Computational Models of Argument
P. Baroni et al. (Eds.)

doi:10.3233/978-1-61499-686-6-21

Tweeties Squabbling: Positive and Negative Results in Applying Argument Mining on Social Media

Tom BOSC [a], Elena CABRIO [a], Serena VILLATA [b]
[a] *Université Côte d'Azur, Inria, CNRS, I3S, France.*
e-mail: tom.bosc@inria.fr; elena.cabrio@unice.fr
[b] *Université Côte d'Azur, CNRS, Inria, I3S, France.*
e-mail: villata@i3s.unice.fr

Abstract. The problem of understanding the stream of messages exchanged on social media such as Facebook and Twitter is becoming a major challenge for automated systems. The tremendous amount of data exchanged on these platforms as well as the specific form of language adopted by social media users constitute a new challenging context for existing argument mining techniques. In this paper, we describe an ongoing work towards the creation of a complete argument mining pipeline over Twitter messages: (i) we identify which tweets can be considered as arguments and which cannot, (ii) over the set of tweet-arguments, we group them by topic, and (iii) we predict whether such tweets support or attack each other. The final goal is to compute the set of tweets which are widely recognized as accepted, and the different (possibly conflicting) viewpoints that emerge on a topic, given a stream of messages.

Keywords. Argument mining, Social media, Supervised classification approaches

1. Introduction

Argumentation has come to be increasingly central as a main study within Artificial Intelligence, due to its ability to conjugate representational needs with user-related cognitive models and computational models for automated reasoning. An important source of data for many of the disciplines interested in such studies is the Web, and social media in particular. Newspapers, microblogs, online debate platforms and social networks provide an heterogeneous flow of information where natural language arguments can be identified and analyzed. The availability of such data, together with the advances in Natural Language Processing and Machine Learning, supported the rise of a new research area called *argument mining*, whose main goal is the automated extraction of natural language arguments and their relations from generic textual corpora, with the final purpose of providing machine-processable data for computational models of argument.

Despite the increasing amount of argument mining approaches [21], none of them has tackled the challenge of extracting arguments and their relations on social media like Twitter or Facebook. Such a kind of natural language arguments raises further issues in addition to the standard problems faced by argument mining approaches typically dealing with newspapers, novels or legal texts: messages from Twitter are squeezed, noisy and often unstructured. More specifically, the following issues have to be considered: *i)* the 140-characters limit forces users to express their ideas very succinctly; *ii)* the quality of the language in Twitter is deteriorated, including a lot of variants in spelling, mistakes and abbreviations, and *iii)* Twitter's API filters tweets on hashtags but cannot retrieve all the replies to these tweets if they do not contain the same hashtags.

In this paper, we provide a preliminary answer to the following research question: *how to extract the arguments and predict the relations among them on Twitter data?* and we highlight the open challenges still to be addressed. We consider both the two main stages in the typical argument mining pipeline, from the unstructured natural language documents towards structured data: we first detect arguments within the natural language texts from Twitter, the retrieved arguments will thus represent the nodes in the final argument graph returned by the system, and second, we predict what are the relations, i.e., *attack* or *support*, holding between the arguments identified in the first stage.

The main advantage of our approach is that it provides a whole argument mining pipeline to analyze flows of tweets, allowing for the application of reasoning techniques over the output structured data, like the identification of the set of widely accepted arguments or trends analysis. However, being it an ongoing work, we highlight in this paper both positive and negative results in applying argument mining on Twitter data, analyzing solutions and potential alternatives to be explored.

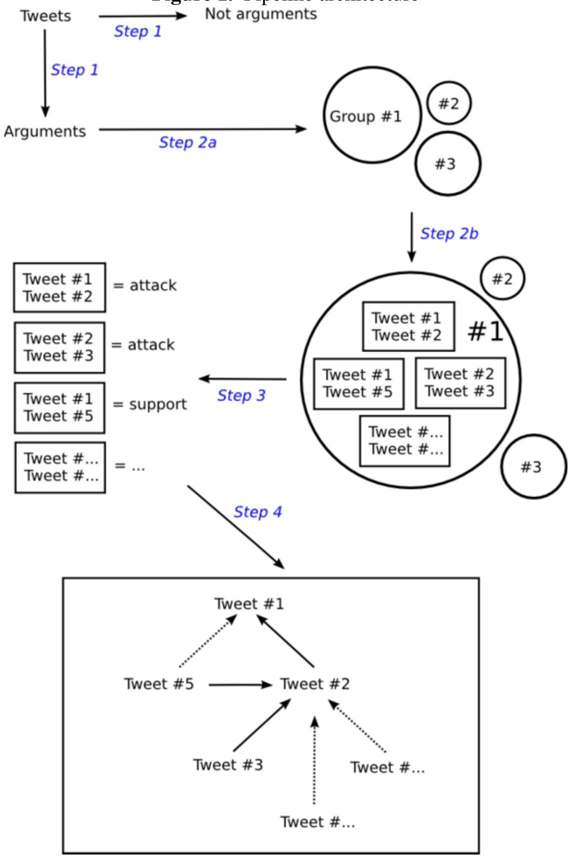
The paper is organized as follows. Section 2 presents our argument mining framework and its evaluation, and Section 3 compares the proposed approach with the related literature. Section 3 describes relevant works in the literature, while Conclusions end the paper, drawing final remarks and describing future work.

2. Argument Mining on Twitter

The argument mining pipeline we propose, visualized in Figure 1, is composed of four main steps, that consist in: *i)* separating tweet-arguments from non-argument tweets; *ii)* grouping tweet-arguments discussing about the same issue, and create pairs of arguments; *iii)* predicting the relations of *attack* and *support* among the tweets in the pairs; and *iv)* building argumentation graphs.

First of all, we need to clarify what we mean by *argument* in this paper: an argument gives a reason to support a claim that is questionable, or open to doubt. In the computational models of argument field, an argument is made of three components: the *premises* representing the reason, a *conclusion* which is the supported claim, and a *relation* showing how the premises lead to this conclusion. Facing the issue of dealing with Twitter data, i.e., dealing with textual arguments of length inferior or equal to 140 characters, we (almost) never find such a kind of complete structure of the arguments. We have thus labeled as arguments all those text snippets providing a portion of a standard argument structure, e.g., opinions under the form of claim, data like in the Toulmin model [30], or persuasive conclusions. Future work includes the "composition" of such elements to build a single well-structured argument. Second, it is worth noticing that the support and the attack relations are not symmetric: we considered the temporal dimension to decide the direction of these relations, i.e., a tweet that is proposed at time $t+1$ attacks (resp.

Figure 1. Pipeline architecture



supports) a tweet which has been provided at time t. In the following, each step of the pipeline is described in detail, together with the experimented approach, and the obtained results of this ongoing work.

Dataset. Up to our knowledge, DART [7] is the only existing dataset of arguments and their relations on Twitter, therefore it has been chosen to test our pipeline. It is composed of:

(a) 4000 tweets annotated as argument/not argument: 1000 tweets for each of the following 4 topics: the letter to Iran written by 47 senators on 10/03/2015; the referendum in Greece for or against Greece leaving European Union on 10/07/2015; the release of Apple Watch on 10/03/2015; the airing of episode 4 (season 5) of the serie Game of Thrones on 4/05/2015. A tweet is annotated as argument if it contains an opinion or factual information, or if it is a claim expressed as question (rhetorical questions, attempts to persuade, containing sarcasms/irony). The argument annotation task is carried out on a single tweet and not on subparts of it. A text containing an opinion is considered as an argument. For example, in the following tweet the opinion of the author is clearly expressed in the second sentence (i.e., *I won't be running out to get one*):

RT @mariofraioli: What will #AppleWatch mean for runners? I can't speak for everyone, but I won't be running out to get one. Will you? http://t.co/xBpj0HWK

We consider as arguments also claims expressed as questions (either rhetorical questions, attempts to persuade, containing sarcasm or irony), as in the following example:

RT @GrnEyedMandy: What next Republicans? You going to send North Korea a love letter too? #47Traitors

or:

Perhaps Apple can start an organ harvesting program. Because I only need one kidney, right? #iPadPro #AppleTV #AppleWatch

Tweets containing factual information are annotated as arguments, given that they can be considered as premises or conclusions. For example:

RT @HeathWallace: You can already buy a fake #AppleWatch in China http://t.co/WpHEDqYuUC via @cnnnews @mr_gadget http://t.co/WhcMKuM

Defining the amount of world knowledge needed to determine whether a text is a fact or an opinion when it contains unknown acronyms and abbreviations can be pretty tough. Consider the following tweet:

RT @SaysSheToday: The Dixie Chicks were attacked just for using 1A right to say they were ashamed of GWB. They didn't commit treason like the #47Senators

where the mentioned entities *The Dixie Chicks*, *GWB*, and *1A right* are strictly linked to the US politics, and hardly interpreted by people out of the US politics matters. In this case, annotators are asked to suppose that the mentioned entities exist, and focus on the phrasing of the tweets.

However, if tweets contain pronouns only (preventing the understanding of the text), we consider such tweets as not "self-contained" , and thus non arguments. It can be the case of replies, as in the following example, in which the pronoun *he* is not referenced anywhere in the tweet.

@FakeGhostPirate @GameOfThrones He is the one true King after all ;)

For tweets containing an advertisement to push into visiting a web page, if an opinion or factual information is also present, then the tweet is considered as an argument, otherwise it is not. Consider the following example:

RT @NewAppleDevice: Apple's smartwatch can be a games platform and here's why http://t.co/uIMGDyw08I

It contains factual information that can be understood even without visiting the link. On the contrary, the following tweet is not an argument, given that it does not convey an independent message while excluding the link:

For all #business students discussing #AppleWatch this morning. Give it a test drive thanks to @UsVsTh3m: http://t.co/x2bGc9j1Gl.

(b) 2181 tweet-arguments on the Apple Watch release classified in 7 categories (i.e. *features (F)*, *price (P)*, *look (L)*, *buying announcement (B)*, *advertisement (A)*, *forecast on the product success (S)*, *news (N)*, *others (O)*) (see Table 3). Moreover, the tweets contained in the category *features* have been grouped in the following more fine-grained categories: *health*, *innovation*, *battery*.

(c) 1891 pairs of tweet-arguments of the categories: *price*, *health*, *look*, *predictions* annotated with the following relations: *support* (446), *attack* (122), *unknown* (1323). After a first annotation round to test the guidelines provided in [9], we realized that a few additional instructions should be added with the aim to consider the specificity of the Twitter scenario. The instructions we introduced are as follows: If both Tweet-A and Tweet-B in a pair are factual tweets, and they are related to the same issue, the pair must be annotated as *support*, as in:

> Tweet-A: *.@AirStripmHealth + #AppleWatch provides HIPPA compliant capabilities for physicians, mothers, babies, and more #AppleEvent*
> Tweet-B: *accessible heart rate monitors and opinions on that #iWatch #apple #accessibility #ios* `https://t.co/ySYM8dk0Pf` *via @audioBoom*

If both Tweet-A and Tweet-B in a pair are opinion tweets, and they are related to the same issue, the pair must be annotated as *support*, as in:

> Tweet-A: *Think of how much other stuff you can buy with the money you spend on an #AppleWatch*
> Tweet-B: *#AppleWatch Tempting, but not convinced. #appletv Yes. #iPhone6sPlus No plan to upgrade #iPadPro little high price, wait & watch*

If Tweet-B is a factual tweet, and Tweet-A is an opinion on the same issue, the pair must be annotated as *support*, as in:

> Tweet-A: *Wow. Your vitals on your iwatch. That's bonkers. #AppleEvent*
> Tweet-B: *accessible heart rate monitors and opinions on that #iWatch #apple #accessibility #ios* `https://t.co/ySYM8dk0Pf` *via @audioBoom*

If Tweet-A is a factual tweet, and Tweet-B expresses someone's wishes to buy the product or an opinion about it, the pair must be annotated as *unknown*, as in:

> Tweet-A: *Mom can listen to baby's heart rate with #AppleWatch #airstrip*
> Tweet-B:*Wow!!! Look at what the #Ap, pleWatch can do for #doctors that's amazing! Seeing their vitals? I just got chills! In a good way #AppleEvent*

Concerning the annotation of the arguments/non arguments, in the reconciliation phase among the three students annotators, the label that was annotated by at least 2 annotators out of 3 was chosen (majority voting mechanism). If all the annotators disagree or if more than one annotator labels the tweet as unknown, then such tweet is discarded. The inter-annotator agreement has been calculated between the expert annotators and the reconciled student annotations on 250 tweets of the first batch, resulting in $\alpha_{47traitors} = 0.81$ (Krippendorff's α handles missing values, the label "unknown" in our case). Concerning the pair annotation with the support/attack/unknown relations, the inter-annotator agreement has been calculated on 99 pairs (33 pairs randomly extracted from each of the three first topics), resulting in Krippendorff $\alpha = 0.67$.

Dataset	# tweet-arg.	not-arg.	unknown	total
Training set	2079	829	92	3000
Test set	623	352	25	1000

Table 1. Statistics of dataset (a)

Approach	Average F1
baseline	0.64
baseline + tokens	0.66
baseline + tokens + bigrams tokens	0.67

Table 2. Validation of the model and feature use

2.1. Step 1: Argument identification.

The first task in our pipeline is the binary classification of tweets as argument/non argument. To train a generic, domain-independent argument detector, we separate the training, validation and test data according to the topics of dataset (a) to avoid overfitting. We train and validate on the first three topics, and we test on the Apple Watch dataset (Table 1 provides some statistics on the data).We ignore tweets classified as unknown. We use 3-fold cross-validation (we alternately train the model on the tweets of the first two topics and leave the third topic out as a validation set) with randomized hyperparameter search [3].[1] Because the classes are unbalanced and the balance is not necessarily the same across all datasets, the training phase weights the errors inversely proportional to class frequencies.

As baseline, we use raw character counts as features (causing smileys, capital letters, punctuation marks to influence the model). Then, tweets have been tokenized with Twokenize[2] and annotated with their PoS applying Stanford POS tagger. POS tags are then used as features, as well as bigrams of tags. As a baseline model, we train a logistic regression model[3] on these features only.

We also augment features with normalized tokens and bigrams of tokens, and this effectively improves over the baseline (see Table 2). The best model (Logistic regression, L2-penalized with $\lambda = 100$) is obtained by using all the features and re-training on the 3 folds. It yields an F1-score of 0.78 over the test set, that can be considered as satisfactory. The difference between the average F1-score over the validation set (see Table 2) and the F1 over the test set is due to the addition of the tweets of the validation set (around 1000 additional tweets) for training the final model.

[1] A randomized hyperparameter search samples parameter settings a fixed number of times and has been found to be more effective in high-dimensional spaces than exhaustive search.

[2] `http://www.cs.cmu.edu/~ark/TweetNLP/`

[3] Like all regression analyses, the logistic regression is a predictive analysis. It is used to describe data and to measure the relationship between one dependent variable and one or more independent variables by estimating probabilities using a logistic function, i.e., the cumulative logistic distribution.

	O	A	B	F	L	N	P	S
#	720	175	370	619	205	65	189	112

Table 3. Statistics on dataset (b), # tweets

	F	L	P	S
average F1-score (train set)	0.36	0.57	0.60	0.15
F1-score (test set)	0.56	0.58	0.60	0.00

Table 4. Classification results (step 2)

2.2. *Step 2: Pairs creation.*

Once we are able to identify tweet-argument, we create pairs of them to predict the relations among them. Given a stream of tweets, it would be impossible to apply a naive approach comparing all the pairs of tweets, since this would lead to the creation of numerous unrelated pairs.

To deal with this issue, we firstly tested the solution of clustering the tweets into *sub-topics*, and then create pairs from these sub-topics. The major problem that we faced is the difficulty of automatically finding meaningful sub-topics. We tested both Latent Dirichlet Allocation[4] [6] and more powerful models such as Correlated Topic Models[5] [5], but the interpretability of the clusters did not improve [11].

Instead, since we have classified goldstandard data for Apple Watch (dataset (b), see Table 3), we decided to focus on this topic only, and turn the clustering problem into a classification problem. Another possibility would have been to tune the hyperparameters before applying the clustering algorithms to retrieve the annotated categories, but given the small size of the goldstandard, we could not explore that direction further.

In particular, we focus on categories F (features), L (look), P (price) and S (predictions about the success of the product) because they contain the most interesting tweets. We use the same features and same hyperparameters selection scheme as in step 1. The training set contains 2031 tweets, and the test set contains 150 tweets. The 3 folds are randomly created across all the training set, and we take the average of all the macro F1-scores on all the folds to select the best model. We use regularized logistic regression and the results obtained by the best model (L1-penalized with $\lambda = 100$) are reported in Table 4 for each category, averaged over all the folds. As can be observed, some categories are harder to predict than others, but the performance on the easy classes (F, L, P here) are quite satisfactory. A paraphrase detection tool could be added at this step to deduplicate similar tweets and give more weights to the arguments that are often used in subsequent steps.

[4] Latent Dirichlet allocation is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

[5] Correlated Topic Models use a more flexible distribution for the topic proportions that allows for covariance structure among the components. This gives a more realistic model of latent topic structure where the presence of one latent topic may be correlated with the presence of another.

2.3. Step 3: Relation detection.

Given the pairs of tweet-arguments returned by step 2, the next step consists in predicting the relation holding between the tweets in a pair. Dataset (c) contains $\sim$600 tweets each for *look*, *price* and *health* categories of the Apple Watch: we put pairs concerning the product price in the test set, whereas all the other tweets are in the training set. An additional validation set contains 100 tweets on the user predictions on the product success.

Given the closeness of the task with textual entailment [9], we decide to explore first a prediction of the support and attack relations using the Excitement Open Platform (EOP)[6] for recognizing textual entailment. The intuition is to consider the support relation as an entailment, and the attack relation as a contradiction, following the approach in Cabrio and Villata [8].

In addition, following the same guidelines proposed by [9], pairs are also annotated according to the Recognizing Textual Entailment (RTE) framework, i.e., pairs linked by a support relation as *entailment/non-entailment*, and pairs linked by an attack relation as *contradiction/non-contradiction*.

However, given the specificity of Twitter data and the fact that predicting support and attack relations is not the same as recognizing entailment, results were far from being satisfying (see Table 5), also due to the huge number of unrelated pairs (tagged as unknown in Dataset (c)). Then we decided to implement a neural sequence classifier inspired by [26]. We encode the tokens as precomputed GloVe embeddings[7] [24] of size 200. When a token does not have an embedding, we generate a random embedding according to a multivariate normal distribution with empirical mean and variance of existing embeddings.

Such a neural classifier is an encoder-decoder architecture with two distinct Long Short-Term Memory networks[8] (LSTM) [16], where we pass the last hidden-state of the first LSTM to initialize the second. The probabilities over the 3 categories are given by a softmax function, i.e., a function which takes as input a C-dimensional vector z and outputs a C-dimensional vector y of real values between 0 and 1, at the output layer of the second LSTM at the last pass. Our objective is cross entropy, and we oversample the attack and support categories so that the probability of drawing a tweet from a category is uniform on the three categories. We use Stochastic Gradient Descent with Adam[9] [17] to optimize. We periodically test our model against the validation set, and stop the training when the validation error stops improving. We select the best performing model on the validation set. However, also in this case, results are not satisfying (see Table 5).

We realize that such classification step on Twitter is pretty hard, even for human. As an example, consider the following pair:

[6] `http://hltfbk.github.io/Excitement-Open-Platform/`

[7] GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

[8] Long Short-Term Memory networks are a special kind of Recurrent Neural Networks, capable of learning long-term dependencies.

[9] Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments.

Model	EOP (MaxEnt)	Neural model
F1-score Support	0.17	0.20
F1-score Attack	0.0	0.16

Table 5. Comparing the two models

T1: *Can't believe the designers of #AppleWatch didn't present a better shaped watch. It's still too clunky looking & could've been more sleek.*
T2: *@APPLEOFFIClAL amazing product updates. Apple TV looks great. BUT! Please make a bigger iWatch! Not buying it until it's way bigger.*

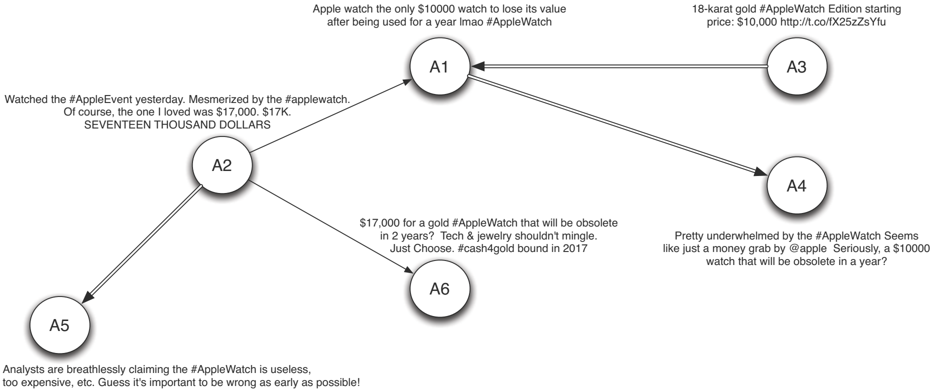
On the one hand, the tweets agree in that the watch is not properly sized. On the other hand, they disagree since one user finds it too big and the other one too small, which are opposite viewpoints.

The neural model is more promising because it can be easily used in a semi-supervised settings, but the lack of a large-sized corpus is a huge hurdle for training such a model (however, there is a huge amount of data in the DART dataset that has not been labeled yet, for which an annotation effort should be considered).

2.4. Step 4: Graph building

We can now build an *argument graph* whose nodes are the arguments and whose edges are the predicted relations (supports/ attacks). An example of such a graph is visualized in Figure 2, where an extract of the tweets for the iWatch topic is presented. It is easy to note that such a kind of visualization allows for a deeper understanding of the ongoing Twitter discussion, and would provide a valuable support for social media content analysis.

Figure 2. Example of argumentation graph (where single edges represent attack and double ones represent support) resulting from the identified arguments and predicted relations for the iWatch topic.


Apple watch the only $10000 watch to lose its value after being used for a year lmao #AppleWatch
18-karat gold #AppleWatch Edition starting price: $10,000 http://t.co/fX25zZsYfu
A1
A3
Watched the #AppleEvent yesterday. Mesmerized by the #applewatch. Of course, the one I loved was $17,000. $17K. SEVENTEEN THOUSAND DOLLARS
A2
A4
$17,000 for a gold #AppleWatch that will be obsolete in 2 years? Tech & jewelry shouldn't mingle. Just Choose. #cash4gold bound in 2017
Pretty underwhelmed by the #AppleWatch Seems like just a money grab by @apple Seriously, a $10000 watch that will be obsolete in a year?
A6
A5
Analysts are breathlessly claiming the #AppleWatch is useless, too expensive, etc. Guess it's important to be wrong as early as possible!

The last step of the pipeline consists in applying argumentation semantics to identify the set(s) of accepted arguments. Several systems can be adopted to perform such a computation in a scalable way, as those participating to the ICCMA challenge [29]. In our framework, we used the ASPARTIX-D system[10], after the flattening of the bipolar

[10] https://ddll.inf.tu-dresden.de/web/Sarah_Alice_Gaggl/ASPARTIX-D

argumentation framework to an abstract Dung-like argumentation framework, as done in [9]. This step returns the set of acceptable arguments such that the different (coherent) viewpoints expressed through the tweets are highlighted, as well as the identifiable attack points in the stream.

Some considerations can be drawn about the resulting graphs. First of all, graphs are, differently from [10] for instance, rather sparse, meaning that they do not present a star structure. They are more like a set of subgraphs connected with each other, where each subgraph concerns a different sub-issue of the general topic, i.e., the price of the Hermes iWatch band inside the *Price* issue of the iWatch topic. This is a specificity of Twitter discussions being them a continuous stream of messages. Second, as for the case of the debates extracted in [10], no cycle is present.

3. Related Work

The first stage of the argument mining pipeline is to detect arguments within the input texts. Many approaches have recently tackled such challenge adopting different methodologies, e.g., SVM [22,23,28,12,20], Naïve Bayes classifiers [4], Logistic Regression [18].

The second stage consists in predicting what are the relations holding between the arguments identified in the first stage. This is an extremely complex task, as it involves high-level knowledge representation and reasoning issues, and, for this reason, existing approaches assume simplifying hypotheses, like the fact that evidence is always associated with a claim [2].

However, all these approaches do not tackle the challenge of applying argument mining to Twitter data. Argumentation is applied to Twitter by [13] who extract a particular version of arguments they called "opinions" based on incrementally generated queries. Their goal is to detect conflicting elements in an opinion tree to avoid potentially inconsistent information. Both the goal and the adopted methodology is different from the one we present in this paper.

Finally, to tackle these challenging tasks, high-quality annotated corpora are needed, see [25,22,18,2,27,10,14], to be used as a training set for any kind of aforementioned prediction. None of these corpora deals with Twitter data. An exhaustive state of the art about argument mining techniques and applications is in [21].

4. Conclusions

In this paper, we present an ongoing work to apply the argument mining pipeline on Twitter data. This challenging task can be divided into the following three sub-tasks: *i)* the identification of tweet-arguments from non argumentative tweets in the stream of tweets, *ii)* the composition of tweet-arguments into meaningful pairs where pairs of completely unrelated tweet-arguments are discarded, and *iii)* the prediction of the relation, i.e., support or attack, between the tweet-arguments in a pair. While we achieved satisfiable results concerning sub-tasks *(i)* and *(ii)*, negative results are shown even by applying different strategies to sub-task *(iii)*. Even if we know that negative results do not convey to solutions, we belive that they represent an unavoidable step in an emerging research topic as argument mining is, and they provide a useful guide to the further exploration of the faced challenge. This is why we report them in this paper.

Investigating potential solutions to this open issue is our main future work direction. To address this argument structure prediction task, we are exploring the application of relation classification in discourse analysis techniques [19], and semantic textual similarity estimation techniques [1]. Another open challenge in dealing with Twitter is about big data: Twitter provides a very large data collection that raises the issue of the scalability of the applied argument mining techniques. Making our framework robust and scalable enough to process the Twitter streams of data is another future research line.

Acknowledgement

The work carried out in this paper is partially funded by the start-up Vigiglobe (Sophia Antipolis, France).

References

[1] Palakorn Achananuparp, Xiaohua Hu, and Xiajiong Shen. The evaluation of sentence similarity measures. In Il-Yeol Song, Johann Eder, and Tho Manh Nguyen, editors, *Data Warehousing and Knowledge Discovery, 10th International Conference, DaWaK 2008, Turin, Italy, September 2-5, 2008, Proceedings*, volume 5182 of *Lecture Notes in Computer Science*, pages 305–316. Springer, 2008.

[2] Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argumentation Mining*, page 6468. Association for Computational Linguistics, 2014.

[3] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305, February 2012.

[4] Or Biran and Owen Rambow. Identifying justifications in written dialogs by classifying text as argumentative. *Int. J. Semantic Computing*, 5(4):363–381, 2011.

[5] David M. Blei and John D. Lafferty. Correlated topic models. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. MIT Press, 2006.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[7] Tom Bosc, Elena Cabrio, and Serena Villata. Dart: a dataset of arguments and their relations on twitter (accepted for publication). In *Proceeding of LREC 2016*, 2016.

[8] Elena Cabrio and Serena Villata. Combining textual entailment and argumentation theory for supporting online debates interactions. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 208–212, 2012.

[9] Elena Cabrio and Serena Villata. A natural language bipolar argumentation approach to support users in online debate interactions. *Argument & Computation*, 4(3):209–230, 2013.

[10] Elena Cabrio and Serena Villata. Node: A benchmark of natural language arguments. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Computational Models of Argument - Proceedings of COMMA 2014, Atholl Palace Hotel, Scottish Highlands, UK, September 9-12, 2014*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 449–450. IOS Press, 2014.

[11] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296, 2009.

[12] Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Meth-*

ods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015, pages 2236–2242. The Association for Computational Linguistics, 2015.

[13] Kathrin Grosse, María Paula González, Carlos Iván Chesñevar, and Ana Gabriela Maguitman. Integrating argumentation and sentiment analysis for mining opinions from twitter. *AI Commun.*, 28(3):387–401, 2015.

[14] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. Argumentation mining on the web from information seeking perspective. In Elena Cabrio, Serena Villata, and Adam Wyner, editors, *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014.*, volume 1341 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.

[15] Jan Hajic and Junichi Tsujii, editors. *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*. ACL, 2014.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. Context dependent claim detection. In Hajic and Tsujii [15], pages 1489–1500.

[19] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 343–351. ACL, 2009.

[20] Marco Lippi and Paolo Torroni. Context-independent claim detection for argument mining. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 185–191. AAAI Press, 2015.

[21] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology*, 2016.

[22] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining. *Artif. Intell. Law*, 19(1):1–22, 2011.

[23] J. Park and C. Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, 2014.

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[25] Chris Reed and Glenn Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(4):983, 2004.

[26] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. In *International Conference on Learning Representations*, 2016.

[27] Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In Hajic and Tsujii [15], pages 1501–1510.

[28] Christian Stab and Iryna Gurevych. Identifying argumentative discourse structures in persuasive essays. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 46–56. ACL, 2014.

[29] Matthias Thimm and Serena Villata. System descriptions of the first international competition on computational models of argumentation (ICCMA'15). *CoRR*, abs/1510.05373, 2015.

[30] S.E. Toulmin. *The Uses of Argument*. Cambridge University Press, 2003.