# A First Approach to Mining Opinions as Multisets through Argumentation


Carlos I. Chesñevar[1,2], María Paula González[1,2]
Kathrin Grosse[3], and Ana G. Maguitman[1,2]

[1]Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering
Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, ARGENTINA
[2]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), ARGENTINA
EMAIL: `kgrosse@uos.de` `{cic,mpg,agm}@cs.uns.edu.ar`
[3] Institut für Kognitionswissenschaft – Universität Osnabrück. Osnabrück, GERMANY



**Abstract.** Web 2.0 technologies have resulted in an exponential growth of text-based opinions coming from different sources (such as online news media, microblogging platforms, social networks, online review systems, etc.). The assessment of such opinions has gained considerable interest within several research communities in Computer Science, particularly in the context of modelling decision making processes. In this context, the scientific study of emotions in opinions associated with a given topic has become particularly relevant. Some approaches for assessing emotions in text-based opinions have been developed, resulting in promising software tools for sentiment analysis. In spite of the existence of such tools, assessing and contrasting text-based opinions is indeed a difficult task. On the one hand, complex opinions are built in many cases bottom up, emerging by aggregation from individual opinions posted online. On the other hand, contradictory and potentially inconsistent information might arise when contrasting such complex opinions. This article introduces an argument-based framework which allows to mine text-based opinions based on incrementally generated topics along with partially-ordered features, which provide a multidimensional comparison criterion. Given a topic, we will model an atomic opinion supporting it as a multiset (or bag) of terms. Atomic opinions can be aggregated, and related to alternative opinions, based on expanded topics. As a result, we will be able to obtain an "opinion analysis tree", rooted in the first original topic.


## 1 Introduction and motivations

Internet and the evolution of Web 2.0 technologies have resulted in an exponential growth of text-based opinions coming from different sources (such as online news media, microblogging platforms, social networks, online review systems, etc.). The assessment of such opinions has gained considerable interest within several research communities in Computer Science, particularly in the context of modelling decision making processes based on such opinions. In this context, the scientific study of emotions in opinions associated with a given topic has become relevant, consolidating a new area known as *sentiment analysis* [1, 2], with application in several real-world problems such as e-government [3, 4] and stock market analysis [5], among others.

Assessing and contrasting text-based opinions is indeed a difficult task. On the one hand, opinions are built in many cases *bottom up*, emerging from different individual opinions posted by different users. For simplicity, in our analysis we will assume that opinions are built from the aggregation of several *atomic opinions* around a particular *topic*. For example, several reviews associated with a particular tablet device in a shopping site could allow us to have an opinion about that device. Similarly, several comments and posts on Twitter about a particular issue (e.g. *water supply*) could give us reasons to have a particular opinion on that issue. In such examples, every review and every post would correspond to an individual atomic opinion. Additionally, we will assume that every atomic opinion conveys some sentiment or emotion, so that by performing an aggregation of those sentiments we can obtain an overall emotion for an aggregated opinion. Following the previous example, we could have some reviews (atomic opinions) which are *negative* with respect to a particular tablet $T$, whereas some others are *positive*. Similarly, some Twitter posts on a particular issue could be *neutral*, others *positive*, and some other *negative*. In both cases, we assume that an appropriate aggregation of such individual sentiment values can be performed, associating the aggregated valuation with our opinion.

On the other hand, contradictory and potentially inconsistent information might arise when contrasting opinions. Our overall opinion about the *shipping service* provided by the company selling a tablet can be positive, but when considering *shipping service and location* the opinion might change (e.g. shipping service outside the US is bad). Similarly, an analysis of tweets about the topic *water supply* might lead to an overall neutral position, whereas the topic *water supply Africa* might result in a more negative view. In this context, argumentation [6, 7] provides an insightful view, according to which opinions can be assimilated to arguments, and conflicts between opinions correspond to a defeat relationship between arguments. In particular, it must be seen that that our analysis is *multi-dimensional*, as several different individual criteria (expertise, price, provenance, etc.) might be considered in order to model preference between opinions.

This article introduces an argument-based framework which allows to mine opinions from text-based information items based on incrementally generated topics. Given a topic, we will model the notion of *opinion* supporting it as a collection of atomic opinions, which can be aggregated according to certain specific criteria. Based on topic specificity and preferences defined on different dimensions or *features*, opinions can be contrasted with *counteropinions*, which have to be preferred (according to a partial order) wrt the opinion at issue. As a result, we will be able to obtain an "opinion analysis tree", rooted in the first original topic. Distinguished, conflicting elements in an opinion tree lead to so-called "conflict opinion analysis trees", which resemble dialectical trees as those used traditionally in argumentation theory. The proposed approach is based on previous research presented in [8], generalizing and expanding several preliminary ideas originally applied to the Twitter microblogging platform (see discussion in Section 5).

The rest of this article is structured as follows. Section 2 presents an overview of the main elements involved in our proposal (atomic and composed opinions, argumentation, sentiment analysis and context-based search) and their interrelationship. Then, in Section 3 we present a generic formal framework for contrasting opinions, based on the previous intuitions. We also analyze how "opinion analysis trees" can be constructed from user queries, allowing to assess alternative opinion nodes which come into conflict with their parents. Afterwards, in Section 4, we introduce a refined relationship which allows to contrast conflicting opinions, modeled as a conflict opinion analysis tree. Section 5 discusses the main characteristics of our approach as well as some comparisons with related work. Finally, Section 6 summarizes the conclusions and discusses some future work.

## 2 Handling Opinions as Multisets: Context-based Search and Argumentation

According to Merriam Webster online dictionary,[1] an *opinion* can be seen as: a) a view, judgment, or appraisal formed in the mind about a particular matter; b) belief stronger than impression and less strong than positive knowledge; a generally held view; c) a formal expression of judgment or advice by an expert. Nowadays, Internet 2.0 technologies have made possible a plethora of different tools for users to interact and argue about different topics, expressing their opinions in different ways. Approaches are diverse, and range from microblogging platforms like Twitter (where users' tweets convey in many cases a particular opinion using a few characters) to customer complaint and review services (where users can express their view on a particular service or issue, e.g. hotel quality). Most Web 2.0 platforms usually incorporate additional features to identify elements which make an opinion relevant (number of retweets, expertise or reliability of a customer measured in terms of a particular scale, etc.).

In spite of the differences, we can identify some underlying elements which are common to all these different approaches to handling opinions on the Web:

- Opinions are usually *text-based*, with additional elements which make it possible to identify relevant features, as discussed before (provenance, reliability, etc.), and usually imply a *position* or *sentiment* with respect to a particular issue or topic. It must be noted that sometimes opinions are built by *aggregation* of other more simple, *atomic* opinions, according to some particular criterion; thus a bunch of tweets referring positively to a specific hashtag

[1] `http://www.merriam-webster.com`

within certain time boundaries may provide an aggregated positive opinion on that topic (based on the number of individual tweets with the same sentiment). Similarly, several independent hotel reviews referring to a particular hotel as a bad venue for academic events might provide an argument for rejecting that hotel as a candidate for hosting a particular conference. The aggregation of atomic opinions results also in identifying some relevant elements or features for contrasting opinions. Thus, elements such as provenance, expertise, specificity can be used for preferring one aggregated opinion over another.

- In order to analyze opinions concerning a topic, users tend to analyze alternative *counteropinions* in order to make decisions, based on more preferred information. Thus, a hotel criticized in a review for a user for being away from the city center could turn out to be desirable for those users wanting to avoid noise and traffic. Therefore, the way opinions are analyzed and contrasted will depend to a large extent on the preference criteria being considered;
- contrasting opinions and counteropinions has a strong resemblance with the *dialectical processes* associated with argumentation frameworks, in which arguments are compared according to a defeat relationship, and their justification status can be determined using different semantics. In the context of opinion mining, argumentation provides a useful conceptualization for identifying and contrasting conflict relationships among opinions.

Given a query, we will model an aggregated opinion supporting it as a distinguished set of atomic opinions selected according to a given criterion. Aggregated opinions will provide support to *arguments*, identifying a prevailing sentiment,[2] as well as prevailing features, which will correspond to relevant meta-elements in the opinion (such as provenance, expertise, specificity, etc.). Such arguments can be in turn attacked by other alternative, preferred counter-arguments (counter-opinions). As a final result, we will be able to obtain a "conflict opinion analysis tree", rooted in the first original query, in a way that resembles a dialectical tree in argumentation [7].

## 3 Aggregating Opinions as Arguments

In this Section we will describe how a group of different atomic information items elements associated with a given topic can be analyzed under an argumentative perspective. First we will characterize a distinguished collection of information items (obtained on the basis of a given query) as an argument with an associated prevailing sentiment. Arguments will correspond to aggregated opinions, based on atomic opinions. Such arguments will be called aggregated-opinion arguments (*AO* arguments). Then, we will formalize interrelationships between them, leading to the notion of *opinion analysis tree*. As stated before, we will assume that

[2] Several software tools have been recently developed for such an association; e.g. for bunch of tweets (Twitter messages), some available online tools are e.g. `www.sentiment140.com`.

opinions can be modeled in different ways in the context of Web 2.0. Possible representations of an atomic opinion are: a) a review provided by a user using a software facility; b) a post on a social network such as Facebook or Twitter; c) a comment sent by email to a customer service. In all those cases, as it is the usual case in information retrieval settings, we will adopt the *bag of words* representation for capturing what an opinion is. Formally:

**Definition 1 (Atomic Opinion. Topic).** *We define an atomic opinion as a bag (or multiset) of terms* $\{t_1,\, t_2,\, \ldots t_k\, \}$*. A* topic *is a non-empty set* $Q = \{d_1, d_2, \ldots, d_k\}$ *of descriptors, where every* $d_i \in Q$ *is a term. We will write* $\mathfrak{AO}\mathfrak{pinions}$ *to denote the set of all possible atomic opinions.*

A topic $Q$ is any set of descriptors used for *filtering* some relevant atomic opinions in $\mathfrak{AO}\mathfrak{pinions}$ based on a given criterion $C$. In order to abstract away how such a filtering is performed, we will define next an aggregation operator $\mathsf{Agg}(Q, C)$. Formally:

**Definition 2 (Aggregated Opinion. Aggregation Operator).** *We will write* $2^{\mathfrak{AO}\mathfrak{pinions}}$ *to denote the set of all possible subsets of* $\mathfrak{AO}\mathfrak{pinions}$*. Any element in* $2^{\mathfrak{AO}\mathfrak{pinions}}$ *will be called an* aggregated opinion *(AO). Given a query* $Q$*, and a criterion* $C$*, we will define an aggregation operator* $\mathsf{Agg}(Q, C)$ *which returns a set of tweets (aggregated opinion) based on* $Q$ *and* $C$*, i.e.,* $\mathsf{Agg}(Q, C)$ *will return an element* $\{T_1, T_2, \ldots, T_n\}$ *of* $2^{\mathfrak{AO}\mathfrak{pinions}}$ *such that for each* $T_i$*,* $Q \subseteq T_i$ *and* $T_i$ *satisfies criterion* $C$*.*

The aggregation operator could be defined in several ways. For instance, suppose that $C_1$ is a criterion that indicates that only atomic opinions posted between time $timestamp_1$ and $timestamp_2$ are to be selected. Then $\mathsf{Agg}(Q, C_1)$ $=_{def}$ $\{\ T \in \mathfrak{AO}\mathfrak{pinions}$ such that $Q \subseteq T$ and $T$ satisfies $C_1\ \}$ will be the set of all atomic opinions $T$ that contain all the terms of query $Q$ and have been posted in the time period $[timestamp_1, timestamp_2]$. Other examples of criteria which can be naturally applied are, for instance, requiring that atomic opinions come from trusted websites, or have a constrained text length (e.g. less than 500 characters).

Note that for the same query $Q$, different alternative criteria $(C_1, C_2, \ldots, C_k)$ can lead to different distinguished elements in $2^{\mathfrak{AO}\mathfrak{pinions}}$. As explained before, an aggregated opinion can be associated with a particular feeling or sentiment (based on the sentiments associated with the atomic opinions involved) along with a tuple of prevailing feature values (meta-information such as expertise, provenance, etc.). Clearly, there may be a considerable variety of emotions for sentiment analysis (such as anger, happiness, and so on), as well as different possible features. In order to make our approach as general as possible, we will assume here that there is a finite set $\mathbb{S} = \{s_1,\, s_2,\, \ldots,\, s_k\}$ of possible sentiments,[3] and a finite set $\mathbb{F} = \{f_1,\, f_2,\, \ldots,\, f_m\ \}$ of possible features. Every feature $f_i$ will

[3] A possible range for $\mathbb{S}$ could be positive, negative and neutral (as done for example in the platform Sentiment140.com for handling Twitter sentiments).

have different possible values $\{v_1^i, v_2^i, \ldots, v_m^i\}$. We will consider that some sentiments might convey different, possibly *conflicting* feelings or emotions (anger and happiness; boredom and excitement, etc.). As before, we will abstract away which are potentially conflicting sentiments in our conceptualization. From a formal perspective, $conflict$ could be defined as a symmetric and irreflexive binary relation. However, from a computational perspective, it is more effective to define $conflict$ as a function that for any given sentiment returns a set of conflicting sentiments. Formally:

**Definition 3 (*Sent* and *conflict* mappings).** *Let* $AO \in 2^{\mathfrak{AOpinions}}$ *be an aggregated opinion, and let* $Sent : 2^{\mathfrak{AOpinions}} \rightarrow \mathbb{S}$ *and* $conflict : \mathbb{S} \rightarrow 2^{\mathbb{S}}$ *be mappings. The sentiment* $Sent(AO)$ *will be called the prevailing sentiment (or just sentiment) for* $AO$*. For any sentiment* $s \in \mathbb{S}$*, we will define* $conflict(s)$ *as a subset of* $\mathbb{S}$*, such that: a)* $s \notin \mathbb{S}$ *(a sentiment is not in conflict with itself); b) for any* $s' \in conflict(s)$*, then* $s \in conflict(s')$ *(the notion of conflict is symmetrical). Given two sentiments* $s_1$ *and* $s_2$*, we will say that they are* in conflict *whenever* $s_2 \in conflict(s_1)$*. For simplicity, given a sentiment* $s \in \mathbb{S}$*, we will write* $\overline{s}$ *to denote any* $s' \in conflict(s)$*.*

Given a query $Q$ and a criterion $C$, every aggregated opinion $\mathsf{Agg}(Q, C)$ will be associated with a *prevailing sentiment* in $\mathbb{S}$ and a tuple of *prevailing feature values* $(fv_1, fv_2, \ldots, fv_m)$, which correspond to meta-information identifying relevant elements which contribute to prefer one aggregated opinion over another. Every feature $f_i$ will correspond to a a finite set of partially ordered values. All aggregated opinions will be contrasted against the same set of possible features, according to a preference ordering. Formally:

**Definition 4 (Features. Preference on Feature Tuples Instances).** *Let* $AO \in 2^{\mathfrak{AOpinions}}$ *be an aggregated opinion, and let* $\mathbb{F} = \{f_1,\ f_2,\ \ldots f_m\ \}$ *be a set of features, such that every* $f_i$ *has a set of values* $FV_i$*. A* feature tuple *based on* $\mathbb{F}$ *is a tuple* $ft = (f_1, f_2, \ldots, f_k)$*, where every* $f_i \in \mathbb{F}$*. A* feature tuple instance $ft^{\downarrow}$ *based on* $ft$ *is a tuple* $(v_1, v_2, \ldots, v_k)$*, where every* $v_i \in FV_i$*. Let* $\mathbb{FV}^{\downarrow}$ *be the set of all possible feature tuples instances. Then we define the* prevailing feature *(PF) mapping* $PF : 2^{\mathfrak{AOpinions}} \rightarrow \mathbb{FV}^{\downarrow}$*, which for every aggregated opinion* $AO$ *assigns a prevailing feature tuple instance* $(v_1, v_2, \ldots, v_k)$*. We will assume a partial order* $\preceq_{f_i}$ *on the possible values* $FV_i$ *for every feature* $f_i$*.*

The previous elements will allow us to characterize the notion of *Opinion-based Argumentation Framework* and AO-argument as follows:

**Definition 5 (Opinion-based Argumentation Framework).** *An* Opinion-based Argumentation Framework *(or OAF for short) is a tuple* $(\mathfrak{AOpinions}, C, \mathbb{S}, Sent, conflict, \mathbb{F}, PF)$*, where*

- $\mathfrak{AOpinions}$ *is the set of all possible atomic opinions;*
- $C$ *is a selection criterion on* $\mathfrak{AOpinions}$*;*
- $\mathbb{S}$ *is a non-empty set of possible sentiments and* $Sent$ *and* $conflict$ *are sentiment prevailing and conflict mappings;*

– $\mathbb{F}$ *is a non-empty set of features, and* $PF$ *is a prevailing feature mapping.*

**Definition 6 (AO-argument).** *Let* $(\mathfrak{AOpinions}, C, \mathbb{S}, Sent, conflict, \mathbb{F}, PF)$ *be an* $OAF$*. An* aggregated opinion argument *(or* AO-argument*) for a conclusion (query)* $Q$ *is a 4-tuple* $\langle Arg, Q, ps, pf \rangle$*, where* $Arg$ *is* $\mathsf{Agg}(Q, C)$*, the prevailing sentiment* $ps$ *is* $Sent(\mathsf{Agg}(Q, C))$ *and the prevailing features* $pf$ *is the set* $PF(\mathsf{Agg}(Q, C))$*.*

### 3.1 Example

Consider a tablet device XYZ90, which is being sold worldwide by an American company. We assume that several text-based reviews about XYZ90 are available on the web. Salient features relevant for analyzing user opinion are the *country* where the opinions come from (for the sake of simplicity, we assume just US and worldwide opinions), the *expertise* from the user making the review or comment (generic user, expert user) and the *provenance* of the review (generic website, press website, store website and specialized press website). In this setting, an atomic opinion *ao* is just a *single* comment, text or review referring to the XYZ90 tablet, whereas a distinguished set of atomic opinions provides an *aggregated opinion*. Following Def. 4, a feature tuple $(origincountry, expertise, provenance)$ can be defined, where *origincountry*= { *us*, *worldwide* }, *expertise* = { *generic_user*, *expert_user* } and *provenance*= { *generic_website*, *store_website*, *press_website*, *spec_press_website* }, and the following preference criterion on feature values can be established:

– *us* $\preceq$ *worldwide*,
– *generic_user* $\preceq$ *expert_user*,
– *generic_website* $\preceq$ *press_website*, *generic_website* $\preceq$ *store_website*, *press_website* $\preceq$ *spec_press_website*.

For the sake of example, let us consider $\mathbb{S} = \{pos, neg\}$, and a criterion $C$ such that it selects all XML reviews on the Web in the last two years which match a given query $Q$. We also consider the conflict mapping defined as follows: $conflict(pos) = \{neg\}$, $conflict(neg) = \{pos\}$ (i.e., positive and negative opinions are in conflict). The above elements allow to characterize the $OAF =$ ($\mathfrak{AOpinions}$, $C$, $\mathbb{S}$, $Sent$, $conflict$, $\mathbb{F}$, $PF$), considering as well that $PF$ determines the values of a feature tuple instance by majority counting (i.e., the tuple formed by the majority counting of each feature individually). In this setting, some aggregated arguments that could be obtained are:

– $\langle A1$, "XYZ90", $pos, (US, generic\_user, generic\_website)\rangle$, stating that there is a set of atomic opinions $A1$ which supports "XYZ90" with a positive sentiment, associated with the feature instance ($US$, *generic_user*, *generic_website* (i.e., most atomic arguments in $A1$ come from $US$, provided by generic users from generic websites).
– $\langle A2$, "XYZ90, shipping", $pos, (US, generic\_user, generic\_website)\rangle$, stating that there is a set of atomic opinions $A2$ which supports "XYZ90, shipping" with a positive sentiment, associated with the feature instance ($US$, *generic_user*, *generic_website* (i.e., most atomic arguments in $A2$ come from $US$, provided by generic users from generic websites).

- $\langle A3, \text{“}\mathsf{XYZ90,\ shipping}\text{”}, neg, (worldwide, generic\_user, press\_website)\rangle$, stating that there is a set of atomic opinions $A3$ which supports “XYZ90, shipping” with a negative sentiment, associated with the feature instance (*worldwide*, *generic_user*, *press_website* (i.e., most atomic arguments in $A3$ come from *worldwide*, provided by generic users from press websites).
- $\langle A4, \text{“}\mathsf{XYZ90,\ memory}\text{”}, neg, (US, expert\_user, press\_website)\rangle$, stating that there is a set of atomic opinions $A4$ which supports “XYZ90, memory” with a negative sentiment, associated with the feature instance ($US$, *expert_user*, *press_website*).
- $\langle A5, \text{“}\mathsf{XYZ90,\ memory,\ expansion}\text{”}, pos, (US, expert\_user, press\_website)\rangle$, stating that there is a set of atomic opinions $A5$ supporting “XYZ90, memory, expansion” with a positive sentiment, associated with the feature instance ($US$, *expert_user*, *press_website*).

Figure 2 illustrates the AO-arguments identified in this example.

### 3.2 Exploring Preference among AO-Arguments: Opinion trees

In the previous section we have shown how to express arguments for queries by associating them with a prevailing sentiment and a prevailing feature tuple. Such arguments might be attacked by other arguments, which on their turn might be attacked, too. In argumentation theory, this leads to the notion of *dialectical analysis* [7], which can be associated with a tree-like structure. This structure takes arguments, counter-arguments, counter-counter-arguments, and so on, into account. Our approach will be more generic, in the sense that for a given argument, the children nodes will correspond to *more preferred* arguments that are not necessarily in conflict with the parent argument. This preference can be defined in two ways: *query preference* or *feature preference*. In the first case, we will prefer those AO-arguments associated with more specific queries. The underlying idea is associated with the notion of *query subsumption*. Consider for example the queries “XYZ90”, “XYZ90 china”, and “XYZ90 beijing china”. When trying to find out stores for XYZ90 in Beijing, China (e.g. using a search engine), the third query is clearly more specific than the other two. Next we will formalize these notions.

**Definition 7 (Query Preference $\preceq_{\mathsf{q}}$).** *Given two queries $Q_1$ and $Q_2$, we will say that $Q_1$* is equivalent to *(resp.* subsumes*) $Q_2$ whenever $\mathsf{Agg}(Q_2, C) = \mathsf{Agg}(Q_1, C)$ (resp. $\mathsf{Agg}(Q_2, C) \subset \mathsf{Agg}(Q_1, C)$). We define a preference relation $\preceq_q$ for queries as follows: $Q_1 \preceq_q Q_2$ iff $Q_1$ is equivalent to $Q_2$ or $Q_1$ subsumes $Q_2$.*

**Definition 8 (Feature Preference $\preceq_{\mathsf{f}}$).** *Let $(\mathfrak{AOpinions}, C, \mathbb{S}, Sent, conflict, \mathbb{F}, PV)$ be an $OAF$. Given two feature tuples instances $ft_1^{\downarrow} = (v_1, v_2, \ldots, v_k)$ and $ft_2^{\downarrow} = (v'_1, v'_2, \ldots, v'_k)$, we will say that $ft_1^{\downarrow} \preceq_f ft_2^{\downarrow}$ iff $v_i \preceq v'_i$, $\forall i = 1 \ldots k$.*

*Example 1.* For the example in Section 3.1, it holds that
$(US, generic\_user, generic\_website) \preceq_{\mathsf{f}} (worldwide, generic\_user, press\_website)$.

**Definition 9 (Argument Preference using $\preceq_{\mathsf{q}}$, $\preceq_{\mathsf{f}}$ and $\preceq_{\mathsf{qf}}$).** *Given an OAF with AO-arguments $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ and $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$. We will say that:*

- $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *is* query-preferred *over* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$ *(denoted* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \preceq_{\mathsf{q}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$*) whenever* $Q_2 \preceq_{\mathsf{q}} Q_1$.
- $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *is* feature-preferred *over* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$ *whenever* $FV_2 \preceq_{\mathsf{f}} FV_1$.
- $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *is* preferred *over* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$ *(denoted* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \preceq_{\mathsf{qf}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *) whenever* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \preceq_{\mathsf{q}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *and* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \preceq_{\mathsf{f}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$.

**Definition 10 (Argument Minimal Strict Preference).** *Given an OAF with AO-arguments $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ and $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$. We we use $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \prec_{\mathsf{qf}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$ to specify that $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \preceq_{\mathsf{qf}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$, and it holds that $Q_1$ is not equivalent to $Q_2$ or $FV_1 \neq FV_2$. We say that $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ is minimally strictly preferred over $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$, and we denote it $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \prec^{\star}_{\mathsf{qf}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$ if $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \prec_{\mathsf{qf}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$ and there is no $\langle Arg_3, Q_3, Sent_3, FV_3\rangle$ such that $\langle Arg_2, Q_2, Sent_2, FV_2\rangle \prec_{\mathsf{qf}} \langle Arg_3, Q_3, Sent_3, FV_3\rangle \prec_{\mathsf{qf}} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$.*

Suppose that an AO-argument supporting the query "XYZ90" is obtained, with a prevailing sentiment *pos* and prevailing feature tuple ($US$, *generic_user*, *generic_website*). If the original query $Q$ is extended in some way into a new query $Q'$ that is more specific than $Q$ (i.e. $Q' = Q \cup \{d\}$), it could be the case that an AO-argument supporting $Q'$ has a different (possibly conflicting) prevailing sentiment. For example, more specific opinions about the table "XYZ90" could be related to other topics, like for example shipping quality, memory size, memory extension capabilities, etc. Additionally, another more specific AO-opinion about "XYZ90" can be identified by considering a more specific feature tuple (e.g. ($US$, *expert_user*, *press_website*). To explore all possible relationships associated with AO-arguments returned for a specified query $Q$ and criteria $C$, we can define an algorithm to construct an "opinion analysis tree" recursively as follows:

1. We start with an AO-argument $A$ obtained from the original query $Q$ (denoted as $\langle A, Q, Sent, FV\rangle$ ), which will be the root of the tree.
2. Next, we compute a set $\{AO_1, AO_2, \dots, AO_k\}$ of possible AO-arguments which are minimally strictly preferred over $\langle A, Q, Sent, FV\rangle$. Query-preferred arguments can be obtained by "extending" the original query $Q$ by adding a new element ($NewTerm$) to the query, obtaining $Q' = Q \cup \{NewTerm\}$. Feature-preferred arguments can be obtained by searching for feature values that are ranked higher than $FV$ (for instance, an expert user is ranked higher than a generic user).
3. For every AO-argument $AO_i$, a subtree will be recursively obtained, rooted as a child node of the original argument $\langle A, Q, Sent, FV\rangle$.

The high-level algorithm can be seen in Fig. 1. As stated before, note that our approach to opinion trees is more generic than the one used for dialectical trees

**ALGORITHM** *BuildOAT*
**INPUT:** $OAF$, $Q$, $DepthLevel$
**OUTPUT:** Opinion Analysis Tree $\mathsf{OAT}_{Q,C}$
{ *opinion analysis tree rooted in* $Root_{\mathsf{OAT}_Q}$ *according to* $C$ }
IF there exists $\langle Arg, Q, Sentiment, FV\rangle$ AND $DepthLevel > 1$ THEN
$Root_{\mathsf{OAT}_{Q,C}} := \langle Arg, Q, Sentiment, FV\rangle$,
$Children := \{ \langle A', Q', S', FV'\rangle \mid \langle Arg, Q, Sentiment, FV\rangle \prec^{\star}_{\mathsf{qf}} \langle A', Q', S', FV'\rangle \}$
IF $Children \neq \emptyset$
THEN
FOR EVERY $\langle A', Q', S', FV'\rangle \in Children$ DO
$\mathsf{OAT}_{Q'} := BuildOAT(OAF, Q', C, DepthLevel - 1)$
$PutSubtree(Root_{\mathsf{OAT}_{Q,C}}, \mathsf{OAT}_{Q',C})$
RETURN $\mathsf{OAT}_{Q,C}$

**Fig. 1.** High-level non-deterministic algorithm for computing $\mathsf{OAT}_{Q,C}$.

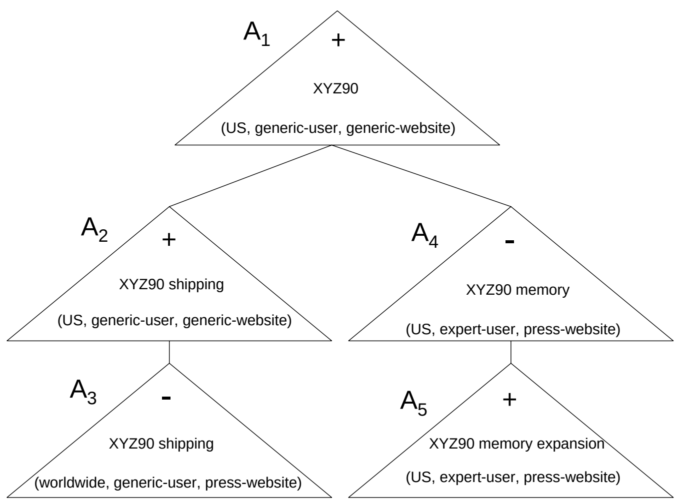


**Fig. 2.** Opinion analysis tree (Example from Section 3.1).

in argumentation (as done e.g. in [9]), in the sense that for a given argument, the children nodes will correspond to more specific arguments that are not necessarily in conflict with the parent argument. The algorithm in Fig. 1 finishes in finite time, as the maximum depth level is eventually reached. Additionally, branches cannot extend infinitely, as the query length is always finite (a finite set of terms), and as the set of feature values is always finite as well. Therefore the algorithm will eventually stop, providing an opinion analysis tree as an output. Fig. 2 illustrates the result of applying the *BuildOAT* algorithm to the scenario described in example 2.

## 4 Conflict trees

Next we will provide a formal definition of conflict between AO-arguments. Intuitively, a conflict will arise whenever two AO-arguments for similar queries lead to *conflicting sentiments* assuming that the involved queries are related to each other by the $\preceq_{\mathsf{qf}}$ relationship.

**Definition 11 (Argument Attack).** *Given an OAF with AO-arguments* $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *and* $\langle Arg_2, Q_2, Sent_2, FV_1\rangle$ *s.t.* $\langle Arg_2, Q_2, Sent_2, FV_1\rangle \preceq_{qf} \langle Arg_1, Q_1, Sent_1, FV_1\rangle$*, we will say that* $\langle Arg_2, Q_2, Sent_2, FV_2\rangle$ attacks $\langle Arg_1, Q_1, Sent_1, FV_1\rangle$ *if* $\langle Arg_1, Q_1, Sent_1, FV_1\rangle \prec^{\star}_{qf} \langle Arg_2, Q_2, Sent_2, FV_2\rangle$ *and* $Sent_1$ *and* $Sent_2$ *are in conflict.*

*Example 2.* Consider the AO-arguments in Section 3.1. Then it holds that:

- $\langle A3$, "XYZ90, shipping"$, neg, (worldwide, generic\_user, press\_website)\rangle$ attacks $\langle A2$, "XYZ90, shipping"$, pos, (US, generic\_user, generic\_website)\rangle$
- $\langle A4$, "XYZ90, memory"$, neg, (US, expert\_user, press\_website)\rangle$ attacks $\langle A1$, "XYZ90"$, pos, (US, generic\_user, generic\_website)\rangle$.

**Definition 12 (Sentiment-Preserving and Sentiment-Shifting Queries and Feature Instances).** *Given two AO-arguments* $\langle A_1, Q_1, Sent_1, FV_1\rangle$ *and* $\langle A_2, Q_2, Sent_2, FV_2\rangle$*,*

- $Q_2$ *is a* sentiment-preserving *(resp.* sentiment-shifting*) query wrt* $Q_1$ *whenever* $Sent_1$ *and* $Sent_2$ *are non-conflicting (resp. conflicting). Argument* $\langle A_2, Q_2, Sent_2, FV_2\rangle$ *will be called* sentiment-preserving *(resp.* sentiment-shifting *argument) wrt* $Q_1$*.*
- $FV_2$ *is a* feature-preserving *(resp.* feature-shifting*) feature instance iff it is the case that* $FV_2 \preceq_{\mathsf{f}} FV_1$ *and* $Sent_1$ *and* $Sent_2$ *are non-conflicting (resp. conflicting). Argument* $\langle A_2, Q_2, Sent_2, FV_2\rangle$ *will be called* feature-preserving *(resp.* feature-shifting *argument).*

*Example 3.* Consider the AO-arguments in Section 3.1. Then
$\langle A3$, "XYZ90, shipping"$, neg, (worldwide, generic\_user, press\_website)\rangle$
is a sentiment-shifting argument wrt
$\langle A2$, "XYZ90, shipping"$, pos, (US, generic\_user, generic\_website)\rangle$
$\langle A4$, "XYZ90, memory"$, neg, (US, expert\_user, press\_website)\rangle$
is a feature-preserving argument wrt
$\langle A1$, "XYZ90"$, pos, (US, generic\_user, generic\_website)\rangle$.

Given a particular query $Q$, note that several alternative expansions (supersets of $Q$) can be identified. We are interested in identifying which is the *smallest* superset of $Q$ along with the *minimum change in features* which is associated with a sentiment-shifting argument. This gives rise to the following definition:

**Definition 13 (Minimal-Shift Query / Minimal-Shift Feature).** *Given two conflicting AO-arguments* $\langle A_1, Q_1, Sent, FV_1\rangle$ *and* $\langle A_2, Q_2, \overline{Sent}, FV_2\rangle$*, we will say that*

- $Q_2$ *is a* minimal shift query *wrt* $Q_1$ *iff* $\not\exists Q' \subset Q_2$ *such that* $\langle A', Q', \overline{Sent}, FV' \rangle$ *is a sentiment-shifting argument wrt* $Q_1$.
- $FV_2$ *is a* minimal shift feature instance *wrt* $FV_1$ *iff* $\langle A_2, Q_2, \overline{Sent}, FV_2 \rangle$ *is a sentiment-shifting argument wrt* $\langle A_1, Q_1, Sent, FV_1 \rangle$ *and* $\not\exists FV' \preceq_f FV_2$ *such that* $\langle A', Q', \overline{Sent}, FV' \rangle$ *is a sentiment-shifting argument wrt* $\langle A_1, Q_1, Sent, FV_1 \rangle$.

*We define a* minimal-shifting query relation "$\preceq_{min}$" *as follows:* $\langle A_1, Q_1, Sent_1, FV_1 \rangle \preceq_{min} \langle A_2, Q_2, Sent_2, FV_2 \rangle$ *iff* $Q_2$ *is a minimal shift query wrt* $Q_1$ *and* $FV_2$ *is a minimal shift feature instance wrt* $FV_1$.

**Definition 14 (Conflict Opinion Analysis Tree).** *Given an* $OAF$*, a query* $Q$*, and its associated argument,* $\langle A, Q, Sent, FV \rangle$ *we will define a* conflict opinion analysis tree (COAT) *for* $Q$ *wrt* $C$ *(denoted* $COAT_{Q,C}$*) recursively as follows:*

1. *If there is no* $\langle A_i, Q_i, Sent_i, FV_i \rangle$ *such that* $\langle A, Q, Sent, FV \rangle \preceq_{min} \langle A_i, Q_i, Sent_i, , \rangle$ *then* $COAT_{Q,C}$ *is a conflict tree consisting of a single node* $\langle A, Q, Sent, FV \rangle$.
2. *Let* $\langle A_1, Q_1, Sent_1, FV_1 \rangle$, $\langle A_2, Q_2, Sent_2, FV_2 \rangle$, $\ldots \langle A_k, Q_k, Sent_k, FV_k \rangle$ *be those arguments in* $OAF$ *such that* $\langle A, Q, Sent, FV \rangle \preceq_{min} \langle A_i, Q_i, Sent_i, FV_i \rangle$ *(for* $i = 1 \ldots k$*). Then* $COAT_{Q,C}$ *is a conflict tree consisting of* $\langle A, Q, Sent, FV \rangle$ *as the root node and* $COAT_{Q_1,C}$, $\ldots COAT_{Q_k,C}$ *are its immediate subtrees.*

Intuitively, a conflict tree depicts all possible ways of computing new OA-arguments which correspond to a sentiment change wrt to the original argument at issue. This is performed recursively on those arguments, so that every node in the tree (except the root) is associated with an AO-argument which is a sentiment-shifting argument wrt its parent. Leaves correspond to nodes for which no further sentiment shift can be found.

Fig. 3 illustrates how the construction of a conflict opinion analysis tree for the worked example in Section 3.1 looks like, depicting nodes and arcs with dotted lines.

## 5 Discussion. Related work

In this paper, we have developed a new conceptualization for characterizing the notion of aggregated opinions and their interrelationships in an abstract way. For that, we departed from the approach originally used for Twitter in [8], extending it with several new concepts (notions of aggregated opinion, feature tuples, etc.) which show how to generalize that idea in a much more powerful and expressive setting. We have shown how a tree-like structure based on the integration of partial orders can be connected with important concepts in argumentation theory (attack, dialectical analysis), using sentiment shift as a way of identifying polarity change in arguments.

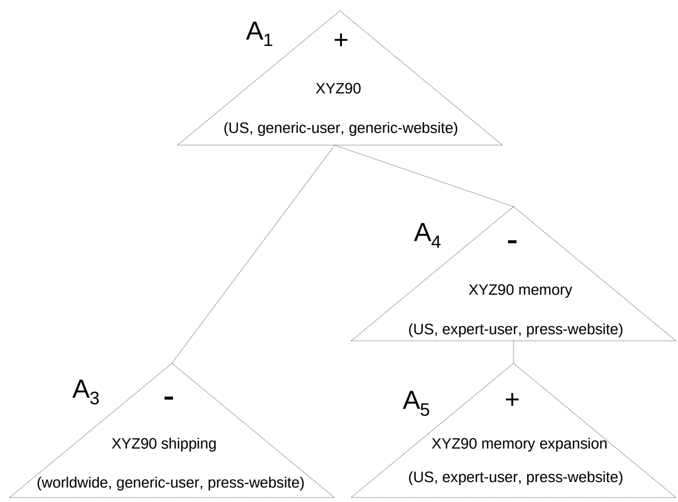


**Fig. 3.** Conflict opinion analysis tree (Example in Section 3.1).

As before, the notion of specificity plays a key role for contrasting arguments. However, in contrast with argument specificity (as a syntactic preference criterion in argumentation theory [10]) our approach is based on the preference ordering given by the $\preceq_{\mathsf{qf}}$ relationship. Thus, our approach to AO-arguments aims at modelling the possible space of alternatives associated with a universe of possible counter-arguments based on the aggregation of atomic opinions, structured according to a partial order which combines *topic specificity* and *feature specificity*. In contrast, traditional dialectical analysis in argumentation frameworks [6, 7] aims at determining the ultimate status of a given argument at issue (in terms of some acceptability semantics).

It must be remarked that the rise of social media such as blogs and social networks has fueled interest in sentiment analysis techniques [1]. With the proliferation of reviews, ratings, recommendations and other forms of online expression, online opinion has turned into a kind of virtual currency for businesses looking to market their products, identify new opportunities and manage their reputations.[4] In this setting, our approach is inspired by recent research in integrating argumentation and social networks, connected with recent contributions in sentiment analysis. To the best of our knowledge, Torroni & Toni [11] were the first that combined social networks and argumentation in a unified approach, coining the term *bottom-up argumentation* for the grass-root approach to the problem of deploying computational argumentation in online systems. In contrast with that proposal, in this paper we generalize this view by identifying

[4] The EU funded Cyberemotions consortium (see `http://www.cyberemotions.eu/`) was created in 2009 to better understand collective emotional phenomena in cyberspace, with the help of knowledge and methods from natural, social, and engineering sciences.

arguments automatically from atomic opinions in a more generic setting. In [12], Leite and Martins introduce a novel extension to Dung's abstract argumentation model, called Social Abstract Argumentation. Their proposal aims at providing a formal framework for social networks and argumentation, incorporating social voting and defining a new class of semantics for the resulting frameworks. In contrast with our approach, the automatic extraction of arguments from social networks data is not considered (as done in this paper), nor the modelling of conflicts between arguments in terms of sentiment analysis. In [13], Amgoud and Serrurier propose a formal argumentation-based model for classification, which generalizes the well-known concept learning model based on version spaces [14]. The framework shares some structural similarities with our approach. However, the aims of the two approaches are different, as our proposal is not focused on solving classification tasks in a machine learning sense.

## 6 Conclusions and Future Work

In this paper we have presented a novel approach which integrates argumentation theory, sentiment analysis and opinion mining. To the best of our knowledge, no other approach has been developed in a similar direction, and we think that our framework provides a valuable contribution for empowering sentiment analysis techniques [1] in an argumentative setting. As presented in the paper, the opinion mining process is characterized in terms of a dialectical analysis of aggregated opinions, according to a preference criterion given by topic and feature specificity. The resulting analysis can be depicted as a tree-like structure, similar to dialectical trees in argumentation frameworks [7].

Based on a previous Java implementation for analyzing the Twitter microblogging platform [8], we are currently implementing a prototype of our proposal as a proof of concept. Clearly, atomic opinions to be handled in this specification are richer than tweets (as done in [8]), which requires to focus search on particular text-based collections of information items (Goolge snipets, Amazon book reviews, etc.). We are also working the analysis of the usability features involved in the specification, in order to provide a suitable interface through which the user can define features to be considered in an interactive way.

Part of our future work is associated with deploying the ideas presented in this paper in a software product. As a basis for such deployment, visual tools for displaying and analyzing dialectical trees have been already developed for Defeasible Logic Programming [15]. As for the Twitter case, we expect to use the underlying algorithms from this tool in our framework, performing as well different experiments for assessing the ultimate applicability of our proposal. Research in this direction is currently being pursued. Another future research avenue is to explore not only contradictory opinions but also ambivalent or uncertain ones. For instance, it would be interesting to identify cases in which participants list positive and negative aspects about an issue but do not take a final positive or negative position. In these cases, in addition to the proposed bottom-up approach, it may also be useful to implement a top-down process, providing positive or negative feedback on the atomic opinions.

**Acknowledgments:** We thank the reviewers for their comments which helped improve the original version of this paper. This research is funded by Projects LACCIR R1211LAC004 (Microsoft Research, CONACyT and IDB), PIP 112-200801-02798, PIP 112-200901-00863 (CONICET, Argentina), PGI 24/ZN10, PGI 24/N006, PGI 24/N029 (SGCyT, UNS, Argentina) and Universidad Nacional del Sur.

## References

1. Feldman, R.: Techniques and applications for sentiment analysis. Commun. ACM **56**(4) (April 2013) 82–89
2. Martineau, J.: Identifying and Isolating Text Classification Signals from Domain and Genre Noise for Sentiment Analysis. PhD thesis, University of Maryland, Baltimore County, USA (2011)
3. Cao, Q., Thompson, M.A., Yu, Y.: Sentiment analysis in decision sciences research: An illustration to IT governance. Decision Support Systems **54**(2) (2013) 1010–1015
4. Chesñevar, C.I., Maguitman, A.G., Estevez, E., Brena, R.F.: Integrating argumentation technologies and context-based search for intelligent processing of citizens' opinion in social media. In Ferriero, D., Pardo, T.A., Qian, H., eds.: ICEGOV, ACM (2012) 166–170
5. Mizumoto, K., Yanagimoto, H., Yoshioka, M.: Sentiment analysis of stock market news with semi-supervised learning. In Miao, H., Lee, R.Y., Zeng, H., Baik, J., eds.: ACIS-ICIS, IEEE (2012) 325–328
6. Besnard, P., Hunter, A.: The Elements of Argumentation. The MIT Press. London, UK (2008)
7. Rahwan, I., Simari, G. In: Argumentation in Artificial Intelligence. Springer Verlag (2009)
8. Grosse, K., Chesñevar, C.I., Maguitman, A.G.: An argument-based approach to mining opinions from twitter. In Ossowski, S., Toni, F., Vouros, G.A., eds.: AT. Volume 918 of CEUR Workshop Proceedings., CEUR-WS.org (2012) 408–422
9. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. TPLP **4**(1-2) (2004) 95–138
10. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Comput. Surv. **32**(4) (2000) 337–383
11. Torroni, P., Toni, F.: Bottom up argumentation. In: Prof. of First Intl. Workshop on Theoretical and Formal Argumentation (TAFA). IJCAI 2011, Barcelona, Spain. (2011)
12. Leite, J., Martins, J.: Social abstract argumentation. In Walsh, T., ed.: IJCAI, IJCAI/AAAI (2011) 2287–2292
13. Amgoud, L., Serrurier, M.: Agents that argue and explain classifications. Autonomous Agents and Multi-Agent Systems **16**(2) (2008) 187–209
14. Mitchell, T.M.: Generalization as search. Artif. Intell. **18**(2) (1982) 203–226
15. Modgil, S., Toni, F., Bex, F., Bratko, I., Chesñevar, C., Dvořák, W., Falappa, M.A., Gaggl, S.A., García, A.J., Gonzalez, M.P., Gordon, T.F., Leite, J., Mozina, M., Reed, C., Simari, G.R., Szeider, S., Torroni, P., Woltran, S.: The Added Value of Argumentation: Examples and Challenges. In: Handbook of Agreement Technologies. Springer (2013) 357–404