A First Approach Towards Integrating Twitter and Defeasible Argumentation

Kathrin Grosse[1]** and Carlos Iván Chesñevar[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@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. 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 for modelling arguments which emerge bottom-up from the social interaction associated with such messages, thus enabling an integration between Twitter and defeasible argumentation. In this paper, we outline the main elements characterizing this integration, identifying "opinions" associated with particular hashtags, obtaining as well other alternative counter-opinions. As a result, we will be able to obtain an "opinion tree", rooted in the first original query, in a similar way as done with dialectical trees in argumentation.

1 Introduction and motivations

Social networks have growth 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 growth into a technology 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.

** First author is a visiting student at the Department of Computer Science and Engineering of the Universidad Nacional del Sur in Bahía Blanca, Argentina.

[1] www.twitter.com

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 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 (i.e., messages posted on Twitter) have a rich structure (see Fig. 1), providing a number of record fields which allow to detect provenance of the tweet (author), number of re-tweets, followers, etc.

We contend that the information made available from such tweets can be useful for modelling 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 instances of the "Argument from Majority" schema [3,

[2] E.g. the current UK Prime Minister 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`

4]. Opinions will have associated sentiments,[4] which might be conflicting, so that counter-opinions might appear. This might lead to a tree-like structure for a dialectical analysis, similar to the one applied in argumentative systems, such as DeLP [5]. In this paper, we analyze the main elements characterizing a possible integration of Twitter and defeasible argumentation. We present as well a particular algorithm for exploring Tweets relevant to a particular hashtag, finding whether it is supported by a positive or negative opinion, obtaining as well other alternative counter-opinions. As a result, we will be able to obtain an "opinion tree", rooted in the first original query.

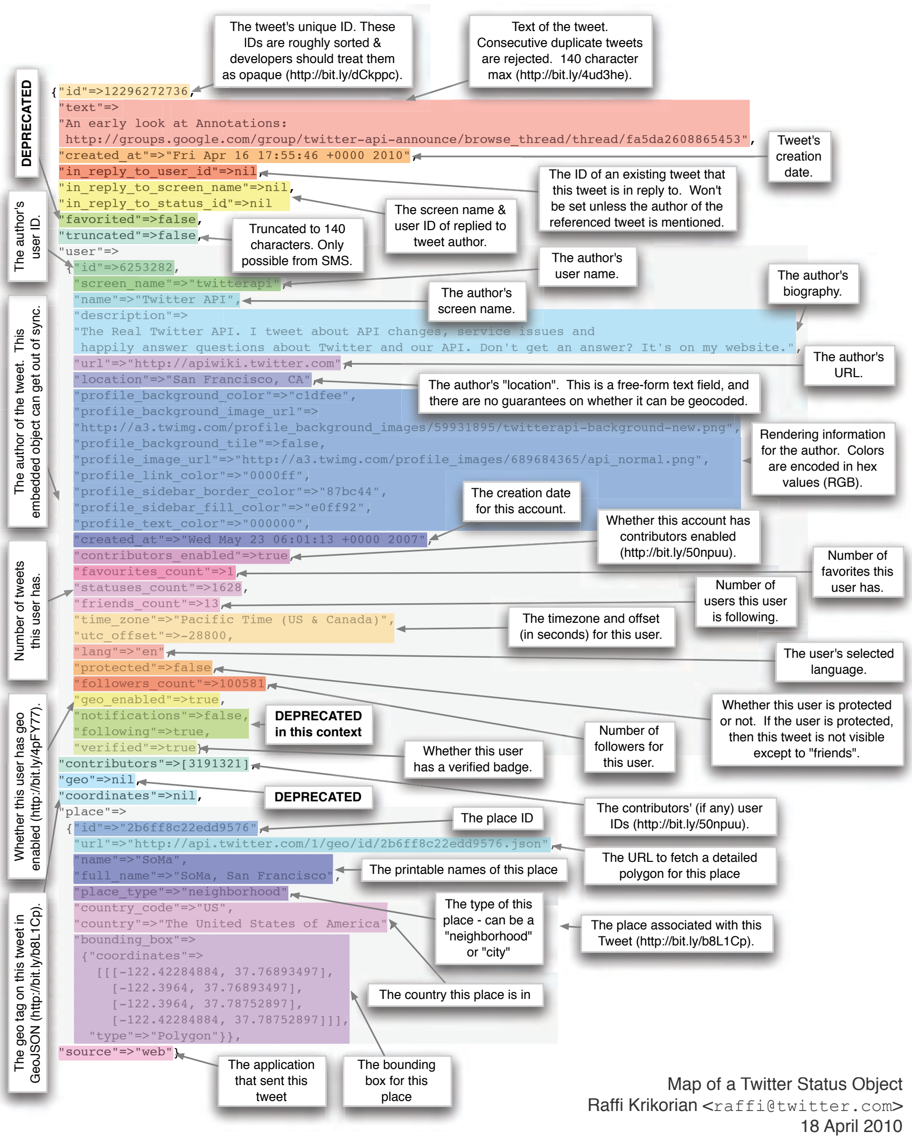
The rest of the paper is structured as follows. Sect. 2 presents an overview of argumentation theory, distinguishing the salient elements in any argumentation system. Then, in Sect. 3 we analyze the parallels between argumentation and Twitter, discussing alternatives for modelling Twitter elements in argumentative terms. Sect. 4 discusses a proposal for exploring Twitter messages in terms "opinion trees", which capture arguments associated with different hashtags. We present a sample case study to illustrate the proposal. Sect. 5 discusses related work, and Sect. 6 concludes.

2 Argumentation: an overview

Argumentation is an important aspect of human decision making. In many situations of everyday's life, people when faced with new information need to ponder its consequences, in particular when attempting to understand problems and come to a decision. Argumentation systems [6] are increasingly being considered for applications in developing software engineering tools, constituting an important component of multi-agent systems for negotiation, problem solving, and for the fusion of data and knowledge. Such systems implement a dialectical reasoning process by determining whether a proposition follows from certain assumptions, analyzing whether some of those assumptions can be disproved by other assumptions in our premises. In this way, an argumentation system provides valuable help to analyze which assumptions from our knowledge base were really giving rise to the inconsistency and which assumptions were harmless.

In defeasible argumentation, an *argument* is a tentative (defeasible) proof for reaching a conclusion. Arguments may compete, rebutting each other, so a *process* of argumentation is a natural result of the search for arguments. Adjudication of competing arguments must be performed, comparing arguments in order to determine what beliefs are ultimately accepted as *warranted* or *justified*. Preference among conflicting arguments is defined in terms of a *preference criterion* which establishes a partial order " $\preceq$ " among possible arguments; thus, for two arguments A and B in conflict, it may be the case that A is strictly preferred over B ($A \succ B$), that A and B are equally preferable ($A \succeq B$ and $A \preceq B$) or that A and B are not comparable with each other.

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



{"id"=>12296272736,
"text"=>
"An early look at Annotations:
http://groups.google.com/group/twitter-api-announce/browse_thread/thread/fa5da2608865453",
"created_at"=>"Fri Apr 16 17:55:46 +0000 2010",
"in_reply_to_user_id"=>nil,
"in_reply_to_screen_name"=>nil,
"in_reply_to_status_id"=>nil
"favorited"=>false,
"truncated"=>false,
"user"=>
{"id"=>6253282,
"screen_name"=>"twitterapi",
"name"=>"Twitter API",
"description"=>
"The Real Twitter API. I tweet about API changes, service issues and
happily answer questions about Twitter and our API. Don't get an answer? It's on my website.",
"url"=>"http://apiwiki.twitter.com",
"location"=>"San Francisco, CA",
"profile_background_color"=>"c1dfee",
"profile_background_image_url"=>
"http://a3.twimg.com/profile_background_images/59931895/twitterapi-background-new.png",
"profile_background_tile"=>false,
"profile_image_url"=>"http://a3.twimg.com/profile_images/689684365/api_normal.png",
"profile_link_color"=>"0000ff",
"profile_sidebar_border_color"=>"87bc44",
"profile_sidebar_fill_color"=>"e0ff92",
"profile_text_color"=>"000000",
"created_at"=>"Wed May 23 06:01:13 +0000 2007",
"contributors_enabled"=>true,
"favourites_count"=>1,
"statuses_count"=>1628,
"friends_count"=>13,
"time_zone"=>"Pacific Time (US & Canada)",
"utc_offset"=>-28800,
"lang"=>"en",
"protected"=>false,
"followers_count"=>100581,
"geo_enabled"=>true,
"notifications"=>false,
"following"=>true,
"verified"=>true},
"contributors"=>[3191321],
"geo"=>nil,
"coordinates"=>nil,
"place"=>
{"id"=>"2b6ff8c22edd9576",
"url"=>"http://api.twitter.com/1/geo/id/2b6ff8c22edd9576.json",
"name"=>"SoMa",
"full_name"=>"SoMa, San Francisco",
"place_type"=>"neighborhood",
"country_code"=>"US",
"country"=>"The United States of America",
"bounding_box"=>
{"coordinates"=>
[[[-122.42284884, 37.76893497],
[-122.3964, 37.76893497],
[-122.3964, 37.78752897],
[-122.42284884, 37.78752897]]],
"type"=>"Polygon"}},
"source"=>"web"}

Fig. 1. Map of a Tweet

For the sake of example, let us consider the well-known example of nonmonotonic reasoning in AI about the flying abilities of birds, recast in argumentative terms. Consider the following sentences: (1) Birds usually fly; (2) Penguins usually do not fly ; (3) Penguins are birds. The first two sentences correspond to *defeasible rules* (rules which are subject to possible exceptions). The third sentence is a *strict rule*, where no exceptions are possible. Given now the fact that *Tweety is a penguin* two different arguments can be constructed:

1. Argument A (based on rules 1 & 3): Tweety is a penguin. Penguins are birds. Birds usually fly. So Tweety flies.
2. Argument B (based on rule 2): Tweety is a penguin. Penguins usually do not fly. So Tweety does not fly.

In this particular situation, two arguments arise that cannot be accepted simultaneously (as they reach contradictory conclusions). Note that argument B seems rationally preferable over argument A, as it is based on more *specific* information. As a matter of fact, specificity is commonly adopted as a syntax-based criterion among conflicting arguments, preferring those arguments which are *more informed* or *more direct* [7, 8]. In this particular case, if we adopt specificity as a preference criterion, argument B is justified, whereas A is not (as it is defeated by B). The above situation can easily become much more complex, as an argument may be defeated by a second argument (a defeater), which in turn can be defeated by a third argument, *reinstating* the first one. As a given argument might have many defeaters, the above situation results in a tree-like structure (called dialectical tree in [5]), rooted in the first argument at issue, where every argument in a branch (except the root) defeats its parent.

As highlighted in [7] several *layers* can be identified in argumentation frameworks: a) an underlying logical language; b) the definition of argument; c) the definition of conflict among arguments; d) the definition of defeat among arguments; e) the status of arguments. In the next section, we analyze some of these elements in the context of analyzing Twitter messages and mining opinions in them.

3 Analyzing Twitter from an argumentative perspective

In this section we will describe how different elements in Twitter can be captured under an argumentative perspective. We will constraint ourselves to the layers a), b) and c) given in the previous section. We are not concerned with the notion of defeat in this paper, as it would imply assigning preference to arguments (opinions) in our context (which is outside the scope of this research). Additionally, as we will see later, we are not concerned in determining whether a particular argument is warranted wrt other possible arguments in terms of some acceptability semantics (as done traditionally in defeasible argumentation [9, 6]). Instead, our dialectical analysis of arguments aims at modeling the possible space of alternatives associated with different (incrementally more specific) queries.

3.1 Logical language for expressing Twitter messages

Twitter messages (Tweets) are 140 character long, with a number of additional fields which help identifying relevant information within a message (sender, number of retweets associated with the message, etc.). In particular, we will focus on the presence of *hashtags* (words or phrases prefixed with the symbol #, a form of metadata tag). Hashtags are used within IRC networks to identify groups and topics and in short messages on microblogging social networking services such as Twitter, identi.ca or Google+ (which may be tagged by including one or more with multiple words concatenated). In the sequel we will assume that the term "hashtag" refers to either actual hashtags in Twitter or to relevant keywords found in tweets.

We define a tweet T as a set of terms $\{t_1, t_2, \ldots t_k\}$. We will consider a distinguished subset t_H of T, where H is a set of hashtags. Let $\mathfrak{Tweets}$ be the set of all currently existing tweets.[5] Given a set of hashtags H, we will write $Tweet_H$ to denote the subset of distinguished elements (tweets) in $\mathfrak{Tweets}$ associated with H. In our approach, a query Q is any set of hashtags used for filtering some relevant tweets $Tweet_Q$ from $\mathfrak{Tweets}$. In order to select those tweets relevant for a particular query Q, we will consider an aggregation operator $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ which returns a subset of tweets associated with Q according to some criterion C. This operator could be defined in several ways, e.g. $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C_1) = \{ T \in \mathfrak{Tweets}$ such that $Q \subset T\}$, or $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C_2) = \{ T \in \mathfrak{Tweets}$ such that $Q \subset T$ and T was retweeted more than 5 times $\}$. Note that for the same query Q, different alternative criteria ($C_1, C_2, \ldots, C_k$) can lead to different distinguished subsets in $\mathfrak{Tweets}$.

As explained before, tweets can be associated with different feelings or sentiments. Even if in real life there may be a lot of emotions in tweets (like anger, happiness, and so on), we will assume here that there is only a set $\mathbb{S}$ of three possible sentiments, which are positive, negative and neutral ones (as done for example in platform Sentiment140.com). Thus our assumption is to a have a mapping s that maps a set of given tweets into a set $\mathbb{S}$ of three sentiments (i.e. $\mathbb{S} = \{$ *positive*, *negative*, *neutral* $\}$). We will abstract away how how the mapping s is computed (we are aware that there may be other ways to rate tweets, such as the number of followers, etc.). We will also abstract away the notion of "potentially conflicting sentiments", assuming that different sentiments are always in conflict.

Next we will formalize the previous notions. Let $s : PartsOf(\mathfrak{Tweets}) \rightarrow \mathbb{S}$ be a mapping. Let $Positive(\mathfrak{Tweets})$, $Negative(\mathfrak{Tweets})$ and $Neutral(\mathfrak{Tweets})$ denote the set of all possible elements in $PartsOf(\mathfrak{Tweets})$ (subset of tweets) that map via s into $\mathbb{S}$. We will assume that $Positive(\mathfrak{Tweets}) \cup Negative(\mathfrak{Tweets}) \cup Neutral(\mathfrak{Tweets}) = \mathfrak{Tweets}$. Two sentiments $s_1, s_2 \in \mathbb{S}$ will be called "in conflict" whenever $s_1 \neq s_2$. (e.g. *positive* will be in conflict with *negative*; *neutral* will be in conflict with *negative* and *positive* will be in conflict with *neutral*).

3.2 Twitter-based Arguments. Conflict

Next we will formalize the notion of Twitter-based argument (TB-argument) and Twitter-based argumentation framework. Intuitively, a TB-argument will

[5] In the analysis that follows, we will assume that the set of all currently existing tweets corresponds to a snapshot of Twitter messages at a given fixed time. It must be noted that the actual Twitter database is highly dynamic.

be be provided by three elements: a support (given by a set of distinguished tweets), a conclusion (associated with a given query Q) and a sentiment *sent*. Formally:

Definition 1. *A Twitter-based argumentation framework (or just framework) is a 5-tuple* $(\mathfrak{Args}, Attacks, C, Agg, Sentiments)$*, where* $\mathfrak{Args}$ *is the set of all possible TB-arguments, Attacks is a partial relationship between elements in* $\mathfrak{Args}$*, Agg is an aggregation operator which selects a subset of elements in* $\mathfrak{Tweets}$ *according to some search preference criterion C for a query Q, and Sentiments is a non-empty set of possible sentiments.*

Definition 2. *Given a framework* $(\mathfrak{Args}, Attacks, C, Agg, Sentiments)$*, a TB-argument for a conclusion Q is a 3-tuple* $\langle Arg, Q, Sentiment\rangle$*, where Arg is* $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ *and Sentiment is* $s(\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C))$*.*

Example 1. Consider a query Q formed by $\{money\}$, and a criterion C defined as "all $t \in \mathfrak{Tweets} \mid \{greece\} \subseteq t$". Then $Arg = \mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ is the set of all possible tweets containing $\{greece\}$. Suppose that $s(\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)) = negative$. Then $\langle Arg, \{money\}, negative\rangle$ is a TB-argument.

Definition 3. *Given a framework* $(\mathfrak{Args}, Attacks, C, Agg, Sentiments)$*, and two queries* Q_1 *and* Q_2*, we will say that* Q_1 *is* strictly more specific *than* Q_2 *whenever* $\mathsf{Agg}_{\mathfrak{Tweets}}(Q_1, C) \subset \mathsf{Agg}_{\mathfrak{Tweets}}(Q_2, C)$*. We will also say that* Q_2 subsumes Q_1*.*

Example: A query Q_2 formed by $\{greece\}$ subsumes the query Q_1 formed by $\{greece, bailout\}$, as all tweets that are returned by Q_1 will also be part of Q_2, but not the other way round.

Definition 4. *Given a framework* $(\mathfrak{Args}, Attacks, C, Agg, Sentiments)$*, and two arguments* $\langle Arg_1, Q_1, Sent_1\rangle$ *and* $\langle Arg_2, Q_2, Sent_2\rangle$ *such that* Q_2 *subsumes* Q_1*, we will say that* $\langle Arg_1, Q_1, Sent_1\rangle$ *attacks* $\langle Arg_2, Q_2, Sent_2\rangle$ *whenever* $Sent_1$ *and* $Sent_2$ *are in conflict.*

Example: Consider two queries $Q_2 = \{greece\}$ and $Q_1 = \{greece, bailout\}$, such that $\langle Arg_1, \{greece, bailout\}, negative\rangle$ and $\langle Arg_2, \{greece\}, positive\rangle$. Then $\langle Arg_1, \{greece, bailout\}, negative\rangle$ attacks $\langle Arg_2, \{greece\}, positive\rangle$.

4 Opinion trees

In the previous section we have shown how to express arguments for queries associated with a given 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 tree* [6]. Based on that notion, we will present next the concept of *opinion tree* to take a closer look to tweets found for a certain hashtag. What do we know if we see that for example for query $\{greece\}$ we obtain the sentiment *positive*? One crucial question surely concerns

ALGORITHM $GetOpinionTree$
INPUT: Query q, Agg, C
OUTPUT: Opinion Tree OT_Q
{ *opinion tree rooted in Q with aggregation Agg under criterion C* }
IF $length(Q) <= 140$
THEN Let $\langle Arg, Q, Sent\rangle$ be the root node
where $Arg = \mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ and Sent is $s(\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C))$.
IF there are other hashtags in $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$ that expand Q
THEN
Compute L= [$h_1, \ldots, h_k$] (list of hashtags that expand Q in $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$)
according to some threshold criterion for considering hashtags to be relevant
(e.g. percentage of occurrence within $\mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$)
FOR EVERY $h_i \in L$ DO
Put $GetOpinionTree(Q \cup \{h_i\}, Agg, C)$ as subtree rooted in $\langle Arg, Q, Sent\rangle$

Fig. 2. High-level algorithm for computing opinion trees from Twitter

if opinions about Greece are related to other topics, like for example vacations, politics, olive oil, etc. It could be for example that all *negative* classified tweets are tweets about the financial crisis and the bailout of the Greek state, and that positive opinions correspond to vacation, tourism, etc.
To explore all possible relationships associated with tweets returned for a specified query Q and criteria C, we developed an algorithm to construct a tree recursively as follows: we start with a TB-argument obtained from the original query ($A = \mathsf{Agg}_{\mathfrak{Tweets}}(Q, C)$), which will be the root of the tree. Next, we compute within A all relevant hashtags that might be used to "extend" Q, by adding a new element ($NewTerm$) to the query, obtaining $Q' = Q \cup \{NewTerm\}$. Then, a new argument for Q' is obtained, which will be associated with a subtree rooted in the original argument A at issue (see high-level algorithm in Fig. 2).

Termination property: For any query Q, the algorithm GetOpinionTree finishes in finite time.
Sketch: Given that a tweet may not contain more than 140 characters, the number of contained hashtags is finite, and therefore the algorithm will eventually stop, providing an opinion tree as an output.

Figure 3 illustrates how the construction of a sample opinion tree for the query {*greece*} could look like. The root node corresponds to those tweets found for the original query, which turns out to be positive (+). Let us suppose that the hashtags { *bailout, inflation* } were found within the previous tweets above a certain threshold, resulting in new conflicting opinions (leading to a negative sentiment). Therefore, two branches will be rooted in the initial argument, one for the query { *greece, bailout* } and the other for the query { *greece, inflation* }. The process could go on further, finding other more specific subsets within the previous nodes (e.g. {*greece, bailout, dracma*} and {*greece, inflation, dracma*}) that lead to conflicting counter-counter opinions, and so on.

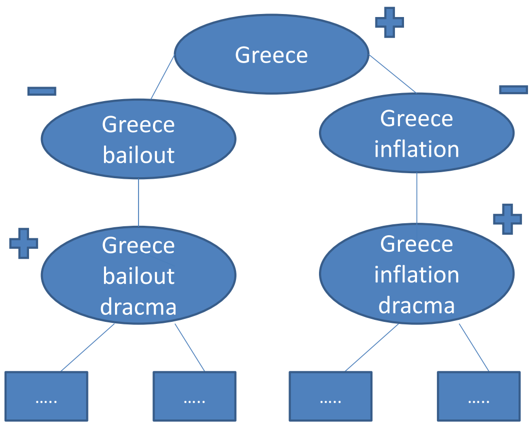


Fig. 3. Example of Opinion Tree based on the query {*greece*}

5 Related Work

Our approach is inspired by recent research in integrating argumentation, social networks and e-democracy. 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 research in using argumentation in e-government contexts. In [10], Cartwright et al presented different issues related to exploiting argument representation in systems for e-democracy. In particular, the authors discuss the contributions of the Parmenides software tool, which is intended as a system for deliberative democracy whereby the government is able to present policy proposals to the public so that users can submit their opinions on the justification presented for the particular policy. In contrast with our approach, this research work assumes that argument schemas are established beforehand, and are not detected as emerging patterns from social network activities. Torroni & Toni [11] coined the term *bottom-up argumentation*, as they take a 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". We generalize this view by identifying two issues: on the one hand, a metalevel characterization of rule-based argument processes, based on social network knowledge bases. On the other hand, we distinguish schema-based argumentation as an alternative for bottom-up argumentation, also obtained in a similar way as for rule-based argumentation. In [12], Heras et al show how the theory of argumentation schemes can provide a valuable help to formalize and structure on-line discussions and user opinions in decision support and business oriented websites that hold social networks among their users. In their analysis real case studies are considered and analyzed, establishing as well guidelines for website and system design to enhance social decision support and recommendations with argumentation. Their research pinpoints several issues presented in our approach, but does not aim at a particular applicability for e-government issues, nor for identifying emerging patterns in network traffic and associating them with high-level arguments. Finally, in [13], 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.

6 Conclusions and Future Work

In this paper we have presented a first approach towards integrating argumentation and microblogging technologies, with a particular focus on Twitter. We have shown how the different elements in argumentation theory can be conceptualized in terms of Twitter messages, according to relevant fields present in those messages (number of retweets, provenance, etc.). We have also presented a definition of argument that considers as a support the bunch of Tweets which are associated with a particular set of terms (hashtags). For such an argument, we also define a polarity (positive, negative, neutral), obtained in terms of sentiment analysis tools. Such polarity allowed us to characterize the notion of conflict between arguments, establishing as well as the backgrounds for formalizing defeat. We showed how this idea could be exploited in terms of so-called "opinion trees", which resemble argumentative dialectical trees. Their aim, in contrast, is to explore the space of possible confronting opinions associated with a given opinion, in terms of the specificity principle 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. 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. In addition, there exists also the possibility of not only expanding hashtags of one set of tweets, but always looking for all tweets given a new hashtag. Thus not a tree but a graph would be built up, and connections between different topics (hashtags) become clear. This would give us the advantage of being able to observe if a special hashtag is positive/negative only together with some other hashtags or by itself (leaving apart indicator words such as “good”, “bad”, etc.). Research in this direction is currently being pursued.

Acknowledgments: We would like to thank the reviewers for their helpful comments and suggestions for improving several aspects of the original presentation of this paper. This research is funded by Projects LACCIR R1211LAC004 (Microsoft Research, CONACyT and IDB), PIP 112-200801-02798 (CONICET, Argentina), PGI 24/ZN10, PGI 24/N006 (SGCyT, UNS, Argentina) and Universidad Nacional del Sur.

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. Bex, F., Prakken, H., Reed, C., Walton, D.: Towards a formal account of reasoning about evidence: Argumentation schemes and generalisations. Artif. Intell. Law **11**(2-3) (2003) 125–165
4. Risse, M.: Arguing for majority rule. The Journal of Political Philosophy **12**(1) (2004) 41–64
5. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. TPLP **4**(1-2) (2004) 95–138
6. Rahwan, I., Simari, G.e. In: Argumentation in AI. Springer (2009)
7. Prakken, H., Vreeswijk, G.: Logics for Defeasible Argumentation. In Gabbay, D., Guenthner, F., eds.: Handbook of Philosophical Logic. Kluwer Academic Publisher (2002) 219–318
8. Chesñevar, C.I., Maguitman, A.G., Loui, R.P.: Logical models of argument. ACM Comput. Surv. **32**(4) (2000) 337–383
9. Besnard, P., Hunter, A.: The Elements of Argumentation. The MIT Press. London, UK (2008)
10. Cartwright, D., Atkinson, K.: Using computational argumentation to support e-participation. IEEE Intelligent Systems **24**(5) (2009) 42–52
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. Heras, S., Atkinson, K., Botti, V.J., Grasso, F., Julián, V., McBurney, P.: How argumentation can enhance dialogues in social networks. In Baroni, P., Cerutti, F., Giacomin, M., Simari, G.R., eds.: COMMA. Volume 216 of Frontiers in Artificial Intelligence and Applications., IOS Press (2010) 267–274
13. 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