpinionML model i.e., how we see the annotation process in the perspectives of OpinionML.

1. Annotator is presented with a graphical user interface where he opens a document to be annotated.
2. Once the document is loaded, all the necessary entities and concepts are marked with different colors automatically.
3. Sets of these concepts and entities are automatically defined that can be browsed by the annotator.
4. In the first step, annotator is asked to ratify all the identified entities and concepts. He can mark or unmark already identified entities and concepts.
5. Once all entities and concepts have been identified semi-automatically, now the annotator is presented with a choice of identifying holders, targets and modifiers of the opinions within each fragment. Generally, they are already marked but a repetition of similar intervention would be required as in the previous step.
6. In this step, the polarity of each fragment is marked.
7. Finally, cultural phrases are identified and presented to the annotator and he is supposed to change the polarity of the fragments accordingly.

The tool automatically links the inter-annotator agreements scores to the algorithms being used for recognition of entities and concepts and polarity of the fragments which is used to improve the accuracy of these algorithms.

5. Conclusions and Future Work

The OpinionML 1.0 focuses on the major issues while performing requirement specifications using use cases. In a future call for implementations, the implement ability of all features provided by the

specification will be verified. Several challenging issues are observed but these issues are much more troublesome in the first version of OpinionML to handle. There is a need for an efficient solution that helps in representing regulation in OpinionML, i.e., the fact that an opinion was suppressed, simulated and masked by another opinion, etc. Another limitation of OpinionML 1.0 is that it does not make use of ontologies for defining the terms in an opinion vocabulary in order to relate the terms to one another, and build an association among the opinion vocabularies where applicable. Another ambiguity lies in describing the specification of scales i.e., should it be discrete, continuous, uni-polar or bipolar, etc.? In the future, a more detailed definition of scales can be introduced by finding a consensus in the opinion community. Once that OpinionML 1.0 has reached its full maturity, these directions can be developed in future versions of OpinionML.

Author Contributions: Conceptualization, M.M.S.M.; Data Curation, G.S.C. and M.H.; Formal analysis, M.S.M.M., N.A. and V.B.S.P.; Funding acquisition, M.Z.J. and G.S.C.; Investigation, N.A. and M.Z.J.; Methodology, M.C. and V.B.S.P.; Resources, M.C. and N.S.; Software, F.S.A.; Supervision, G.S.S. and M.M.S.M.; Validation, G.S.C.; Writing—review and editing, M.H., M.M.S.M. and V.B.S.P.

Funding: This research was supported by MSIT (Ministry of Science & ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2016-0-00313) supervised by the IITP (Institute for Information and Communications Technology Promotion) and 2017 Yeungnam University Research Grant.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Elements of OpinionML

The following describes the syntax of the main elements of OpinionML.

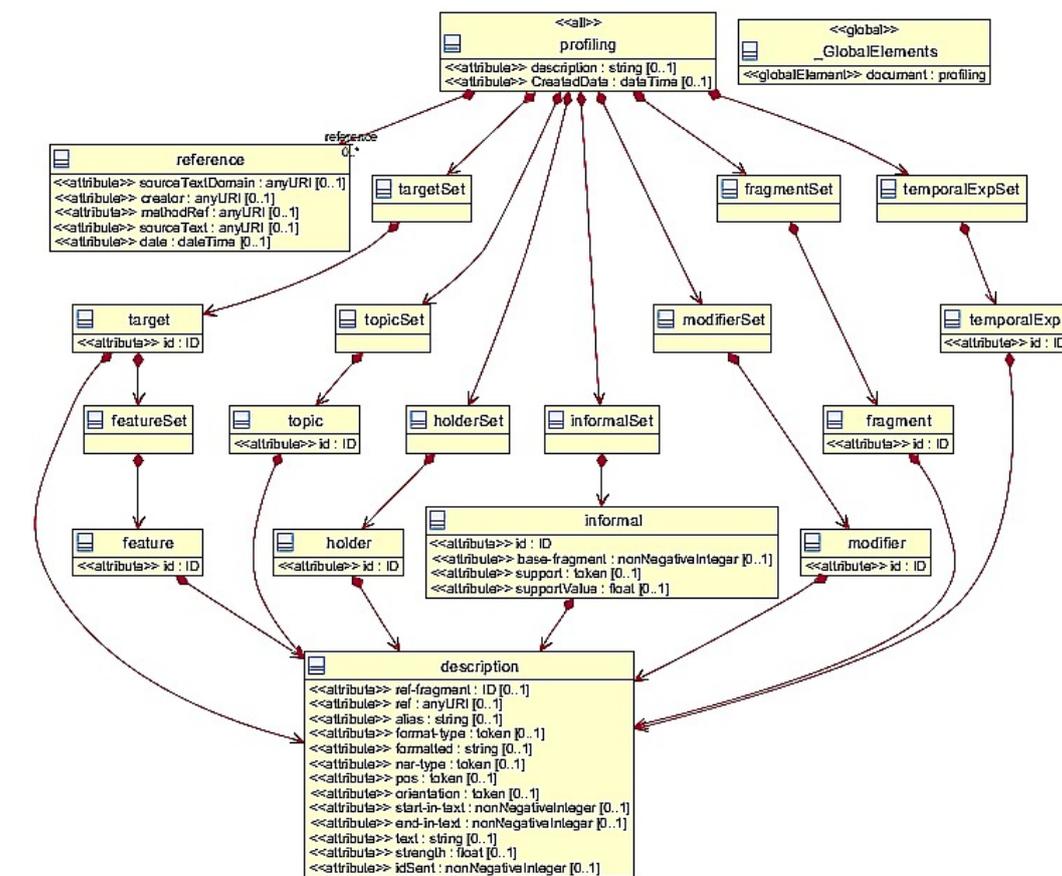


Figure A1. OpinionML: Profiling elements.

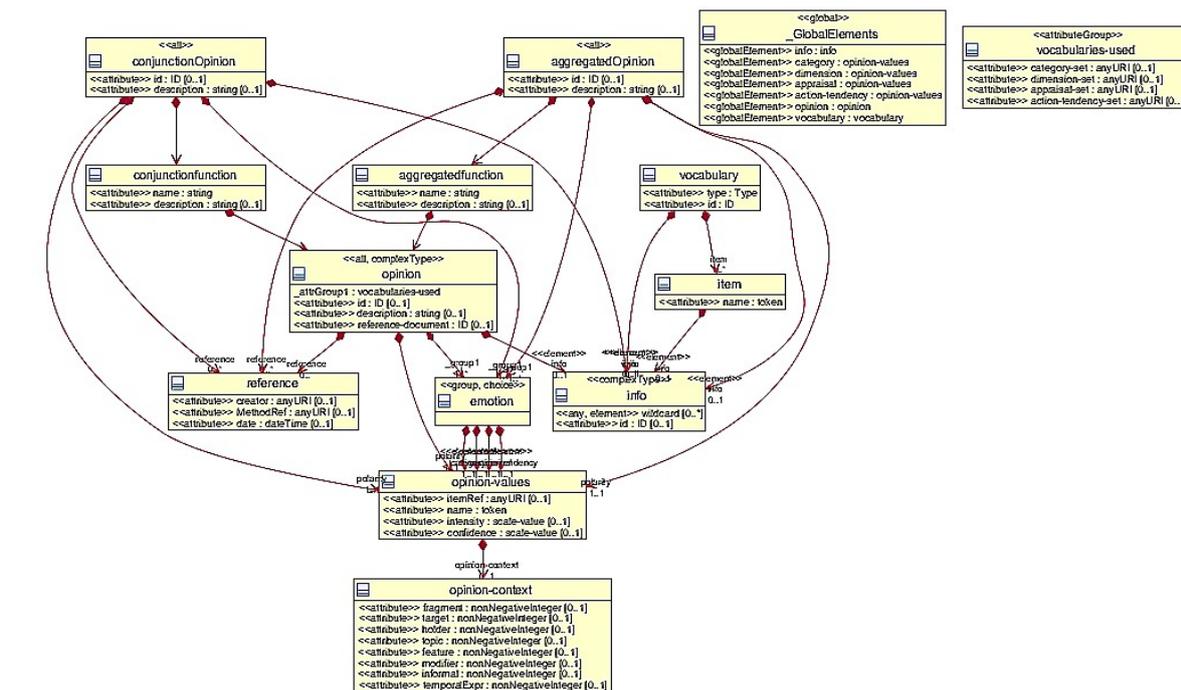


Figure A2. OpinionML: Opinion elements.

Appendix A.1. The <OpinionML> Element

This is the root element of OpinionML which connects all other elements of OpinionML (please see Listing 8).

Attributes

- Optional
 - category-set, dimension-set, appraisal-set or action-tendency-set: This attribute is used to define the global emotion vocabulary set for the current OpinionML document. The category-set, dimension-set, appraisal-set or action-tendency-set can be used to declare a global vocabulary (of type category, dimension, appraisal or action-tendency respectively). The attribute MUST be a URI.
 - lexicon: This is the reference to the lexicon being used to compute the prior polarities of the modifiers.
- Required
 - version : This attribute defined the version of OpinionML being used.

Appendix A.2. The <profiling> Element

This is the first element of the OpinionML description containing profiles of several entities of OpinionML in the form of sets. <profiling> is globally described by its child element <reference> which contains attributes described below (please see Listing 9).

Attributes

- Optional
 - sourceText: It is the link towards the text document being annotated,
 - sourceTextDomain: This attribute defined the domain of the text i.e., whether this is a document talking about sports, politics, or some event etc.

- creator: This attribute contains the name of the person annotating the document,
- methodRef: Any method used to create profiling of this document can be listed here.
- Required
 - time: This is the time of the creation/publication of the document being annotated.

Appendix A.3. The <opinionSet> Element

This element <opinionSet> describes the start of opinion section of the OpinionML. It contains many <opinion> elements describing an annotated opinion while a <reference> element giving global information about the <opinionSet>. Each <opinion> is further described by a child elements like <context>, <emotion> and <polarity>. attributes (described below). Besides this, an <opinion> contains other child elements i.e., <context>, <emotion> and <polarity>. The <context> elements link the items forming an opinion from the profiling section while <emotion> and <polarity> describe the emotion vocabulary of the opinion and its polarity respectively. Please see Listing 10.

Attributes

- Optional
 - methodRef (<reference>): This attribute describes whether method used for opinion detection was manual or algorithmic.
 - value (<polarity>): This is the value of the polarity of the opinion as computed.
 - confidence (<polarity>): The confidence value of the annotator or precision of the algorithm used. The value remains between 0 and 1.
- Required
 - id (<opinion>): A number uniquely identifying an opinion,
 - fragment (<context>): It is reference to the fragment about which opinion is being expressed,
 - holder, target, topic, feature, modifier, informal, temporalExp (<context>): All these attributes provide IDs (or list of IDs) to their corresponding entities defined in the profiling section to represent entity of an opinion. For example, if a fragment contains an informal or cultural phrase then this attribute will contain reference of that informal phrase already existing in profiling section of OpinionML. It becomes necessary in this case that semantic orientation of the current fragment be judged without considering the informal/cultural phrase which is being referenced.
 - name (<category>)
 - name (<polarity>) This attribute represents the annotated semantic orientation of the opinion. It could have one of three values i.e., “positive”, “negative” or “neutral”.

Appendix A.4. The <fragmentSet> Element

Fragments in OpinionML are segments of text that have some semantics of opinion. OpinionML gives liberty to the users to define the granularity of fragments themselves unlike others where this granularity is imposed. This element <fragments> contain one or more <fragment> elements and is defined in <profiling> section of OpinionML. Its elements can be used by giving references. Please see Listing 11.

Attributes of the element <fragment> are listed below:

Attributes

- Optional
 - NA

- Required
 - id: Unique identity of the fragment.
 - start: A number representing start of the text of this fragment in the text,
 - end: A number representing end of the text of this fragment in the text,
 - SentID: Reference of the sentence this fragment is part of,
 - text: It represents the exact text of the fragment as found in the given document to be annotated.

Even OpinionML works on user-defined granularity levels but it keeps track of the textual sentences to which these fragments belong (see attribute “*sentid*”). Sentence IDs (*sentid*) to the sentences are allocated in order as they appear in the given text. It helps in responding many extra questions in case it is needed. Therefore, *sentid* = “10” means 10th sentence of the text.

Appendix A.5. The <holderSet> Element

Opinion holder means the entities that holds a specific opinion on a particular topic or issue. <holders> element contain many but unique <holder> elements. Each <holder> has a unique identity using which a holder can be referenced anywhere in the document. Please see Listing 12.

Attributes

- Optional
 - alias: Set of alias for the holder entity as found in the used knowledge base
- Required
 - id: Unique identity of the holder.
 - text: Exact text of the holder as found,
 - ner-type: Type of the holder entity i.e., whether it is person, organization, country, place or thing,
 - ref-fragment: It is the list of fragments with their identifiers where this particular holder or its aliases appear as their text or as pronouns. It becomes necessary to keep this to resolve anaphore problem,
 - orientation: If the holder entity is “positive”, “negative” or “neutral” in its perception. For example, a murderer found to be holder in a text would be taken as “negative”,
 - ref-uri: URI reference of the holder in the knowledge base used.

Appendix A.6. The <TargetSet> Element

Opinion target can be a product, person, event, organization, or topic on which an opinion is expressed. The <Targets> element contains one or more <Target> elements each identified by a unique identifier. Please see Listing 13.

Attributes

- Optional
 - alias: Set of alias for the target entity as found in the used knowledge base
- Required
 - id: Unique identity of the target.
 - text: Exact text of the target as found,

- ner-type: Type of the target entity, i.e., whether it is person, organization, country, thing, place or concept (like, love, etc.),
- ref-fragment: It is the list of fragments with their identifiers where this particular holder or its aliases appear as their text or as pronouns. It becomes necessary to keep this to resolve the anaphore problem,
- orientation: If the target entity is “positive”, “negative” or “neutral” in its perception.
- ref-uri: URI reference of the target in the knowledge base used.

A target may be considered as a stratum of components (and sub-components) where each component may have a set of attributes. For instance, a mobile phone has the following components: a screen, a battery with other sub-components; the attributes of a mobile phone may include the size and the weight. These components and attributes are named collectively as aspects.

Appendix A.7. The *<featureSet>* Element

<features> elements describes all features on which opinion has been expressed found in the text to be annotated using element *<feature>*. Generally, a *<feature>* element is supposed to be linked to a *<target>* element but not necessarily. A *<feature>* element is equipped with many attributes. Please see Listing 14.

Attributes

- Optional
 - alias: Set of alias for the feature as found in the used knowledge base
- Required
 - id: Unique identity of the feature,
 - text: Exact text of the target as found,
 - ref-fragment: It is the list of fragments with their identifiers where this particular feature or its aliases appear,
 - value-type: The type of value this feature could have, i.e., “numeric”, “category” or “mixed”.
 - orientation: If the feature is “positive”, “negative” or “neutral” in its perception.
 - ref-uri: URI reference of the feature in the knowledge base used. It is especially important for features which can be referred to using several different terms. For example, the sentence “I cannot hear well” or “the speakers are weak” talk about the same feature, i.e., sound of a television but using different vocabulary. Such problems are hard to deal with but using an entry of the knowledge base as a reference (like “sound” in this case) could be useful to resolve this problem.

Appendix A.8. The *<modifierSet>* Element

The element *<modifiers>* contains many *<modifier>* elements. Modifiers are the subjective words that give the semantics of opinions to the text they appear in. Words like “excellent”, “poor” etc. are examples of modifiers. Please see Listing 15.

Attributes

- Optional
 - alias: Set of alias for the modifier as found in the used knowledge base
 - strength: It shows the intensity of semantic orientation a modifier has. It could be extracted from a lexical resource or given by the annotators. Generally, value remains between 0 and 1.

- Required
 - id: Unique identity of the modifier,
 - text: Exact text of the modifier as found,
 - pos: pos stands for “parts of speech” i.e., whether the feature is a noun, adjective, verb or adverb.
 - orientation: If the modifier is “positive”, “negative” or “neutral” as found in one of the lexicons,
 - ref-fragment: The list of fragments where this modifier or its alias appear,
 - ref-uri: URI reference of the modifier in the knowledge base used.

Appendix A.9. The <informalSet> Element

Opinions are frequently written using informal languages that include the use of emoticons and ironical terms like “it slipped my mind”, “your performance was killing me” etc. These terms may create confusion while analyzing the text surrounding these phrases. There is a need to develop a system that helps in auto-detection and extraction of such phrases from opinions.

The usage of informal language in giving opinions or writing some review of a product is increasing with the increase of online social sites [49]. The identification of sarcastic phrases, proverbs, idioms and emoticons (like ☹️ etc.) in an opinionated text is not an easy task using conventional NLP approaches. Therefore, the researchers have to adopt some novel ideas to perform this interesting task. One of the ideas is to annotate a data corpus with such type of sarcastic terms and phrases, and then by applying efficient machine learning models we can predict the actual semantic of the statement. In order to perform this task, we annotate a textual segment “*You have a huge belly because you’re a couch potato.*” using some sarcastic words. Please see Listing 16.

Attributes

- Optional
 - NA
- Required
 - id: Unique identity of the fragment.
 - start: A number which tells the starts of the informal expression as given in the attribute “text”
 - end: A number which tells the end of the informal expression as given in the attribute “text”
 - orientation: The actual semantics of the given phrase and not literal semantics.
 - support: This attribute could have three possible values i.e., “favors”, “reverses” or “neutral” meaning that whether this phrase supports the semantics of actual phrase it appeared in or it reverses it or it stays neutral,
 - ref-fragment: The list of fragments where this modifier or its alias appear,
 - base-fragment: This is the ID of the fragment this particular phrase appears in.

Appendix A.10. The <topicSet> Element

<topics> element is used to capture knowledge about the topics relevant to different fragments. The idea behind capturing topics is to analyze the topic-based contextual polarities of words i.e., modifiers. For example, the term “predictable” may have positive meaning if written in describing the feature of a specific product in some review (e.g., picture quality of camera) and this same word may depict the negative sense while talking about some movie related feature (e.g., story plot of the movie). In order to develop a better opinion mining system, one has to collect an appropriate amount of data corpus annotated with sarcastic and informal terms from different domains. Please see Listing 17.

Attributes

- Optional
 - NA
- Required
 - id: Unique identity of the topic.
 - ref: The referenced source used to find the topic. Topics are extracted for each defined fragment but only unique topics are kept in the profiling section and then each fragment provides a reference to the related fragment,
 - chain: The actual hierarchy of topics as found in the referenced source of knowledge from which topic has been detected,
 - orientation: If the topic detected is positive, negative or neutral in its semantics; For example, a topic labelled as “festival” could be considered as positive while a topic relevant to “wars” would be tagged as negative.
 - ref-fragment: The list of fragments this topic is associated with separated by a comma.

Appendix A.11. The <temporalExpSet> Element

The <temporalExpSet> is composed of many <temporalExp> elements. Each <temporalExp> element groups information about a temporal expression as found in the given text. Temporal expressions are very important to predict time-based trending of different issues (for example, to predict behavior of stock exchange next month). Therefore, OpinionML pays attention to processing of temporal expression and can integrate the TIMEML standard (TIMEX3) [50] or another standard for representing temporal expressions by using the “formatted” element. Please see Listing 18.

Attributes

- Optional
 - NA
- Required
 - id: Unique identity of the time fragment.
 - text: Exact text of the expression as it exists in the given text,
 - type: It contains one of the following values i.e., Date, Duration, Set or Time. Details about range of temporal expression that these values cover can be consulted in documentation of TimeML [51].
 - value: It lists the TIMEML equivalent expression of the given temporal expression. For example, if the given temporal expression is at date 24 April then its TIMEML equivalent will be XXXX-04-24. The value XXXX replaces the missing year.
 - nature: Whether the time mentioned in the text is referring to real time or something virtual time like “here today, gone tomorrow” (meaning appearing or existing only for a short time).
 - tense: It represents whether the time expression referring to some past, present or future time.
 - start: A number which tells the starts of the time expression as given in the attribute “text”
 - end: A number which tells the end of the time expression as given in the attribute “text”
 - ref-fragment: The list of fragments where this time expression appears.

Appendix B. Listings

Listing 4: Vocabulary Section

```
<vocabularies>

<vocabulary type="category" id="big6">
<item name="anger"/>
<item name="disgust"/>
<item name="fear"/>
<item name="happiness"/>
<item name="sadness"/>
<item name="surprise"/>
</vocabulary>

<vocabulary type="dimension" id="fsre-dimensions">
<item name="valence"/>
<item name="potency"/>
<item name="arousal"/>
<item name="unpredictability"/>
</vocabulary>

<vocabulary type="action-tendency" id="frijda-subset">
<item name="approach"/>
<item name="avoidance"/>
<item name="rejecting"/>
</vocabulary>

<vocabulary type="appraisal" id="occ-appraisals">
<item name="desirability"/>
<item name="praiseworthiness"/>
<item name="appealingness"/>
<item name="desirability-for-other"/>
<item name="deservingness"/>
<item name="liking"/>
<item name="likelihood"/>
<item name="effort"/>
<item name="strength-of-identification"/>
<item name="expectation-of-deviation"/>
<item name="familiarity"/>
</vocabulary>

</vocabularies>
```

Listing 5: Vocabulary Citation in Opinion Element of OpinionML

```
<opinion category-set="#big6" action-tendency-set="#frijda-subset">
<category name="fear"/>
<action-tendency name="approach" value="0.0"/>
<action-tendency name="avoidance" value="0.9"/>
</opinion>

<opinion dimension-set="#fsre-dimensions">
<dimension name="arousal" value="0.3"/>
</opinion>
```

Listing 6: External Vocabulary Section Reference in Opinion Element

```
<opinion category-set="http://www.w3.org/TR/emotion-voc/xml#big6">
```

Or please see Listing 7.

Listing 7: Vocabulary section reference in OpinionML Element

```
<OpinionML category-set="http://www.w3.org/TR/emotion-voc/xml#big6" version="1.0">
```

Listing 8: OpinionML Element

```
<OpinionML category-set= <link> dimension-set=<link> appraisal-set=<link>
  action-tendency-set= <link> lexicon= <string> version="1.0">
</OpinionML>
```

Listing 9: Profiling Element

```
<profiling>
<reference time= <time> sourceText= <link> sourceTextDomain= <string>
creator= <string>
  methodRef= <string>/>
</profiling>
```

Listing 10: OpinionSet Element

```
<opinionSet>
<reference methodRef= <Algorithm/Manual> />
<opinion id= <ID> >
<context fragment= <idref> holder= <idref> target= <idref> topic= <idref>
  feature= <idref> modifier= <idref> informal= <idref> temporalExp=
  <idref> time=<dateTime>/>
<emotion>
<category name= <string>/>
</emotion>
<polarity name= <positive/negative/neutral> value= <float> confidence=
  <float> />
</opinion>
</opinionSet>
```

Listing 11: FragmentSet Element

```
<fragmentSet>
<fragment id= <ID> >
<description start= <numeric> end= <numeric> sentid= <numeric> text=
  <string> />
</fragment>
</fragmentSet>
```

Listing 12: HolderSet Element

```
<holderSet>
<holder id= <ID> >
<description text= <string> ner-type="country" ref-fragment= <idref> orientation=
  <positive/negative/neutral> ref-uri= <link> />
</holder>
</holderSet> %\label{lst:hldrset}
```

Listing 13: TargetSet Element

```
<targetSet>
<target id= <ID> >
<description text= <string> alias= <string> ner-type= <person, organization, place,
  country> ref-fragment= <idref> orientation= <positive/negative/neutral>
  ref-uri= <link> />
</target>
</targetSet>
```

Listing 14: FeatureSet Element

```

<featureSet>
<feature id= <ID> >
<description text= <string> alias= <string> ref-fragment= <idref> value-type=
  <numeric/categorical> orientation= <positive/negative/neutral> ref-uri=
  <link>/>
</feature>
</featureSet>

```

Listing 15: ModifierSet Element

```

<modifierSet>
<modifier id= <ID> >
<description text= <string> alias= <string> pos= <adjective/adverb/
  noun/verb>
  orientation= <positive/negative/neutral> strength= <float>
  ref-fragment= <idref> ref-uri= <link> />
</modifier>
</modifierSet>

```

Listing 16: InformalSet Element

```

<informalSet>
<informal id= <ID> >
<description start= <numeric> end= <numeric> text= <string> orientation=
  <positive/negative/neutral> support= <favors/reverses/neutral> ref-fragment
  = <idref> base-fragment= <idref> />
</informal>
</informalSet>

```

Listing 17: TopicSet Element

```

<topicSet>
<topic id= <ID> >
<description ref= <link> orientation= <positive/negative/neutral> chain= <string>
  ref-fragment= <idref> />
</topic>
</topicSet>

```

Listing 18: TemporalExpSet Element

```

<temporalExpSet>
<temporalExp id= <ID>>
<description start= <numeric> end= <numeric> text= <string> type="timeML-timex3"
  ref-fragment= <ID> >
<timex3 type= <Date/Duration/Set/Time> value= <string> nature=
  <real/conceptual> tense= <past/present/future> />
</description>
</temporalExp>
</temporalExpSet>

```

Listing 19: An OpinionML Example

```

<OpinionML category-set="http://www.w3.org/TR/emotion-voc/xml#big6"
  lexicon="SentiWordNet1.0" version="1.0">

<profiling>

<fragmentSet>

<fragment id="FRG0">
<description start="" end="" sentid="S10" text="The U.S. State Department on
  Tuesday (KST) rated the human rights situation in North Korea poor in
  its annual human rights report"/>

```

```

<fragment id="FRG01">
<description start="" end="" sentid="S10" text="casting_dark_clouds_on_the_
already tense relationship between Pyongyang and Washington."/>
</fragmentSet>
<holderSet>
<holder id="H0">
<description text="U.S._State_Department" alias="" ner-type="organization"
ref-fragment="FRAG0" orientation="neutral" ref-uri="xxx" />
</holderSet>
<targetSet>
<target id="T0">
<description text="North_Korea" alt="N._Korea" ner-type="country" ref-fragment="
FRAG0" orientation="neutral" ref-uri="yyy" />
</target>
<target id="T1">
<description text="Pyongyang" alt="" ner-type="place" ref-fragment="FRAG01"
orientation="neutral" ref-uri="yyy" />
</target>
<target id="T2">
<description text="Washington" alt="" ner-type="place" ref-fragment="FRAG01"
orientation="neutral" ref-uri="yyy" />
</target>
</targetSet>
<featureSet>
<feature id="F0">
<description text="human_rights_situation" alias="" ref-fragment="FRAG0" value-type
="categorical" orientation="neutral" ref-uri="" />
</feature>
<feature id="F1">
<description text="relationship" alias="" ref-fragment="FRAG01" value-type=
"categorical" orientation="neutral" ref-uri="" />
</feature>
</featureSet>
<modifierSet>
<modifier id="M0">
<description text="poor" pos="adjective" orientation="negative" ref-fragment=
"FRAG0" ref-uri="xxx" lsttarget="xxx" />
</modifier>
<modifier id="M01">
<description text="dark" pos="adjective" orientation="negative" ref-fragment=
"FRAG01" ref-uri="xxx" lsttarget="xxx" />
</modifier>
<modifier id="M02">
<description text="tense" pos="adjective" orientation="negative" ref-fragment=
"FRAG01" ref-uri="xxx" lsttarget="xxx" />
</modifier>
</modifierSet>
<temporalExprSet>
<temporalExpr id="TMPO">

```

```

<description start="" end="" text="Tuesday" ref-fragment="FRAGO" type=
  "TimeML-timex3">
<timex3 type="DATE" value="XXXX-WXX-2" nature="real" tense="past" />
</description>
</temporalExpr>

</temporalExpSet>

<topicSet>
<topic id="TP0">
<description ref="www.dmoz.org" orientation="neutral" chain="Society:Issues:Human
  Rights and Liberties" ref-fragment="FRAGO,□FRAG01" />
</topic>
</topicSet>

<informalSet>

<informal id="IF0">
<description start="" end="" text="casting□dark□clouds" orientation="negative"
  support= "favors" base-fragment= "FRG01" />
</informal>

</informalSet>

</profiling>

<opinionSet>

<reference creator="" MethodRef="Manual" creator="SAAD-MISSEN"/>

<opinion id="OP0">

<context fragment="FRG0" holder="H0" target="T0" topic="TP0" feature="F0"
  modifier="M0" informal="na" temporalExp="TMPO"/>
<emotion>
<category name="reproach"/>
</emotion>

<polarity name="negative" value="" confidence=""/>

</opinion>

<opinion id="OP01">

<context fragment="FRG01" holder="na" target="T1,T2" topic="TP0" feature="F0"
  modifier="M0" informal="IF0" temporalExp="na"/>
<emotion>
<category name="disappointment"/>
</emotion>

<polarity name="negative" value="" confidence=""/>

</opinion>

</opinionSet>

</OpinionML>

```

Listing 20: Core Opinion Representation

```

<category name="affectionate" />
<dimension name="valence" value="0.9" />
<appraisal name="agent-self" />
<action-tendency name="approach" />

```

References

- Murphy, J.; Link, M.W.; Childs, J.H.; Tesfaye, C.L.; Dean, E.; Stern, M.; Pasek, J.; Cohen, J.; Callegaro, M.; Harwood, P. *Social Media in Public Opinion Research: Report of the AAPOR Task Force on Emerging Technologies in Public Opinion Research*; American Association of Public Opinion Research (AAPOR): Anaheim, CA, USA, 2014.
- Thelwall, M. Blog searching: The first general-purpose source of retrospective public opinion in the social sciences? *Online Inf. Rev.* **2007**, *31*, 277–289. [[CrossRef](#)]
- Verma, B.; Thakur, R.S. Sentiment Analysis Using Lexicon and Machine Learning-Based Approaches: A Survey. In *Proceedings of the International Conference on Recent Advancement on Computer and Communication*; Springer: Singapore, 2018; pp. 441–447.
- Kiyavitskaya, N.; Zeni, N.; Cordy, J.R.; Mich, L.; Mylopoulos, J. Semi-Automatic Semantic Annotations for Web Documents. In *Proceedings of the SWAP 2005, 2nd Italian Semantic Web Workshop, Trento, Italy, 14–16 December 2005*; pp. 14–15.
- Oren, E.; Muller, K.; Scerri, S.; Handschuh, S.; Sintek, M. What Are Semantic Annotations? 2006. Available online: <https://www.ontotext.com/knowledgehub/fundamentals/semantic-annotation/> (accessed on 1 March 2019).
- Hazarika, D.; Poria, S.; Gorantla, S.; Cambria, E.; Zimmermann, R.; Mihalcea, R. CASCADE: Contextual Sarcasm Detection in Online Discussion Forums. *arXiv* **2018**, arXiv:1805.06413.
- AL-Sharuee, M.T.; Liu, F.; Pratama, M. Sentiment analysis: An automatic contextual analysis and ensemble clustering approach and comparison. *Data Knowl. Eng.* **2018**, *115*, 194–213. [[CrossRef](#)]
- Das, S.; Behera, R.K.; Rath, S.K. Real-Time Sentiment Analysis of Twitter Streaming data for Stock Prediction. *Procedia Comput. Sci.* **2018**, *132*, 956–964. [[CrossRef](#)]
- Holzinger, A. *Social Media Mining and Social Network Analysis: Emerging Research*; Emerald Group Publishing Limited Howard House: Bingley, UK, 2014.
- Kim, S.M.; Hovy, E. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text, Sydney, Australia, 22 July 2006*; pp. 1–8.
- Thelwall, M. Gender bias in sentiment analysis. *Online Inf. Rev.* **2018**, *42*, 45–57. [[CrossRef](#)]
- Thelwall, M. Gender bias in machine learning for sentiment analysis. *Online Inf. Rev.* **2018**, *42*, 343–354. [[CrossRef](#)]
- Ravi, K.; Ravi, V. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowl.-Based Syst.* **2015**, *89*, 14–46. [[CrossRef](#)]
- Kim, Y.; Dwivedi, R.; Zhang, J.; Jeong, S.R. Competitive intelligence in social media Twitter: iPhone 6 vs. Galaxy S5. *Online Inf. Rev.* **2016**, *40*, 42–61. [[CrossRef](#)]
- Saad Missen, M.M.; Coustaty, M.; Salamat, N.; Prasath, V.S. SentiML++: An extension of the SentiML sentiment annotation scheme. *New Rev. Hypermedia Multimed.* **2018**, *24*, 28–43. [[CrossRef](#)]
- Liu, B. Sentiment Analysis and Subjectivity. *Handb. Nat. Lang. Process.* **2010**, *2*, 627–666.
- Missen, M.M.S.; Boughanem, M.; Cabanac, G. Challenges for Sentence Level Opinion Detection in Blogs. In *Proceedings of the Eighth IEEE/ACIS International Conference on Computer and Information Science*, Shanghai, China, 1–3 June 2009; pp. 347–351.
- Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. *Assoc. Comput. Linguist.* **2002**, *10*, 79–86. doi:10.3115/1118693.1118704. [[CrossRef](#)]
- Maynard, D.; Bontcheva, K.; Rout, D. Challenges in developing opinion mining tools for social media. In *Proceedings of the @ NLP Can U Tag# Usergeneratedcontent, Istanbul, Turkey, 21–27 May 2012*; pp. 15–22.

20. Yang, H.L.; Chao, A.F. Sentiment annotations for reviews: An information quality perspective. *Online Inf. Rev.* **2018**, *42*, 579–594. [CrossRef]
21. Turney, P.D.; Littman, M.L. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Trans. Inf. Syst.* **2003**, *21*, 315–346. doi:10.1145/944012.944013. [CrossRef]
22. Zhao, J.; Dong, L.; Wu, J.; Xu, K. Moodlens: An emoticon-based sentiment analysis system for chinese tweets. In Proceedings of the 18th ACM SIGKDD International Conference On Knowledge Discovery and Data Mining, Beijing, China, 12–16 August 2012; pp. 1528–1531.
23. Wilson, T.; Wiebe, J.; Hoffmann, P. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology And Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 347–354.
24. Councill, I.G.; McDonald, R.; Velikovich, L. What’s Great and What’s Not: Learning to Classify the Scope of Negation for Improved Sentiment Analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing; NeSp-NLP ’10*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 51–59.
25. Kennedy, A.; Inkpen, D. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Comput. Intell.* **2006**, *22*, 110–125. [CrossRef]
26. Krestel, R.; Siersdorfer, S. Generating Contextualized Sentiment Lexica Based on Latent Topics and User Ratings. In Proceedings of the 24th ACM Conference on Hypertext and Social Media, Paris, France, 1–3 May 2013; pp. 129–138. doi:10.1145/2481492.2481506. [CrossRef]
27. Stylios, G.; Tsolis, D.; Christodoulakis, D. Mining and Estimating Users’ Opinion Strength in Forum Texts Regarding Governmental Decisions. In *Artificial Intelligence Applications and Innovations*; Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., Sioutas, S., Eds.; IFIP Advances in Information and Communication Technology; Springer: Berlin/Heidelberg, Germany, 2012; Volume 382, pp. 451–459. doi:10.1007/978-3-642-33412-2_46.
28. Efron, M. Cultural orientation: Classifying subjective documents by co-citation analysis. In Proceedings of the AAAI Fall Symposium on Style and Meaning in Language, Art, and Music, Washington, DC, USA, 21–24 October 2004; pp. 41–48.
29. Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Mohammad, A.S.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. SemEval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 19–30.
30. Di Bari, M.; Sharoff, S.; Thomas, M. SentiML: Functional Annotation for Multilingual Sentiment Analysis. In Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment: Metadata, Vocabularies and Techniques in the Digital Humanities, Florence, Italy, 10 September 2013; ACM: New York, NY, USA, 2013; pp. 15:1–15:7. doi:10.1145/2517978.2517994. [CrossRef]
31. Robaldo, L.; Caro, L.D. OpinionMining-ML. *Comput. Standards Interfaces* **2013**, *35*, 454–469. [CrossRef]
32. Schroder, M.; Baggia, P.; Burkhardt, F.; Pelachaud, C.; Peter, C.; Zovato, E. EmotionML—An upcoming standard for representing emotions and related states. In *International Conference on Affective Computing and Intelligent Interaction*; Springer: Berlin/Heidelberg, Germany, 2011.
33. Cambria, E.; Das, D.; Bandyopadhyay, S.; Feraco, A. Affective computing and sentiment analysis. *IEEE Intell. Syst.* **2016**, *30*, 102–107. [CrossRef]
34. Shankland, S. *EmotionML: Will Computers Tap into Your Feelings?* CNET News, 30 August 2010. Available online: <https://www.cnet.com/news/emotionml-will-computers-tap-into-your-feelings/> (accessed on 1 March 2019.)
35. Liu, B. *Sentiment Analysis and Opinion Mining: Synthesis Lectures on Human Language Technologies*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2012; p. 167. doi:10.2200/S00416ED1V01Y201204HLT016.
36. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Inf. Retrieval* **2008**, *2*, 1–135. [CrossRef]
37. Cambria, E.; Schuller, B.; Xia, Y.; Havasi, C. New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intell. Syst.* **2013**, *28*, 15–21. doi:10.1109/MIS.2013.30. [CrossRef]
38. Swami, A.; Mete, A.; Bhosle, S.; Nimbalkar, N.; Kale, S. *Ferom: Feature Extraction and Refinement for Opinion Mining*; Wiley Online Library: Hoboken, NJ, USA, 2017; pp. 720–730.

39. Munezero, M.; Montero, C.S.; Sutinen, E.; Pajunen, J. Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text. *Affect. Comput. IEEE Trans.* **2014**, *5*, 101–111. doi:10.1109/taffc.2014.2317187. [CrossRef]
40. Westerski, A.; Iglesias, C.A.; Ric, F.T. Linked opinions: Describing sentiments on the structured web of data. In Proceedings of the 4th International Workshop Social Data on the Web (SDoW2011), Bonn, Germany, 23 October 2012; pp. 10–21.
41. Sánchez-Rada, J.F.; Iglesias, C.A. Onyx: Describing Emotions on the Web of Data. In Proceedings of the First International Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and perspectives from AI (ESSEM 2013), Torino, Italy, 3 December 2013; AI*IA, Italian Association for Artificial Intelligence, CEUR-WS: Torino, Italy, 2013; Volume 1096, pp. 71–82.
42. Dragoni, M.; Poria, S.; Cambria, E. OntoSenticNet: A commonsense ontology for sentiment analysis. *IEEE Intell. Syst.* **2018**, *33*, 77–85. [CrossRef]
43. Peroni, S. Graffoo: Graphical Framework for OWL Ontologies; 2011. Available online: <https://opencitations.wordpress.com/2011/06/29/graffoo-a-graphical-framework-for-owl-ontologies/> (accessed on 1 March 2019).
44. Abecker, A.; van Elst, L. *Ontologies for Knowledge Management, Handbook on Ontologies*; Springer: New York, NY, USA, 2004; pp. 435–454.
45. Van Elst, L.; Abecker, A. Ontologies for information management: Balancing formality, stability, and sharing scope. *Expert Syst. Appl.* **2002**, *23*, 357–366. [CrossRef]
46. Wilson, T.A. Fine-Grained Subjectivity and Sentiment Analysis: Recognizing The Intensity, Polarity, and Attitudes of Private States. Ph.D. Thesis, University of Pittsburgh, Pittsburgh, PA, USA, 2008.
47. Fleiss, J.L.; Cohen, J. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educ. Psychol. Meas.* **1973**, *33*, 613–619. [CrossRef]
48. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press: New York, NY, USA, 1999; Volume 463.
49. He, W.; Tian, X.; Tao, R.; Zhang, W.; Yan, G.; Akula, V. Application of social media analytics: A case of analyzing online hotel reviews. *Online Inf. Rev.* **2017**, *41*, 921–935. [CrossRef]
50. Pustejovsky, J.; Castano, J.M.; Ingria, R.; Sauri, R.; Gaizauskas, R.J.; Setzer, A.; Radev, D.R. Timeml: Robust specification of event and temporal expressions in text. *New Direct. Quest. Answ.* **2003**, *3*, 28–34.
51. Sauri, R.; Littman, J.; Knippen, B.; Gaizauskas, R.; Setzer, A.; Pustejovsky, J. *TimeML Annotation Guidelines*, version 1.2.1; 2006. Available online: https://www.researchgate.net/profile/James_Pustejovsky/publication/248737128_TimeML_Annotation_Guidelines_Version_121/links/55c9d67c08aeb97567483792.pdf (accessed on 1 March 2019).

