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

© 2016 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/978-1-61499-686-6-431

Assessing Weight of Opinion by Aggregating Coalitions of Arguments

Pavithra RAJENDRAN [a], Danushka BOLLEGALA [a] and Simon PARSONS [b]
[a] *Department of Computer Science, University of Liverpool*
[b] *Department of Informatics, King's College London*

Abstract. Argument mining promises to be able to extract information from unstructured text that can help us to understand that text. This paper suggests a novel way to use such information once it has been extracted. Attack and support relations between arguments from a set of test texts are identified, the strength of the arguments is computed based on the relations, and arguments are grouped into coalitions. The resulting set of arguments is then used to predict the weight of opinion in new text, by identifying arguments whose weight has been computed, and aggregating these weights. Our approach is evaluated on a corpus of hotel reviews, and compared with an existing method of predicting the sentiment of reviews.

Keywords. argument mining, bipolar argumentation, coalitions of arguments

1. Introduction

Argumentation mining is an emerging field that focuses on the identification and extraction of arguments from natural language texts. The aim of such work is to pinpoint what opinions are expressed for and against some point of view. These arguments can then be used in understanding the text, perhaps to highlight the key issues raised, or to summarise the overall view expressed in the text. In this paper we contribute to the growing literature around argument mining, studying the use of arguments that have been extracted. In particular, we are interested in investigating how techniques from computational argumentation can be used to process the results of argument mining, establishing what can be done to gain insights about the texts from which the arguments were mined.

In this paper the texts we process are online reviews. We take a set of arguments that are hand-extracted from reviews, and, based on ideas from bipolar argumentation [1], extract coalitions of arguments that relate to the products (hotels) that are reviewed. We then evaluate different approaches for aggregating these coalitions, and assess whether the result of the aggregation can be used to predict the weight of opinion about the products, as expressed by the star rating of the reviews. Note that we are not interested in reviews *per se* — for a set of reviews, the overall star rating is probably the best guide to the weight of opinion. However, precisely because of these star ratings, reviews are a very convenient dataset to refine our approach before analysing more general texts.

Any work on argument mining will be dependent on the precise definition of "argument" that is used in that work. This term has several definitions. A typical definition is the combination of a set of premises and the conclusion that these premises lead to. Argu-

Table 1. Statements present in a review annotated as argument or not using our definitions.

Statements	Sentiment	Aspect	Argument	Type
My mother and I stayed at the Warwick for 2 nights in November.	objective	none	no	-
The hotel itself was ok, fairly clean and decent location.	positive	yes	yes	support
The front desk staff, however are not helpful and pass the buck so as not to have to deal with a problem.	negative	yes	yes	attack

ments of this form are difficult to extract from the unstructured text present in online reviews, forums, blogs etc and methods to accurately extract them are under-development. Wyner et al. [2], for example, describes work extracting such arguments using a set of argumentation schemes. Garcia-Villalba and Saint-Dizier [3] also show how this approach can help in generating arguments using evaluative expressions such as "*well located hotel*" and also evaluate such statements using rhetorical relations for argument extraction.

Rather than focussing on extracting structured arguments, we use knowledge of product attributes to extract statements that can be considered arguments for or against a product. We deal with what we call *aspects*. In our terminology an aspect is an entity relating to a product or service about which a review writer expresses an opinion. Both aspects and the relation between aspect properties and opinions about the product or service are highly domain dependent. For example, "*battery is small and lightweight*" in an electronics review is a positive statement about the battery aspect while "*rooms are small*" is a negative statement about the room aspect in a hotel review.

Statements about aspects are then classified as arguments for or against a product:

Argument A statement that is either a supporting argument or an attacking argument.

Supporting argument A statement that has positive polarity and can be considered to support the product by supporting an aspect of the product or the product itself.

Attacking argument A statement that has a negative polarity and can be considered to attack the product by attacking an aspect of the product or the product itself.

Examples of arguments can be found in Table 1. Note that we group related aspects into *aspect categories*, and consider that there is some level of equivalence between statements about aspects in the same category. Given a set of arguments of this form, this paper examines whether they can be used to establish the overall opinion in a way that agrees with the review writers.

2. Background

Dung's abstract argumentation framework [4] provides a framework for analysing a set of arguments with attack relations between them. Bipolar argumentation [5] extends this by introducing the notion of support as an independent interaction among arguments:

Definition 1. An abstract bipolar argumentation framework is a 3-tuple $\langle \mathscr{A}, \mathscr{S}, \mathscr{R} \rangle$ where $\mathscr{A}$ is a set of arguments such that $\mathscr{S}$ represents the support relation and $\mathscr{R}$ represents the attack relation between the arguments.

A bipolar argumentation framework can be represented as a bipolar interaction graph in which arguments are nodes and support and attack relations are edges. Cayrol & Lagasquie-Schiex [1] further proposed the structuring of a bipolar argumentation framework into coalitions of arguments.

Definition 2. A coalition of arguments is a set of arguments supporting each other directly or indirectly where conflicts occur among such coalitions. These coalitions of arguments satisfy the following properties:

1. There is no direct attack among pairs of arguments belonging to the same coalition.
2. Any pair of arguments in a coalition will have a direct or indirect support relation between them.
3. If an argument in coalition A attacks an argument in coalition B, then A attacks B.

Since a set of reviews of a given product will contain multiple arguments for and against different aspects of that product, we consider such a set of reviews as a coalition of arguments.

3. Data preparation

3.1. Dataset

We used an existing dataset, the ArguAna corpus [6], which contains manually annotated hotel reviews from TripAdvisor.com. The corpus contains each review, identified with a review id, the author name, the local sentiment of each statement (positive or negative) in the review, and the aspects present in the statement. Each review has the star rating provided by the reviewer. Several existing classifiers are available for automatically identifying sentiment, but since sentiment data was already available, we used it. We manually collected the aspects present in each review, and each statement that contained any of the aspects and was labelled as positive or negative was considered to be an argument. Every statement was extracted from each review for a given hotel, and the arguments were collected together regardless of whether or not they belonged to the same review.

3.2. Automatic identification: Support/Attack

The ArguAna corpus does not contain relations between arguments present in the reviews. To extract this information we used the Takelab STS[1] System. There are three types of relations that we wanted to identify between pairs of arguments — support (arguments about aspects in the same aspect category with same sentiment), attack (arguments about aspects in the same aspect category with opposite sentiment) and unknown (arguments about aspects in different aspect categories).

These definitions are based on inferring whether two statements support/attack in different ways but target the same conclusion. The aspects of the product or service are grouped into different categories based on their common properties. For instance, in a hotel review, the aspects *staff* and *manager* belong to the same aspect category.

To detect relations, we took a sample set of arguments, paired them according to the above definitions and manually annotated the relations. We then used Takelab STS to obtain the semantic similarity scores for each pair of arguments. TakeLab STS accepts

[1] `http://takelab.fer.hr/sts/`

two statements as input and produces a semantic similarity score ranging from 0 (lowest semantic similarity) to 5 (highest semantic similarity). In our experiments, the maximum similarity score was 3. To avoid errors, we set a minimum similarity score of 1.0 as a threshold below which we consider that there is no relation between statements (and above which we considered a relation to hold), which gave a macro-averaged F1-score of 0.18 for automatically predicting the manually annotated relations. While relation prediction was not perfect, it was sufficient for our purposes.

3.3. Coalitions of arguments in reviews

Arguments present in reviews, according to our definition, relate to aspects. Considering the properties of the support and attack relations with respect to aspects, we noticed that the support relations naturally fall into coalitions, where each argument within a coalition relates to the same aspect and all the arguments support each other directly or indirectly. This gives rise to several questions such as what kind of coalitions of arguments are formed in a single review, and in a set of randomly selected reviews. We were not able to find coalitions of arguments in a single review, since it seems that in our dataset each review contains at most one statement about each aspect. For the remainder of the paper we study coalitions of arguments across sets of Low reviews (reviews with 1 star and 2 star rating) and sets of High reviews (4 star and 5 star ratings).

4. Aggregating natural language arguments

We are interested in interpreting a set of reviews for a particular hotel. Across all the reviews of that hotel, a number of aspects will have been mentioned by the reviewers. We consider all the comments about a specific aspect as being an argument for or against the hotel, and we will aggregate these arguments to get the overall opinion about the hotel.

4.1. Arguments for aspects

The first step in the process is to identify attack and support relations between arguments. This is done, as described above, using TakeLab STS. The second step is to compute the weight of each argument. There are several methods that we could use to compute the weight of an argument on the basis of the arguments that support and attack it, and from these possibilities we picked the intrinsic generic gradual valuation method proposed by Cayrol and Lagasquie-Schiex [7] which takes into account arguments that support and attack the argument in question:

Definition 3. For every argument $a \in \mathscr{A}$ with a set of supporters $\mathscr{B} = \{b_1, b_2, \ldots, b_n\}$, and attackers $\mathscr{C} = \{c_1, c_2, \ldots, c_m\}$, the gradual valuation function $\mathrm{g} : \mathbb{R}^2 \to \mathbb{R}$ is defined as:

$$v(a) = g_{\mathrm{agg}}(h^{sup}_{agg}, h^{att}_{agg}) = \left(\frac{1}{h^{att}_{agg} + 1} - \frac{1}{h^{sup}_{agg} + 1} \right); h^{sup}_{agg}(\mathscr{A}) = \sum_{i=1}^{n} v(b_i), h^{att}_{agg}(\mathscr{A}) = \sum_{i=1}^{m} v(c_i) \quad (1)$$

We assume the initial strength value of each argument satisfies $v(b_i) = 1$ for $i = 1, \ldots, n$, and $v(c_j) = 1$ for $j = 1, \ldots, m$, irrespective of whether they are supporting or attacking.

The previous step gives us a value for each individual argument. Before combining arguments to summarise reviews, we structure agruments into coalitions. We do this exactly following Definition 2. This gives us a set of coalitions, each of which supports or attacks an aspect. Each coalition is a set of arguments, and each argument has a weight.

4.2. Aggregating coalitions

We consider the opinion about each aspect category of a hotel to be an argument about the hotel. The strength of opinion about the hotel is then a combination of the strengths of opinions about the aspect categories (which depend on the arguments in the coalitions that relate to the aspects). We could establish the opinion about the hotel by combining all the aspect categories and all the arguments for each aspect category, but it isn't clear that we want to include either all the arguments that bear on each aspect category or all the aspect categories that relate to each hotel. We infer this on the basis of the work of Wachsmuth et al. [6] who studied the patterns of positive and negative statements in the same corpus that we use and showed that the most negative reviews contain most of the negative statements and the most positive reviews contain most of the positive statements. This suggests that we should only consider a subset of the arguments present when assessing hotels. To do this, we divided the arguments into two categories, Low and High, using the ground-truth data provided by the star ratings of the reviews in which each of the arguments were present. Arguments in Low reviews were rated Low, those in High reviews were rated High. We then considered four different ways in which to choose the arguments that should be taken into account:

ArgAll All arguments, regardless of the coalition they belong to, are taken into account, and we consider arguments for all aspect categories when rating a hotel.

AttSupCoal All attacking arguments from coalitions of arguments in Low rated reviews, and all supporting arguments in coalitions of arguments in High rated reviews are taken into account. Again we consider arguments for all aspect catgeories.

AttSupArg This is a refinement of *AttSupCoal* in which we only consider the arguments relating to the aspect catgeory attacked by the strongest attacking coalition, and the arguments relating to the aspect category supported by the strongest supporting coalition when rating a hotel.

AttSupBoth A hybrid version of *AttSupCoal* and *AttSupArg*. It initially picks all attacking arguments from coalitions of arguments in Low rated reviews, and all supporting arguments in coalitions of arguments in High rated reviews just like *AttSupCoal*. However, these arguments are then filtered by only including arguments for those aspects (rather than aspect categories) that are present in the review being rated. The strengths of the resulting sub-coalitions are computed, and the strongest attacking and supporting sub-coalitions are used to rate the hotel (echoing *AttSupArg*).

These four approaches all identify a set of arguments to take into account. We then experimented with two ways of using these sets of arguments, both taking all the arguments in the set into account, an approach we call f_{agg}, and just taking the strongest arguments in a set into account, an approach we call f_{max}. Again, the idea of focusing on the strongest arguments comes from [6].

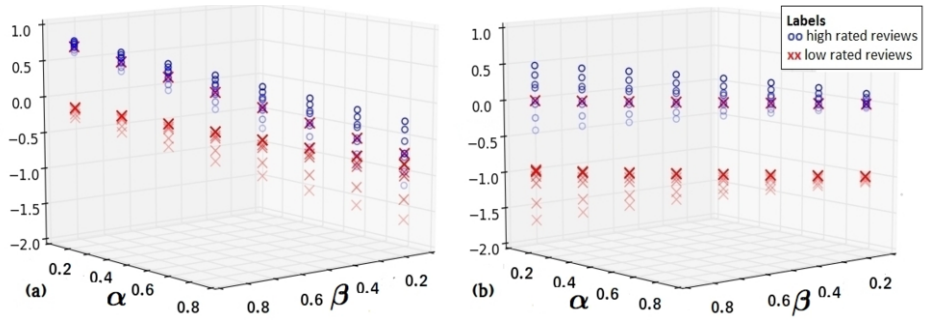

Figure 1. Scores of each review for a single hotel. A red cross denotes a review that belongs to Low and a blue circle denotes a review that belongs to High. (a) Scores vs α and β aggregating using *ArgAll* (b) Scores vs α and β aggregating using *AttSupBoth*. Both use f_{max}.

4.3. Argument aggregation value function

For each hotel review we consider, we now have a set of arguments for and against it, selected using the methods described above. Having already used Eq. 1 to compute the strength of each argument, it is natural to use Eq. 1 to combine the strengths of the arguments for and against each hotel. However, such a combination does not distinguish well between Low and High rated reviews. As a result, we introduce a generalisation of Eq. 1 in which supporting and attacking arguments are weighted differently. In particular we considered the overall strength of an argument to be a function of the strength of the coalition of supporting arguments (SCV: supporting coalition value) and the coalition of attacking arguments (ACV: attacking coalition value):

$$f(h^{sup}(SCV), h^{att}(ACV)) = \left(\frac{1}{\beta h^{att}(ACV)+1} - \frac{1}{\alpha h^{sup}(SCV)+1}\right) \tag{2}$$

where, α, β, $\alpha+\beta=1$ provide a simple way of weighting the support and attack components differently. Exactly which arguments are included in *ACV* and *SCV* depends on the choice from $\{AllArg, AttSupArg, AttSupCoal, ArgSupBoth\}$ and $\{f_{max}, f_{agg}\}$.

To establish the optimum values of α and β to use with Eq, 2, we computed results for values of α and β across $[0,1]$. For each pair of values, we followed a process analogous to 10-fold cross-validation, training on 90% of the reviews and testing on 10%, and averaging results across 10 repetitions[2]. We did this for 14 different random hotel datasets, each of which contains an average of 25 individual reviews. We performed the experiment for both balanced and unbalanced sets of reviews, recognising that this gave us three different categories of hotel that we were attempting to categorise — hotels with a majority of low rated reviews (unbalanced), hotels with a majority of high rated reviews (unbalanced) and hotels with a balanced set of low rated and high rated reviews (balanced). We repeated the experiment for *ArgAll* and *AttSupBoth* with f_{max}. Figure 1 shows the results, scores for a set of reviews belonging to a particular hotel. Each Low-rated review is represented by a red cross and each High-rated review is represented by a blue circle. The score of each review is computed using varying values of α and β, and

[2]In this work "training" is going through the process of extracting arguments from reviews, weighting the arguments and identifying coalitions. "Testing" is then using these arguments to rate new reviews.

Table 2. Results for prediction of reviews. The numbers are the percentage of reviews correctly predicted into category Low and High. The highest value on each line is highlighted. Test data was reviews from 14 different hotels. There were 217 Low reviews and 148 High reviews. Because of the imbalance between Low and High, we report results for hotels where the majority of reviews were Low, where the majority of reviews were High, and where the number of reviews were approximately equal, as well as the overall results.

		AttSupBoth		*AttSupArg*		*AttSupCoal*		*AllArg*	
	Category	f_{agg}	f_{max}	f_{agg}	f_{max}	f_{agg}	f_{max}	f_{agg}	f_{max}
Majority Low reviews	Low	96	**97**	88	92	74	90	80	68
	High	37	**50**	22	35	31	22	16	16
Balanced reviews	Low	90	**93**	85	90	76	87	85	72
	High	35	35	33	45	**54**	26	23	23
Majority High reviews	Low	84	**92**	88	**92**	52	88	80	64
	High	23	40	38	38	**76**	25	28	28
Overall	Low	93	**96**	86	90	72	87	80	70
	High	36	46	31	39	**54**	24	20	20

Table 3. Comparison with ArguAna. Conditions as in Table 2.

	Category	Majority High	Balanced	Majority Low	Overall
AttSupBoth, f_{max}	Low	97	93	92	96
	High	50	35	40	46
ArguAna	Low	99	93	100	97
	High	29	21	30	28

from the figures it is evident that, (a) for *ArgAll* aggregation there is no clear separation between Low and High reviews for any value of α and β whereas (b) for *AttSupBoth* aggregation, there is a clear gap between the scores of low rated and high rated reviews that seems to widen for particular values of α and β. This suggests that our approach, along with aggregation of arguments based on Eq. 2, *AttSupBoth* and f_{max} can weigh up arguments in a review in a way that broadly agrees with the writer of the review.

4.4. Evaluation

Having established the potential of our approach, we carried out a more detailed evaluation. First we examined the relative performance of the four methods for picking which coalitions to take into account (*ArgAll*, *AttSupCoal*, *AttSupArg*, *AttSupBoth*) and the two methods for selecting arguments to aggregate (f_{agg} and f_{max}). We ran the same 10-fold cross-validation exercise as before, set $\alpha = 0.75$ and $\beta = 0.25$, and evaluated the methods by predicting whether reviews for 14 randomly selected hotels were High or Low. This was a set of 217 Low reviews and 148 High reviews. The results are given in Table 2 which reports the percentage of reviews that were correctly predicted. We ran two-tailed t-tests on each pair of comparable results — that is every pair of results on the same line of the table. All differences in value are significant at the 0.05 level except those between the predictions made by *AttSupArg*/f_{agg} and *AttSupCoal*/f_{max} for the Low category.

The results suggest that the combination of *AttSupBoth* and f_{max} is the best predictor across the different segments, though it is outperformed by *AttSupCoal* and f_{agg} in terms of the prediction of High reviews. We interpret this as evidence that focusing on the most strongly held relevant opinion (as Wachsmuth et al. [6] suggest) is the key to good

prediction, but that the best way to pick the relevant opinions (the ones from which the strongest are selected) varies depending on whether the the review is positive or negative. To come back to our original question — whether we can combine arguments to reach a view that matches the opinion of the review writers — the results suggest we can do this well for High, and with some accuracy for Low reviews, though with the latter there is considerable room for improvement. The dataset we used was developed to test a sentiment classification tool called *ArguAna* [8]. We compared our approach (*AttSupBoth*, f_{max}) with ArguAna. The results are given in Table 3. Considering the Overall results, a two-tailed t-test tells us that our approach is significantly better in predicting High reviews and not significantly worse in predicting Low reviews. In fact, in all categories, our approach does much better in predicting High reviews.

5. Conclusion

This paper considered the task of weighing up the arguments in a text to determine the overall opinion being expressed. We proposed a method that starts from a set of arguments extracted from online reviews. This involves identifying support and attack relations between these arguments, computing the weight of the arguments, and identifying coalitions of arguments. Having established a training set of such arguments, we showed that the arguments, weights and coalitions could be used to evaluate new reviews in a way that can distinguish between two broad classes of positive and negative reviews. Our approach compares well with an existing approach to sentiment analysis of reviews, outperforming the existing approach in identifying positive reviews and doing no worse on negative reviews. Note that the overall aim of this work is not to predict the sentiment of reviews. We concentrated on reviews here because reviews come with star ratings that provide a form of ground truth data about the opinion of the review writer. Our aim is to be able to summarise the opinion expressed in general texts.

Acknowledgement: PR is supported by a scholarship from the University of Liverpool.

References

[1] C. Cayrol and M.-C. Lagasquie-Schiex. Coalitions of arguments: A tool for handling bipolar argumentation frameworks. *Int. J. Intell. Syst.*, 25(1):83–109, 2010.

[2] A. Wyner, J. Schneider, K. Atkinson, and T. J. M. Bench-Capon. Semi-automated argumentative analysis of online product reviews. In B. Verheij, S. Szeider, and S. Woltran, editors, *COMMA'12*, volume 245 of *Frontiers in Artificial Intelligence and Applications*, pages 43–50. IOS Press, 2012.

[3] M. P. Garcia-Villalba and P. Saint-Dizier. A framework to extract arguments in opinion texts. *IJCINI*, 6(3):62–87, 2012.

[4] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games. *Artif. Intell.*, 77:321–357, 1995.

[5] L. Amgoud, C. Cayrol, and M.-C. Lagasquie-Schiex. On the bipolarity in argumentation frameworks. In J. P. Delgrande and T. Schaub, editors, *NMR'04*, pages 1–9, 2004.

[6] H. Wachsmuth, M. Trenkmann, B. Stein, G. Engels, and T. Palakarska. A review corpus for argumentation analysis. In *ICCLITP'14*, pages 115–127, April 2014.

[7] C. Cayrol and M.-C. Lagasquie-Schiex. Gradual valuation for bipolar argumentation frameworks. In L. Godo, editor, *ECSQARU'05*, volume 3571 of *LNCS*, pages 366–377. Springer, 2005.

[8] H. Wachsmuth, M. Trenkmann, B. Stein, and G. Engels. Modeling review argumentation for robust sentiment analysis. In *ICCL'14*, pages 553–564, 2014.