An Argument-based Approach to Mining Opinions from Twitter[*]

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

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

Abstract. Social networks have grown exponentially in use and impact on the society as a whole. In particular, microblogging platforms such as Twitter have become important tools to assess public opinion on different issues. Recently, some approaches for assessing Twitter messages have been developed, identifying sentiments associated with relevant keywords or hashtags. However, such approaches have an important limitation, as they do not take into account contradictory and potentially inconsistent information which might emerge from relevant messages. We contend that the information made available in Twitter can be useful to extract a particular version of arguments (called "opinions" in our formalization) which emerge bottom-up from the social interaction associated with such messages. In this paper we present a framework which allows to mine opinions from Twitter based on incrementally generated queries. As a result, we will be able to obtain an "opinion tree", rooted in the first original query. Distinguished, conflicting elements in an opinion tree lead to so-called "conflict trees", which resemble dialectical trees as those used traditionally in defeasible argumentation.

1 Introduction and motivations

Social networks have grown exponentially in use and impact on the society as a whole, aiming at different communities and providing differentiated services. In particular, microblogging has become a very popular communication tool among Internet users, being Twitter[1] by far the most widespread microblogging platform. Twitter, created in 2006, enables its users to send and read text-based posts of up to 140 characters, known as "tweets". It has grown into a technology

---

[***] First author is a student carrying out a research stance at the Department of Computer Science and Engineering of the Universidad Nacional del Sur in Bahía Blanca, Argentina.
[*] AT2012, 15-16 October 2012, Dubrovnik, Croatia. Copyright held by the author(s).
[1] www.twitter.com

which allows to assess public opinion on different issues. Thus, for example, it is common to read nowadays newspapers articles referring to the impact of political decisions measured by their associated positive or negative comments in Twitter. Symmetrically, policy makers make public many of their claims and opinions, having an influence on the citizenry,[2] prompting their "tweeting back" with further comments and opinions. As the audience of microblogging platforms and services grows everyday, data from these sources can be used in opinion mining and sentiment analysis tasks [1].

As pointed out in [2], microblogging platforms (in particular Twitter) offer a number of advantages for opinion mining. On the one hand, Twitter is used by different people to express their opinion about different topics, and thus they are a valuable source of people's opinions. Given the enormous number of text posts, the collected corpus can be arbitrarily large. On the other hand, Twitter's audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups. According to Merriam Webster online dictionary,[3] 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. Clearly, there is a natural link between opinion and argument. In many cases, opinions by themselves do not provide arguments, as they do not necessarily imply giving reasons or evidence for accepting a particular conclusion. However, from a meta-level perspective, policy makers devote much effort in analyzing the reasons underlying complex collections of opinions from the citizenry, as they indicate the willingness of the people to accept or reject some particular issue. A well-known example in this setting is the analysis of public opinion (e.g. through the quantitative measurement of opinion distributions through polls and the investigation of the internal relationships among the individual opinions that make up public opinion on an issue).

A fundamental need for policy makers is to back their decisions and agreements on reasons or opinions provided by citizens. They might even argue with other policy makers about why making a particular decision is advisable (e.g. "according to the last poll, 80% of the people are against the health system reform; therefore, the reform should not be carried out"). From this perspective, social networks like Twitter provide a fabulous knowledge base from which information could be collected and analyzed in order to enhance and partially automatize decision making processes. In particular, tweets have a rich structure, providing a number of record fields which allow to detect provenance of the tweet (author), number of re-tweets, followers, etc.

---

[2] E.g. the current UK Prime Minster David Cameron and the current US President Barack Obama can be followed on Twitter at `@Number10gov` and `@BarackObama`, respectively.

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

We contend that the information made available from such tweets can be useful for modeling opinions which emerge bottom-up from the social interaction existing in Twitter. In our analysis, we will assume that opinions *are* arguments, which can be seen as particular instances of the "Argument from Majority" schema [3]. In this paper, we analyze the main elements characterizing an argument-based approach to mining opinions from Twitter based on incrementally generated queries. Given a query, we will model an opinion supporting it as a set of aggregated tweets along with a prevailing sentiment,[4] which can be attacked by alternative counter-opinions. As a final result, we will be able to obtain a "conflict tree", rooted in the first original query, in a way that resembles dialectical trees in argumentation.

The rest of the paper is structured as follows. In Section 2 we present our proposal for characterizing Twitter-based arguments and their interrelationships. We will formalize the notion of *opinion tree*, which can be constructed from user queries, allowing to assess alternative opinions associated with incrementally generated queries. A high-level algorithm for computing opinion trees is presented, along with a case study to illustrate our proposal. Section 3 discusses the relationship emerging from opinions in conflict, modeled as a conflict tree. Section 4 generalizes previous results using superior lattices, for both opinion and conflict trees. The benefits of applying this mathematical approach are discussed in Section 5. Section 6 reviews related work, and finally Section 7 summarizes the conclusions.

2 Twitter-based Argumentation Framework: viewing aggregated tweets as arguments

In this Section we will describe how different elements in Twitter can be captured under an argumentative perspective [4, 5]. First we will characterize distinguished collections of tweets (obtained on the basis of a given query) as arguments with an associated prevailing sentiment. Such arguments will be called TB-arguments (Twitter-based arguments). Then, we will formalize interrelationships between TB-arguments, which lead to the notion of *opinion tree*.

2.1 Formalizing Aggregation of Twitter Messages

Twitter messages (Tweets) are 140 character long, with a number of additional fields which help identify relevant information within a message (sender, number of retweets associated with the message, etc.). In particular, we will focus on the presence of *descriptors* which are either *hashtags* (words or phrases prefixed with the symbol #, a form of metadata tag) or terms that tend to occur often in the context of a given topic. Hashtags are used within IRC networks to identify groups and topics and in short messages on microblogging social networking

---

[4] Several software tools have been recently developed for such an association, such as `www.sentiment140.com` or `tweetsentiments.com`.

services such as Twitter, identi.ca or Google+ (which may be tagged by including one or more with multiple words concatenated). Other good descriptors can be dynamically found by looking for terms that are frequently used in tweets related to the topic at hand. In the sequel we will assume that the term "descriptor" refers to either actual hashtags in Twitter or to relevant keywords found in tweets.

Definition 1 (Tweet. Twitter Query). *We define a tweet T as a bag (or multiset) of terms $\{t_1, t_2, \ldots t_k \}$, where every $t_i \in T$ is a string. A Twitter query (or just query) is a non-empty set $Q = \{d_1, d_2, \ldots, d_k\}$ of descriptors, where every $d_i \in Q$ is a string.*

In the analysis that follows, we will assume that a tweet is just a bag of words, not taking into account the actual order of terms in the tweet. Additionally, we assume that the set of all currently existing tweets corresponds to a snapshot of Twitter messages at a given fixed time, as the Twitter database is highly dynamic. In our approach, a query Q is any set of descriptors used for *filtering* some relevant tweets from $\mathfrak{Tweets}$ based on a given criterion C. In order to abstract away how such selection is performed, we will define an aggregation operator $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$. Formally:

Definition 2 (Tweet set. Aggregation Operator). *Let $\mathfrak{Tweets}$ be the set of all currently existing tweets. We will write $2^{\mathfrak{Tweets}}$ to denote the set of all possible subsets of $\mathfrak{Tweets}$. Any element in $2^{\mathfrak{Tweets}}$ will be called a tweet set. Given a query Q, and a criterion C, we will define an aggregation operator $Agg_{\mathfrak{Tweets}}(Q, C)$ which returns an element (tweet set) in $2^{\mathfrak{Tweets}}$ based on Q and C.*

The aggregation operator could be defined in several ways. For instance, suppose that C_1 is a criterion that indicates that only tweets posted between time $timestamp_1$ and $timestamp_2$ are to be selected. Then $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C_1)$ $=_{def} \{ T \in \mathfrak{Tweets}$ such that $Q \subseteq T$ and T satisfies $C_1 \}$ will be the set of tweets that contain all the terms of query Q and have been posted in the time period $[timestamp_1, timestamp_2]$. Other examples of criteria that can be naturally applied are, for instance, requiring that those tweets T were retweeted more than n times, requiring that every user that posted tweets T has at least m followers, etc.

Note that for the same query Q, different alternative criteria $(C_1, C_2, \ldots, C_k)$ can lead to different distinguished elements in $2^{\mathfrak{Tweets}}$. As explained before, tweet sets can be associated with different feelings or sentiments. Even if in real life there may be a lot of emotions in tweets (such as anger, happiness, and so on), we will assume here that there is a distinguished set $\mathbb{S}$ of possible sentiments. Thus, given a query Q and a criterion C, we assume that the tweet set $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ is associated with a prevailing sentiment in $\mathbb{S}$.[5] We will consider that some

---

[5] A possible range for $\mathbb{S}$ could be positive, negative and neutral (as done for example in platform $\mathsf{Sentiment140.com}$) In this platform, prevailing sentiments associated with a tweet set are expressed by percentages.

sentiments might convey different, possibly *conflicting* feelings or emotions (e.g. anger and happiness; boredom and excitement, etc.). As before, we will abstract away which are potentially conflicting sentiments as follows.

Definition 3 (*Sent* **and** *conflict* **mappings**). *Let* $\mathbf{T} \in 2^{\mathfrak{Tweets}}$ *be a tweet set, and let Sent* $: 2^{\mathfrak{Tweets}} \rightarrow \mathbb{S}$ *and conflict* $: \mathbb{S} \rightarrow 2^{\mathbb{S}}$ *be mappings. The sentiment Sent*$(\mathbf{T})$ *will be called the prevailing sentiment (or just sentiment) for* $\mathbf{T}$. *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* $\bar{s}$ *to denote any* $s' \in conflict(s)$.

The previous elements will allow us to characterize the notion of TB-framework and TB-argument as follows:

Definition 4 (TB-framework). *A* Twitter-based argumentation framework *(or* TB-framework*) is a 5-tuple* $(\mathfrak{Tweets}, C, \mathbb{S}, Sent, conflict)$, *where* $\mathfrak{Tweets}$ *is the set of available tweets,* C *is a selection criterion,* $\mathbb{S}$ *is a non-empty set of possible sentiments and Sent and conflict are sentiment prevailing and conflict mappings.*

Definition 5 (TB-argument). *Let* $(\mathfrak{Tweets}, C, \mathbb{S}, Sent, conflict)$ *be a TB-framework. A* Twitter-based argument *(or* TB-argument*) for a conclusion (query)* Q *is a 3-tuple* $\langle Arg, Q, Sentiment \rangle$, *where* Arg *is* $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ *and Sentiment is* $Sent(\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C))$.

Example 1. Consider a TB-framework $(\mathfrak{Tweets}, C, \mathbb{S}, Sent, conflict)$, where C is defined as "all $T \in \mathfrak{Tweets} \mid timestamp(T) \geq$ 2012-01-01T00:00:00", $Q =$ {"#greece", "#eurozone"}, $\mathbb{S} = \{pos,\ neg,\ neutral\}$, and $conflict(pos) =_{def}$ $\{neg, neutral\}$, $conflict(neg) =_{def} \{pos, neutral\}$ and $conflict(neutral) =_{def}$ $\{pos, neg\}$. Then $Arg = \mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ is the set of all possible tweets containing {"#greece","#eurozone"} that have been published since January 1, 2012. Suppose that $Sent(\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)) = negative$. Then $\langle Arg, \{$"#greece", "#eurozone"$\}, negative \rangle$ is a TB-argument.

2.2 Specificity in a TB-framework. Opinion trees

In the previous section we have shown how to express arguments for queries associated with a given prevailing sentiment. 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* [5], which can be associated with a tree-like structure in which arguments, counter-arguments, counter-counter-arguments, and so on, are taken into account. Our approach will be more generic, 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. Next we will formalize these notions.

Definition 6 (Query Equivalence). *Let* $(\mathfrak{Args}, \mathfrak{Tweets}, C, \mathbb{S}, s)$ *be a TB-framework. Given two queries* Q_1 *and* Q_2*, we will say that* Q_1 *is equivalent to* Q_2 *whenever* $\mathsf{Agg}_{\mathfrak{Tweets}}(Q_2, C) = \mathsf{Agg}_{\mathfrak{Tweets}}(Q_1, C)$*.*

Definition 7 (Query Subsumption). *Let* $(\mathfrak{Args}, \mathfrak{Tweets}, C, \mathbb{S}, s)$ *be a TB-framework. Given two queries* Q_1 *and* Q_2*, we will say that* Q_1 *subsumes* Q_2 *whenever it holds that* $\mathsf{Agg}_{\mathfrak{Tweets}}(Q_2, C) \subset \mathsf{Agg}_{\mathfrak{Tweets}}(Q_1, C)$*.*

Example 2. A query Q_1 formed by $\{$"#greece"$\}$ subsumes the query Q_2 formed by $\{$"#greece", "#eurozone"$\}$, as all the tweets that are returned by Q_2 will be part of the tweets returned by Q_1, but not the other way around.

Note that the *subsumption* relation is more general than the *inclusion* relation, since Q_1 *subsumes* Q_2 whenever $Q_1 \subset Q_2$ (as $\mathsf{Agg}_{\mathfrak{Tweets}}(Q_2, C) \subset \mathsf{Agg}_{\mathfrak{Tweets}}(Q_1, C)$). However, it is possible that Q_1 *subsumes* Q_2 even when $Q_1 \not\subset Q_2$.

Definition 8 (Argument Specificity). *Given a TB-framework with TB-arguments* $\langle Arg_1, Q_1, Sent_1 \rangle$ *and* $\langle Arg_2, Q_2, Sent_2 \rangle$*, we say that* $\langle Arg_2, Q_2, Sent_2 \rangle$ *is* strictly more specific *than* $\langle Arg_1, Q_1, Sent_1 \rangle$ *and we denote it* $\langle Arg_2, Q_2, Sent_2 \rangle \prec \langle Arg_1, Q_1, Sent_1 \rangle$ *if* Q_1 *subsumes* Q_2*. We will write* $\langle Arg_2, Q_2, Sent_2 \rangle \preceq \langle Arg_1, Q_1, Sent_1 \rangle$ *when* Q_1 *subsumes* Q_2 *or* Q_1 *is equivalent to* Q_2*.*

Suppose that a TB-argument supporting the query "#greece" is obtained, with a prevailing sentiment *neutral*. 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 a TB-argument supporting Q' has a different (possibly conflicting) prevailing sentiment. For example, more specific opinions about Greece are related to other topics, like for example vacations, politics, philosophy, etc. To explore all possible relationships associated with TB-argument returned for a specified query Q and criteria C, we can define an algorithm to construct an "opinion tree" recursively as follows:

1. We start with a TB-argument obtained from the original query Q (i.e., $\langle A, Q, Sent \rangle$), which will be the root of the tree.
2. Next, we compute within A all relevant descriptors that might be used to "extend" Q, by adding a new element ($NewTerm$) to the query, obtaining $Q' = Q \cup \{NewTerm\}$.
3. Then, a new argument for Q' is obtained, which will be associated with a subtree rooted in the original argument A.

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 in argumentation (as done e.g. in [6]), 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.

It is also easy to see that for any query Q, the algorithm *BuildOT* finishes in finite time: given that a tweet may not contain more than 140 characters, the number of contained descriptors is finite, and therefore the algorithm will eventually stop, providing an opinion tree as an output

ALGORITHM *BuildOT*
INPUT: $\mathfrak{Tweets}$, $\mathbf{R}$, Q, C
      { *Initially,* $\mathbf{R} = \mathfrak{Tweets}$. *Each* $T_i \in \mathfrak{Tweets}$ *is represented as a multiset* }
OUTPUT: Opinion Tree OT_Q
      { *opinion tree rooted in* $Root_{\mathsf{OT}_Q}$ }
$Root_{\mathsf{OT}_Q} := \langle Arg, Q, Sentiment \rangle$,
where Arg is $\mathsf{Agg}_{\mathfrak{Tweets}}(Q,C)$ and $Sentiment$ is $Sent(\mathsf{Agg}_{\mathfrak{Tweets}}(Q,C))$.
IF $|\mathbf{R}| >$ threshold { *Cardinality of* $\mathbf{R}$ *determines maximum depth level* }
THEN
  REPEAT
    $W := \{\ d \mid d$ is the most frequent word in $\biguplus_{T_i \in \mathbf{R}} T_i$
        such that $d \notin Q \cup Stopwords\ \}^a$
   IF $W \neq \emptyset$
   THEN
    $Q_{new} := Q \cup W$
    $Tweets_{Q_{New}} := \mathsf{Agg}_{\mathfrak{Tweets}}(Q_{new}, C)$
    $\mathsf{OT}_{Q_{new}} := BuildOT(Tweets_{Q_{New}}, \mathbf{R}, Q_{new}, C)$
    $PutSubtree(Root_{\mathsf{OT}_Q}, \mathsf{OT}_{Q_{new}})$
    $\mathbf{R} := \mathbf{R} - Tweets_{Q_{New}}$
 UNTIL $W = \emptyset$
  RETURN OT_Q

---

[a] For simplicity, we assume W is a singleton. In a more general case, a distinguished element from W could be selected according to some criterion (e.g. overall frequency in Twitter, etc.). The operation $\biguplus$ is the union operation on multisets.

Fig. 1. High-level algorithm for computing an opinion trees OT_Q from Twitter

2.3 Case study

As discussed before, the algorithm shown in Fig. 1 allows to obtain an opinion tree from a given query Q, a criterion C, and the set $\mathfrak{Tweets}$ of all possible tweets. An additional parameter $\mathbf{R}$ allows us to specify the set of tweets to be considered when searching for a new descriptor. Initially, $\mathbf{R} = \mathfrak{Tweets}$. The cardinality of $\mathbf{R}$ determines the threshold associated with the depth of the tree.

Consider the query $Q = $ "#greece", and a criterion $C = \{\ T \in \mathfrak{Tweets} \mid T$ is among the 1000 most recent Tweets $\}$. A root TB-argument is computed for Q, C and $\mathfrak{Tweets}$, obtaining an associated prevailing sentiment (*neutral*). If $|\mathbf{R}|$ is above a given threshold value, the algorithm computes the most frequent word d in $\mathbf{R}$ whenever d is not already present in $Q \cup Stopwords$. The underlying idea is that any sibling node in the same level of the tree refers to terms *not appearing already in previous siblings* (from left to right). The set *Stopwords* will usually include terms such as *the*, *as*, *which*, etc. In our example, $d = $ "coalition". A new TB-argument can now be built for query $Q' = \{$"#greece"$\} \cup \{$"coalition"$\}$, criterion C and the preserving sentiment calculated for the new subset of tweets $Tweets_{Q_{New}}$. In the recursive call, the most frequent word is calculated for this subset (obtaining the result "democratic"), so that a new TB-argument for the query $\{$"#greece", "coalition", "democratic"$\}$ is obtained, with a new associated
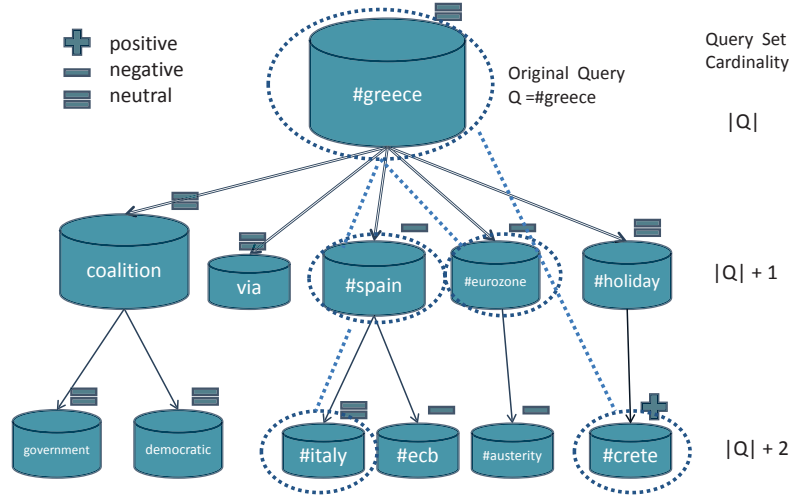
Fig. 2. Opinion Tree based on query "#greece" (computation based on status of Twitter as present at June 26, 2012). The associated conflict tree for the same query is shown in dotted lines.

prevailing sentiment. Note that within a particular instance of the recursive call, the REPEAT loop takes care that alternative ways of "extending" Q are considered by adding a particular descriptor d, not repeating d within the sibling nodes at a given level of the tree. The process is repeated until the threshold has been reached. At the end, the resulting opinion tree OT_Q is returned.

Figure 2 illustrates how the construction of an opinion tree for the query $Q = \{$"#greece"$\}$ looks like. Distinguished symbols ("+", "-", "=") are used to denote positive, negative and neutral sentiments, respectively. Note that the original query Q has cardinality 1, and further levels in the opinion tree refer to incrementally extended queries (e.g. $\{$"#greece", "coalition"$\}$, or $\{$"#greece", "#spain"$\}$). Leaves correspond to arguments associated with a query Q' which cannot be further expanded, as the associated number of tweets is too small for any possible query $Q' \cup W$. Furthermore, we can identify some subtrees in $\mathsf{OT}_{\{$"#greece"$\}}$ which consist of nodes which have *all the same sentiment*. In other words, further expanding a query into more complex queries does not change the prevailing sentiment associated with the root node. In other cases, expanding some queries results in a sentiment change (e.g. from $\{$"#greece"$\}$ into $\{$"#greece", "#spain"$\}$). This situation will allow us to characterize *conflict trees*, in which we take into account opinions that attack each other, as discussed in the next Section.

3 Conflict trees

Next we will provide a formal definition of conflict between TB-arguments. Intuitively, a conflict will arise whenever two arguments for similar queries lead to conflicting sentiments assuming that the involved queries are related to each other by the subsumption relationship.

Definition 9 (Argument Attack). *Given a TB-framework with TB-arguments $\langle Arg_1, Q_1, Sent_1 \rangle$ and $\langle Arg_2, Q_2, Sent_2 \rangle$ such that Q_1 subsumes Q_2, we say that $\langle Arg_2, Q_2, Sent_2 \rangle$ attacks $\langle Arg_1, Q_1, Sent_1 \rangle$ whenever $Sent_1$ and $Sent_2$ are in conflict.*

Example 3. Consider two queries $Q_1 = \{$"#greece"$\}$ and $Q_2 = \{$"#greece", "#eurozone"$\}$, such that $\langle Arg_1, Q_1, neutral \rangle$ and $\langle Arg_2, Q_2, negative \rangle$. Then $\langle Arg_2, Q_2, negative \rangle$ attacks $\langle Arg_1, Q_1, neutral \rangle$.

Note that in the previous situation, adding the descriptor "eurozone" to the the original query "#greece" involves a sentiment change. We will formalize this situation as follows:

Definition 10 (Sentiment-Preserving and Sentiment-Shifting Descriptor). *Given an argument $\langle A_1, Q, Sent_1 \rangle$, d is a* sentiment-preserving *(resp.* sentiment-preserving*) descriptor wrt Q whenever there exists an argument $\langle A_2, Q \cup \{d\}, Sent_2 \rangle$ such that $Sent_1$ and $Sent_2$ are non-conflicting (resp. conflicting). Argument $\langle A_2, Q \cup \{d\}, Sent_2 \rangle$ will be called* sentiment-preserving *(resp.* sentiment-shifting *argument).*

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

Definition 11 (Minimal-Shift Descriptor. Minimal-Shifting Relation). *Let $(\mathfrak{Args}, \mathfrak{Tweets}, C, \mathbb{S}, s)$ be a TB-framework. Given two conflicting arguments $\langle A_1, Q_1, Sent \rangle$ and $\langle A_2, Q_2, \overline{Sent} \rangle$, we will say that Q_2 is a* minimal shift descriptor *wrt Q_1 iff $\langle A_2, Q_2, \overline{Sent} \rangle$ is a sentiment-shifting argument wrt Q_1 and $\nexists Q' \subset Q_2$ such that $\langle A', Q', \overline{Sent} \rangle$ is a sentiment-shifting argument wrt Q_1.*
We define a minimal-shifting relation *"$\preceq_{min}$" as follows: $\langle A_1, Q_1, Sent_1 \rangle \preceq_{min} \langle A_2, Q_2, Sent_2 \rangle$ iff $\langle A_2, Q_2, Sent_2 \rangle$ attacks $\langle A_1, Q_1, Sent_1 \rangle$ and Q_2 is a minimal-shifting descriptor wrt Q_1.*

Definition 12 (Conflict tree). *Let $(\mathfrak{Args}, \mathfrak{Tweets}, C, \mathbb{S}, s)$ be a TB-framework. Given a query Q, and its associated argument $\langle A, Q, Sent \rangle$ we will define a* conflict tree *for Q (denoted $C(Q)$) recursively as follows:*

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

Intuitively, a conflict tree depicts all possible ways of extending the original query Q such that every extension (child node in the tree) corresponds to a sentiment change. Figure 2 illustrates how the construction of a conflict tree for

the query $Q = \{$"#greece"$\}$ looks like, depicting nodes and arcs with dotted lines. Every node in the tree (except the root) is associated with a TB-argument which is a sentiment-shifting argument wrt its parent. Leaves correspond to nodes for which no further sentiment shift can be found.

4 Generalizing Opinion and Conflict Trees as Superior Lattices

Next we will show a formal lattice-based characterization of our approach. Superior lattices will account for a more generic view of opinion trees, identifying relevant sublattices based on an equivalence relation between TB-arguments. First we will review some background definitions to make our presentation self-contained.

Definition 13 (Partial Order. Partially Ordered Set). *A* partial order *is a binary relation "$\preceq$" over a set A which is reflexive, antisymmetric, and transitive, i.e., for all a, b, and c in A, we have that (1) $a \preceq a$ (reflexivity); if $a \preceq b$ and $b \preceq a$ then $a = b$ (antisymmetry); if $a \preceq b$ and $b \preceq c$ then $a \preceq c$ (transitivity). A set with a partial order is called* partially ordered set *(or just ordered set).*

Definition 14 (Cover Relation). *Given an ordered set $(A, \preceq)$, for two elements $a, b \in A$ we use $a \prec b$ to specify that $a \preceq b$ and $a \neq b$. Let $(A, \preceq)$ be an ordered set. Then for any $a, b \in A$ we say that a covers b if $b \prec a$ and there is no $c \in A$ such that $b \prec c \prec a$.*

Definition 15 (Tree Order). *An ordered set $(A, \preceq)$ is a tree if (1) there is a unique $a \in A$ such that $b \preceq a$ for all $b \in A$, and (2) for all $a, b, c \in A$, if b covers a and c covers a, then $b = c$.*

Definition 16 (Superior Lattice. Inferior Lattice. Lattice). *Let $(A, \preceq)$ be an ordered set. Then for any $a, b \in A$, we will say that $c \in A$ is the* least upper bound *of a and b (also called the* join *of a and b), denoted $c = a \vee b$, whenever i) $a \preceq c$ and $b \preceq c$; ii) if for $x \in A$, it holds that $x \preceq a$ and $x \preceq b$, then $c \preceq x$. An ordered set $(A, \preceq)$ is a* superior lattice *whenever for any pair of elements $a, b \in A$ there is a least upper bound element in A. The notions of* greatest lower bound *(or* meet*), denoted $c = a \wedge b$, and* inferior lattice *are defined analogously as the duals of the notions of* least upper bound *and* superior lattice. *An ordered set $(A, \preceq)$ that is both a* superior lattice *and an* inferior lattice *is called a* lattice.

Definition 17 (Join-Homomorphism. Meet-Homomorphism. Lattice Homomorphism). *The mapping h from $(X, \preceq)$ to $(Y, \preceq)$ is a* join-homomorphism *provided that for any $a, b \in x$, $h(a \vee b) = h(a) \vee h(b)$. It is also said that "$h$ preserves joins." The notion of* meet-homomorphism *is defined analogously as the dual of the notion of* join-homomorphism. *The mapping h is a* lattice homomorphism *if it is both a join-homomorphism and a meet-homomorphism.*
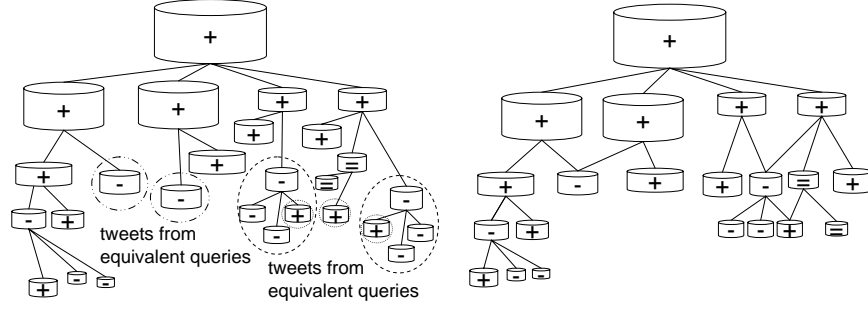
Fig. 3. Opinion tree (left) and its corresponding opinion lattice (right).

In the rest of this section we will show that the above definitions provide a solid mathematical foundation for the study of TB-arguments. Note, in the first place, that the ordered set $(2^{\mathfrak{Tweets}}, \subseteq)$ is a lattice, as is the case for any power set of a given set, ordered by inclusion. The join is given by the union and the meet by the intersection of the subsets. More interestingly, it can be shown that for any query Q, the resulting opinion tree OT_Q associated with a query Q defines a tree order (see Def. 15).

Lemma 1. *Let Q be a query and let OT_Q be an opinion tree for Q in a TB-framework $(\mathfrak{Args}, \mathfrak{Tweets}, C, \mathbb{S}, s)$. Then $(\mathsf{OT}_Q, \preceq)$ defines a tree order.*[6]

On the left-hand side of Figure 3 we illustrate an opinion tree as a tree order $(\mathsf{OT}_Q, \preceq)$. Note that each element in OT_Q is of the form $\langle Arg_i, Q_i, Sentiment_i \rangle$, while the order relation "$\preceq$" is defined as $\langle Arg_1, Q_1, Sentiment_1 \rangle \preceq \langle Arg_2, Q_2, Sentiment_2 \rangle$ if and only if $Q_2 \subseteq Q_1$. In this opinion tree we have indicated that some queries are equivalent (see Def. 6). As a consequence, we can identify a quotient set, where each member is an equivalence class $[\langle Arg_i, Q_i, Sentiment_i \rangle]$ defined as follows:

$$[\langle Arg_i, Q_i, Sentiment_i \rangle] = \{\langle Arg_j, Q_j, Sentiment_j \rangle \mid Q_j \text{ is equivalent to } Q_i\}.$$

We show on the right-hand side of Figure 3 the quotient set resulting from the given opinion tree. Note that this new set is a superior lattice (see Def. 16). In general, any opinion tree induces a superior lattice OL_Q, which we will refer to as opinion lattice. This is formally stated in the following lemma:

Lemma 2. *Let Q be a query and let OL_Q be the quotient set of OT_Q by the query equivalence relation. Then $(\mathsf{OL}_Q, \preceq)$ is a superior lattice.*

Although an opinion lattice is typically more compact than an opinion tree, we might be interested in finding the minimal structure that reflects all existing

---

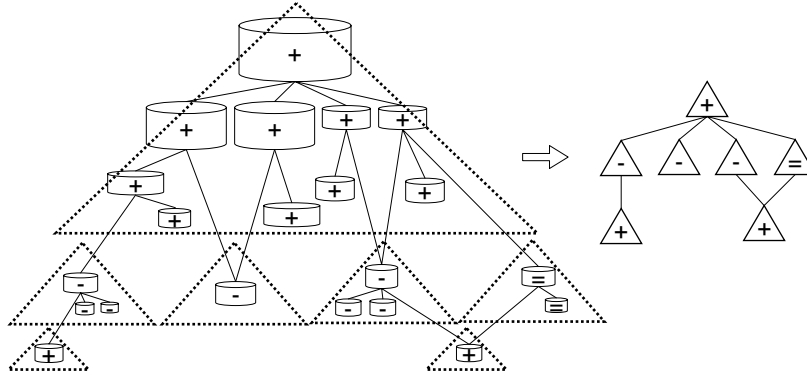[6] Proofs not included for space reasons.

Fig. 4. From an opinion lattice (left) to a conflict lattice (right).

conflicts between opinions for a given query Q. In other words, we want to build a minimal superior lattice $(\mathsf{CL}_Q, \preceq)$ such that it is possible to define a join-homomorphism h (see Def. 17) from $(\mathsf{OL}_Q, \preceq)$ to $(\mathsf{CL}_Q, \preceq)$. In addition, we will require that whenever $h(\langle A_i, Q_i, Sent_i \rangle) = \langle A_j, Q_j, Sent_j \rangle$ then $Sent_i$ and $Sent_j$ are non-conflicting. We will call CL_Q the conflict lattice for Q. By applying a partitioning algorithm it is possible to obtain a conflict lattice from any opinion lattice. This transformation is illustrated in Figure 4.

5 Discussion

Our mathematical characterization of opinion and conflict trees as superior lattices provides a natural foundation for the analysis of important concepts prevailing in argumentation theory. In particular, the use of conflict lattices to represent diverging arguments leads to the identification of the minimal structure that reflects the existing collective positions with respect to a topic of interest.

Argument specificity is a key notion in argumentation theory, as it is the first purely syntactic preference criterion proposed to compare arguments. In our framework, specificity goes hand by hand with the "$\preceq$" relation identified in the resulting superior lattices. The use of minimal structures to represent conflicting views facilitates the identification of specificity relations as well as the recognition of relevant (or irrelevant) elements in the argumentation space, as it is formalized by the notions of sentiment-shifting descriptors (or sentiment-preserving descriptors). Similarly, the minimal-shift relation "$\preceq_{min}$" can be intuitively studied in the light of the resulting mathematical structures.

It must be remarked that our dialectical analysis of TB-arguments aims at modeling the possible space of alternatives associated with different (incrementally more specific) queries. In contrast, the dialectical analysis in standard argumentation frameworks [4, 5] aims at determining the ultimate status of a given argument at issue (in terms of some acceptability semantics).

6 Related Work

Our approach is inspired by recent research in integrating argumentation and social networks. In the last years, there has been growing interest in assessing meaning to streams of data from microblogging services such as Twitter, as well as some recent research on using argumentation for social networks.

To the best of our knowledge, Torroni & Toni [7] 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 this novel view, argumentation frameworks are obtained bottom-up starting from the users' comments, opinions and suggested links, with no top-down intervention of or interpretation by "argumentation engineers". As the authors point out "topics emerge, bottom-up, during the underlying process, possibly serendipitously". In contrast with that proposal, in this paper we generalize this view by identifying arguments automatically from Twitter messages, establishing as well conflict relationships in terms of sentiment analysis (and not specified at the meta-level using rules, as is the case of [7]). In [8], Abbas and Sawamura formalize argument mining from the perspective of intelligent tutoring systems. In contrast with our approach, they rely on a relational database, and their aim is not related with identifying arguments underlying social networks as done in this paper. In [9], 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 modeling of conflicts between arguments in terms of sentiment analysis. In [10], Amgoud and Serrurier propose a formal argumentation-based model for classification, which generalizes the well-known concept learning model based on version spaces [11]. The framework shares some structural similarities with our approach (as a lattice-based characterization is also involved when contrasting hypotheses). However, the aims of the two approaches are different, as our proposal is not focused on solving classification tasks in a machine learning sense.

A related research area is formal concept analysis [12], which is a method for deriving conceptual structures out of data. As done in our approach, the theory of partial orders is used to formally characterize these structures. However, it differs from our proposal in dealing with concepts rather than opinions and in not attempting to associate sentiments with the elements of the partial order. In addition, it does not deal with notions such as arguments, conflict and attack.

It must be remarked that the rise of social media such as blogs and social networks has fueled interest in sentiment analysis. 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. Several research teams in universities around the world currently focus on understanding

the dynamics of sentiment in e-communities through sentiment analysis. The EU funded Cyberemotions consortium[7] 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. Within this project, Thelwall et al.[13, 14] carried out a number of experiments to assess the feasibility of sentiment analysis within social networks, with a particular focus on Twitter. In contrast with our approach, no opinion mining was considered in this context, nor the analysis of alternative opinions (as modelled by conflict trees in our proposal).

7   Conclusions and Future Work

In this paper we have presented a novel approach which integrates argumentation theory and microblogging technologies, with a particular focus on Twitter. To the best of our knowledge, no other approach has been developed in a similar direction. We have also presented a definition of a Twitter-based argument for a query Q that considers as a support the bunch of tweets which are associated with Q according to a given criterion. For such an argument, we also define a prevailing sentiment, obtained in terms of sentiment analysis tools. This allowed us to characterize the notion of opinion tree, which can be recursively built by considering arguments associated with incrementally extended queries . We have implemented a prototype of our proposal as a proof of concept, which was used to compute the opinion tree for the case study presented in the paper.

We have also presented a theoretical setting for analyzing Twitter-based arguments, associating a superior lattice rooted in the initial argument for the first given query. Based on the notion of attack between arguments, we have established as well a refined order relationship between conflicting arguments. As a result, from every superior lattice associated with a given query Q, a *conflict tree* rooted in Q can be built, in which alternating opinions can be better contrasted. Given a node A (argument) associated with query Q' with a prevailing sentiment s, every children node for A in a conflict tree corresponds to an argument for a more specific query Q', which is in conflict with A as it is associated with a sentiment shift. Conflict trees allow us to explore the space of possible confronting opinions associated with a given opinion, using the specificity principle as traditionally used in argumentation for preferring arguments.

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]. We expect to use the underlying algorithms from this tool in our framework. Additionally, we expect to perform different experiments with hashtags associated with relevant topics, assessing the applicability of our approach in a real-world context. Additionally, we are working on extending the current Twitter-based model to a more generic setting, in which

---

[7] http://www.cyberemotions.eu/

opinions are collected from other social networks (such as Facebook).[8] Research in this direction is currently being pursued.

References

1. 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)
2. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: LREC, European Language Resources Association (2010)
3. Prakken, H., Reed, C., Walton, D.: Argumentation schemes and generalizations in reasoning about evidence. In: ICAIL. (2003) 32–41
4. Besnard, P., Hunter, A.: The Elements of Argumentation. The MIT Press. London, UK (2008)
5. Rahwan, I., Simari, G. In: Argumentation in Artificial Intelligence. Springer Verlag (2009)
6. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. TPLP **4**(1-2) (2004) 95–138
7. Torroni, P., Toni, F.: Bottom up argumentation. In: Prof. of First Intl. Workshop on Theoretical and Formal Argumentation (TAFA). IJCAI 2011, Barcelona, Spain. (2011)
8. Abbas, S., Sawamura, H.: Argument mining based on a structured database and its usage in an intelligent tutoring environment. Knowl. Inf. Syst. **30**(1) (2012) 213–246
9. Leite, J., Martins, J.: Social abstract argumentation. In Walsh, T., ed.: IJCAI, IJCAI/AAAI (2011) 2287–2292
10. Amgoud, L., Serrurier, M.: Agents that argue and explain classifications. Autonomous Agents and Multi-Agent Systems **16**(2) (2008) 187–209
11. Mitchell, T.M.: Generalization as search. Artif. Intell. **18**(2) (1982) 203–226
12. Ganter, B., Wille, R.: Formal concept analysis - mathematical foundations. Springer (1999)
13. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. JASIST **63**(1) (2012) 163–173
14. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. JASIST **62**(2) (2011) 406–418
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 (2012) (in press)

---

[8] http://www.facebook.com