Computational Models of Argument
H. Prakken et al. (Eds.)
© 2020 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/FAIA200513

Analysing Product Reviews Using Probabilistic Argumentation

Kawsar NOOR [a], Anthony HUNTER [a]
[a] *Department of Computer Science, University College London*

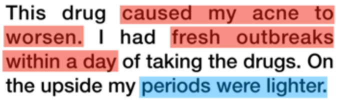
Abstract. Product reviews which are increasingly commonplace on the web typically contain a textual component and a numerical rating. The textual component can be viewed as a collection of arguments for and against the product. Whilst the reviewer may not have provided the attacks between these arguments they typically provide an indication of which set of arguments they view as being more acceptable/winning via the numerical rating (i.e. a positive rating indicates that the positive arguments are accepted and vice versa). Our framework builds upon this intuition and we propose a two step process for identifying a probability distribution over the set of possible argument graphs that the reviewer may have had in mind. The first is the *identification step* in which for a given review, we identify a distribution by analysing the relationship between the rating and polarity of arguments in the review via the constellations approach to probabilistic argumentation. The second step is the *refinement step* in which we harness ratings from multiple reviews and use this to refine our probability distribution thus enabling us to learn from the data. We illustrate the applicability of our approach by testing it with real data.

Keywords. Probabilistic argumentation; online reviews; abstract argumentation

1. Introduction

An abstract argument framework, as proposed by Dung, [7] is a graph structure in which the vertices denote arguments and the edges denote attacks between the arguments. Probabilistic argumentation extends abstract argumentation by allowing one to associate probabilities with the argument frameworks. In the epistemic approach probabilities are associated with arguments and represent uncertainty in the arguments themselves. In contrast in the constellations approach probabilities are associated with the topology of a graph and this enables one to model uncertainty in the structure of the graph.

In this paper we explore the use of the constellations approach to model agent reasoning within product reviews. These reviews contain arguments for and against the product; many reviews also have numerical ratings that capture the reviewer's overall sentiment towards the product. In essence the rating represents the reviewer's final verdict on the product and is provided in light of the arguments they have provided in favour of and against the product. With this in mind we therefore assume that for each review there is an underlying abstract argument graph which captures the reviewer's reasoning. In order to predict this graph we use the constellations approach to identify a probability distribution over potential argument graphs for each review. The distribution is useful as it can be used to predict a graph for the review that can be used to then understand the



This drug caused my acne to worsen. I had fresh outbreaks within a day of taking the drugs. On the upside my periods were lighter.

Figure 1. Example drug review in which the reviewer gave a rating of 3/10. Text spans labelled red indicate a negative argument against the drug and blue labels indicate positive arguments

reviewer's reasoning. Likewise when considering multiple reviews for a product one can develop an understanding of how all of the reviewers view a product by aggregating and reasoning with the distributions identified.

To illustrate the problem consider a reader browsing through many product reviews that contain arguments in favour of (**positive arguments**) or against (**negative arguments**) a particular product where each review comes with a numerical rating. We interpret this rating as a proxy for the polarity of the winning arguments. Hence we view a review with a high rating as an indication that the winning arguments are positive and vice versa. This also affords us an understanding of the potential graphs assignable to the review.

As an example consider the review shown in Figure 1. We can reason that the low rating is being driven by one or both of the negative arguments and consequently that the positive argument does not play much of a role in the overall assessment. We can express our reasoning using Dung's grounded semantics and say that the argument graph that the reviewer had in mind will likely have one or both of the negative arguments in its grounded extension and not the positive argument. This can be formalised further using probabilistic argumentation.

In this paper we propose a method for identifying a probability distribution over the set of graphs that the reviewer may have had in mind. This is achieved in two steps. The first is the *identification step* in which we identify a distribution for a review by building upon the assumption that there is a relationship between the rating and the winning / acceptable arguments in that review. This distribution can then be sampled from to assign a graph to the review. When considering multiple reviews we propose an additional *refinement step* that makes use of data derived from ratings taken from a dataset of reviews in order to refine the probability distribution so as to better reflect the data.

However not all reviews contain ratings and hence in our experiment section we demonstrate that we can train a machine learning model to predict ratings for such reviews. Also we see that our proposal is not limited to product reviews and can indeed be used in any situation in which agents posit arguments and proxy measures that indicate which arguments win. For example in a public debate where the viewers, over the course of the debate, accumulate arguments from both parties and instead of providing the attacks between these arguments or directly identifying the winning arguments they may instead rate each party thus indicating their overall verdict. Our approach could thus be used to identify a probability distribution over the set of possible argument graphs for each viewer.

To summarise we make two main contributions with this paper. Our first contribution is providing a methodology for identifying a probability distribution over a set of argument graphs given a review. Our second contribution is refining this distribution by incorporating data derived by analysing the ratings from multiple reviews and thus having a distribution that better reflects the reviews we are modelling.

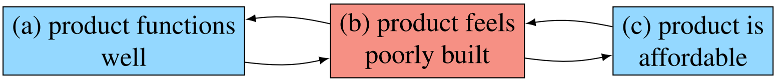

(a) product functions well
(b) product feels poorly built
(c) product is affordable

Figure 2. An example of an argument graph containing two positive arguments in favour of a product (a,c) and one negative argument against it (b)

2. Identifying a Probability Distribution for a Review

Given an argument graph (A,R) a set $B \subseteq A$ is **conflict-free** iff no two arguments $a,b \in B$ exists s.t $(a.b) \in R$. An argument b is **defended** by a set $B \in A$ iff any argument $a \in A$ attacks b then $\exists c \in B$ s.t. $(c,a) \in R$. A conflict-free set $B \subseteq A$ is an **admissible** extension iff each argument in B is defended by B. An admissible extension B is an **complete** extension iff each argument defended by B is in B. A complete extension B is a **grounded** extension if it is minimal (w.r.t set inclusion). We use the notation $\mathsf{gr}((A,R))$ to indicate the grounded extension for a graph.

We start by considering a setting in which users state positive and negative arguments which are arguments in favour of or against a particular conclusion (e.g. in the case of product reviews these are in favour or against the product). In other words we consider, as a simplifying assumption, only bipartite graphs. We thus see reviews as user provided arguments for or against the product and the rating they provide as indicators of the underlying argument graph and therefore the winning arguments. Although on the web ratings tend to be integers they can in fact be any real number.

Definition 2.1. Let A^+ be a set of positive arguments and A^- be a set of negative arguments s.t. $A^+ \cap A^- = \emptyset$. Let the minimum rating be b^{Neg}_{min} and the maximum be b^{Pos}_{max}. A **view** is a tuple $v = (A,b)$ where $A \subseteq A^+ \cup A^-$ and $b \in [b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}]$ is a **rating** s.t $b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max} \in \mathbb{R}$ and $b^{\mathsf{Neg}}_{min} < b^{\mathsf{Pos}}_{max}$.

Example 2.1. Consider the arguments depicted in Figure 2 and rating. Some examples of views using the arguments $\{a,b,c\}$ and ratings in the range $[1,10]$ would be $(\{a,b,c\},9)$ and $(\{a,b\},10)$.

When considering the set of possible argument graphs that an agent may have had in mind when providing a view we are dealing with all argument graphs which contain the arguments in that view. This translates as the set of all spanning sub graphs using those arguments. We refer to this set as the **graph space**. Formally we say that given disjoint sets A^+ and A^- that the graph space is the set returned by the function $\mathsf{Space}(A^+,A^-) = \{(A^+ \cup A^-, R) | R \in \mathscr{P}((A^+ \times A^-) \cup (A^- \times A^+)\}$. An example of a a graph space given two positive and one negative argument is provided in Table 1.

Proposition 2.1. *Given a set of positive and negative arguments A^+ and A^- let $m = |A^+|$ and $n = |A^-|$. The size of the graph space is then 2^{2mn}.*

In identifying the probability distribution for a view we make the assumption that the rating in a view is proportional to the ratio of positive/negative arguments in the grounded extension for the graph the agent intended for that view; hence if the rating is high we expect this ratio to be high and vice versa. In terms of the graph space we expect that when a rating is high, those graphs that have a high proportion of positive arguments in their grounded extension will have more mass assigned to them and vice versa.

To this end we rank graphs in the graph space based on two criteria which we define in the rest of this section: the degree of polarity of the graph's grounded extension (proportion of positive/negative arguments) and an assessment of the topological structure of each graph. An analysis of the graph's attack structure provides a finer-grained understanding of how the grounded extension is achieved, therefore enabling us to better differentiate between graphs that share the same grounded extension.

Definition 2.2. Let A^+, A^- be positive and negative arguments and $S = \mathsf{Space}(A^+,A^-)$ be a graph space. For each graph $G \in S$ we define the sets $\mathsf{gr}^+(G) = \{a \in \mathsf{gr}(G) | a \in A^+\}$ and $\mathsf{gr}^-(G) = \{a \in \mathsf{gr}(G) | a \in A^-\}$. We then say that the **polarity** of a graph G is $\mathsf{Pol}(G) = |\mathsf{gr}^+| - |\mathsf{gr}^-|$ and that the graph space can be partitioned into the following sets: $\mathsf{Pos}(S) = \{G \in S | \mathsf{Pol}(\mathsf{gr}(G)) > 0\}$, $\mathsf{Ntl}(S) = \{G \in S | \mathsf{Pol}(\mathsf{gr}(G)) = 0\}$ and $\mathsf{Neg}(S) = \{G \in S | \mathsf{Pol}(\mathsf{gr}(G)) < 0\}$.

Proposition 2.2. $\mathsf{Pol}(G) \in \mathbb{Z}$ *(i.e. the set of integers) and* $|A^-| \leq \mathsf{Pol}(G) \leq |A^+|$.

Proposition 2.3. *For any graph space in which there are m positive arguments and n negative arguments then* $|\mathsf{Pos}(S)| > |\mathsf{Neg}(S)|$ *when* $m > n$ *and* $|\mathsf{Neg}(S)| > |\mathsf{Pos}(S)|$ *when* $n > m$.

To analyse the polarity of an argument graph based on its attacks we define a function that scores each argument based on the number of attacks it inflicts and sustains.

Definition 2.3. Let $G = (A,R)$ be an argument graph. For an argument $a \in A$ the number of attacks it receives is $\mathsf{def}(a) = |\{(x,y) \in R | y = a\}|$ and the number of attacks it inflicts is $\mathsf{att}(a) = |\{(x,y) \in R | x = a\}|$. The **grade** of argument a in graph G is then $\mathsf{Grade}(a,G) = \mathsf{att}(a) - \mathsf{def}(a)$.

Example 2.2. Consider graph G_1 in Table 1. We can see that $\mathsf{Grade}(a,G_1) = 1$, $\mathsf{Grade}(c,G_1) = 1$ and $\mathsf{Grade}(b,G_1) = -2$.

The grade of an argument is maximal when it attacks all of its opponents without being attacked at all and vice versa.

Proposition 2.4. *Let* $\mathsf{Space}(A^+,A^-)$ *be a graph space. Given* $B \in \{A^+,A^-\}$ *and* $a \in B$ *then* $\mathsf{max}_{G \in \mathsf{Space}(A^+,A^-)} \mathsf{Grade}(a,G) = |A^+ \cup A^- \setminus B|$ *and* $\mathsf{min}_{G \in \mathsf{Space}(A^+,A^-)} \mathsf{Grade}(a,G) = -|A^+ \cup A^- \setminus B|$.

Proposition 2.5. *Given an argument graph G it holds that* $\sum_{a \in A^+} \mathsf{Grade}(a,G) + \sum_{b \in A^-} \mathsf{Grade}(b,G) = 0$.

The grade of an argument is a score that is a combined indicator of an argument's ability to defend its coalition whilst not being attacked by the opposition in a particular graph [10]. Given our aim of ranking graphs in a graph space we are however interested in comparing the grade of an argument in a particular graph to its grades in the other graphs in the graph space so as to assess how well it performed in that particular graph. In order to gain this relative perspective we use a process of normalisation as follows:

Definition 2.4. Given a graph space $S = \mathsf{Space}(A^+,A^-)$ and an argument $a \in A^+ \cup A^-$ the **normalised grade** for a is given below where $\mathsf{min}(a,S) = \mathsf{min}_{G \in S} \mathsf{Grade}(a,G)$ and $\mathsf{max}(a,S) = \mathsf{max}_{G \in S} \mathsf{Grade}(a,G)$.

$$\mathsf{NormGrade}(G,S,a) = \frac{\mathsf{Grade}(G,a) - \mathsf{min}(a,S)}{\mathsf{max}(a,S) - \mathsf{min}(a,S)}$$

Example 2.3. Consider Table 1 which uses the arguments $a,c \in A^+$ and $b \in A^-$. The normalised grade for arguments a,c are highest in G_1 as they are attacking all of their opponents and not being attacked. The opposite is true in G_{16}.

Proposition 2.6. *Given a graph space $S = \mathsf{Space}(A^+,A^-)$ where $p = |A^+|$ and $n = |A^-|$ then for a positive argument $a \in A^+$, $\max_{G\in S} \mathsf{Grade}(G,a) = n$ and $\min_{G\in S} \mathsf{Grade}(G,a) = -n$. Likewise for a negative argument that $b \in A^-$, $\max_{G\in S} \mathsf{Grade}(G,b) = p$ and $\min_{G\in S} \mathsf{Grade}(G,b) = -p$.*

By summing the normalised grades for the arguments in an argument graph we are able to produce a value that summarises the polarity of attacks in that graph.

Definition 2.5. Given a graph space $S = \mathsf{Space}(A^+,A^-)$, for each $G \in S$ the aggregate score for positive arguments is $AttackScore^+ = \sum_{a\in A^+} \mathsf{NormGrade}(G,S,a)$ and the aggregate score for negative arguments is $AttackScore^- = \sum_{a\in A^-} \mathsf{NormGrade}(G,S,a)$. The **aggregate polarity score** for the graph is then given by $\mathsf{AttackScore}(S,G) = AttackScore^+ - AttackScore^-$.

When considering the ordered set of attack scores for graphs in S the difference in attack score between any two consecutive graphs is a constant ΔAtt as given in the following result.

Proposition 2.7. *Let $p = |A^+|$, $n = |A^-|$ and $(AttackScore_0,..,AttackScore_m)$ be a sequence of all the attack scores ordered from largest to smallest in the set $\{\mathsf{AttackScore}(G)|G \in S\}$ s.t for each i, $AttackScore_{i+1} \geq AttackScore_i$. For any two values in the sequence it holds that the pairwise difference between them is a constant ΔAtt i.e. ΔAtt = $AttackScore_i - AttackScore_{i+1}$ where $\Delta Att = \frac{1}{2n} + \frac{1}{2p}$.*

We can then use ΔAtt to bring together the functions Pol and AttackScore to give a combined assessment of the polarity of the graphs in a graph space.

Definition 2.6. Let $(G_1,..,G_m)$ be a sequence of graphs in S s.t. for any two graphs G_i, G_{i+1} it holds that $\mathsf{Pol}(G_i) \geq \mathsf{Pol}(G_{i+1})$ and $\mathsf{AttackScore}(G_i) \geq \mathsf{AttackScore}(G_{i+1})$. We say that $\mathsf{alike}(G_i,G_{i+1})$ holds iff $\mathsf{Pol}(G_i) = \mathsf{Pol}(G_{i+1})$ and $\mathsf{AttackScore}(G_i) = \mathsf{AttackScore}(G_{i+1})$. We define the **aggregate score** of a graph G_i s.t. $i > 1$, as $\mathsf{Agg}(G_i) = \mathsf{Agg}(G_{i-1})$ if $\mathsf{alike}(G_i,G_{i-1})$ and $\mathsf{Agg}(G_i) = \mathsf{Agg}(G_{i-1} - \Delta Att$ otherwise and $\mathsf{Agg}(G_1) = \mathsf{AttackScore}(G_1)$.

To illustrate we can see that in Table 1 the attack scores are non-unique. The *Agg* function enables us distinguish between graphs such as G_{10} and G_{11} which share the same attack score but not the same grounded extension.

We now consider how ratings can be used in identifying a probability distribution for a view. Our proposal for this is a function which maps a rating to an aggregate value which is in turn used in specifying our probability distribution. In order to produce this polynomial we partition the rating scale into three categories which correspond to the three categories of polarity defined in Definition 2.2.

No	Graph	Gr(G)	Attack Score	Agg	P(G)						
					10	9	8	7	6	5	4,3,2,1
G_1	a → b ← c	a,c	2	2	0.16	0.06	0.01	0	0	0	0
G_2	a b ← c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_3	a → b c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_4	a → b ↔ c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_5	a ↔ b ← c	a,c	1.25	1.25	0.13	0.11	0.04	0.01	0.01	0.01	0.01
G_6	a → b → c	a,c	0.5	0.5	0.08	0.12	0.11	0.05	0.03	0.02	0.02
G_7	a ← b ← c	a,c	0.5	0.5	0.08	0.12	0.11	0.05	0.03	0.02	0.02
G_8	a b ↔ c	a	0.5	-0.25	0.05	0.07	0.15	0.11	0.06	0.05	0.05
G_9	a ↔ b c	c	0.5	-0.25	0.05	0.07	0.15	0.11	0.06	0.05	0.05
G_{10}	a b c	a,b,c	0.5	-0.25	0.05	0.07	0.15	0.11	0.06	0.05	0.05
G_{11}	a ↔ b ↔ c		0.5	-1	0.02	0.03	0.07	0.15	0.11	0.09	0.08
G_{12}	a b → c	a,b	-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{13}	a ← b c	b,c	-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{14}	a ↔ b → c		-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{15}	a ← b ↔ c		-0.25	-1.75	0.01	0.01	0.02	0.08	0.13	0.14	0.13
G_{16}	a ← b → c	b	-1	-2.5	0	0	0	0.03	0.01	0.13	0.19

Table 1. Breakdown of probability distribution and aggregate graded scores for each graph in a graph with 2 positive arguments and one negative

Definition 2.7. Let $[b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}]$ be the range of possible ratings assignable in a view where $b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}, \in \mathbb{R}$. Within this range we define the positive partition as $[b^{\mathsf{Pos}}_{min}, b^{\mathsf{Pos}}_{max}]$, the neutral partition as $[b^{\mathsf{Ntl}}_{min}, b^{\mathsf{Ntl}}_{max}]$ and the negative partition as $[b^{\mathsf{Neg}}_{min}, b^{\mathsf{Neg}}_{max}]$ s.t $b^{\mathsf{Pos}}_{max} > b^{\mathsf{Pos}}_{min} > b^{\mathsf{Ntl}}_{max} > b^{\mathsf{Ntl}}_{min} > b^{\mathsf{Neg}}_{max} > b^{\mathsf{Neg}}_{min}$.

Example 2.4. Consider a set of views that use a rating scale range $[1, 10]$. In this case $b^{\mathsf{Neg}}_{min} = 1$ and $b^{\mathsf{Pos}}_{max} = 10$. Example boundaries in between this range could be $b^{\mathsf{Neg}}_{min} = 4$, $b^{\mathsf{Ntl}}_{min} = 5$, $b^{\mathsf{Ntl}}_{max} = 7$ and $b^{\mathsf{Pos}}_{min} = 8$.

We can now relate these three partitions to the three sets $\mathsf{Pos}(S), \mathsf{Ntl}(S), \mathsf{Neg}(S)$ in the graph space using a polynomial function that allows us to go from ratings to Agg scores.

Definition 2.8. Let $\sigma \in \{max, min\}$, $polarity \in \{\mathsf{Pos}, \mathsf{Ntl}, \mathsf{Neg}\}$ and $V = \{b^{\mathsf{Pos}}_{max}, b^{\mathsf{Pos}}_{min}, b^{\mathsf{Ntl}}_{max}, b^{\mathsf{Ntl}}_{min}, b^{\mathsf{Neg}}_{max}, b^{\mathsf{Neg}}_{min}\}$. We say the corresponding **aggregate value** for a boundary $b^{polarity}_{\sigma} \in V$ is given by $\Gamma(b^{polarity}_{\sigma}) = \sigma_{G \in polarity(S)}(\mathsf{Agg}(G))$. In the case that $\mathsf{Pos}(S)$ is a singleton set then $\Gamma(b^{\mathsf{Pos}}_{max}) = \mathsf{max}_{G \in \mathsf{Pos}(S)}(\mathsf{Agg}(G)) + \Delta Att$ and likewise if $\mathsf{Neg}(S)$ is a singleton set then $\Gamma(b^{\mathsf{Neg}}_{min}) = \mathsf{min}_{G \in \mathsf{Neg}(S)}(\mathsf{Agg}(G)) - \Delta Att$. We then say that the set of all corresponding **aggregate coordinates** is $AggCoordinates = \{(b, \Gamma(b)) | b \in V\}$.

Example 2.5. In Table 1 we have $AggCoordinates = ((10, 2), (8, -0.25), (7, -1), (5, -1.75), (4, -2.5), (1, -3.25))$.

With the coordinates *AggCoordinates* we can then fit our polynomial function which will enables us to map a rating to an aggregate score. We experimented with randomly generated graphs of different sizes and found that second order polynomials were sufficient for fitting to these coordinates.

Definition 2.9. Given aggregate coordinates *AggCoordinates* and a rating b we define a function $\mathsf{ratingToAgg} : b \to \mathbb{R}$ which is a second-order polynomial function $\mathsf{ratingToAgg}(b) = c_0 b^2 + c_1 b + c_2$ where $c_0, c_1, c_2 \in \mathbb{R}$. The coefficients $c_0, c_1, c_2 \in \mathbb{R}$ are learnt by fitting *AggCoordinates* to the polynomial using the least squares approximation method.

The $\mathsf{ratingToAgg}$ function provides an aggregate score for a view based on its rating. Using this we calculate the differences between this aggregate score and the aggregate scores of all of the graphs in the graph space. These differences serve as the basis for identifying a probability distribution. In essence we want those graphs that have a similar aggregate score to be assigned a larger probability mass.

Definition 2.10. Given a function $\mathsf{ratingToAgg}$ and a rating $b \in [b^{\mathsf{Neg}}_{min}, b^{\mathsf{Pos}}_{max}]$ we define a distance function $\mathsf{AggDist}(G,b) = \frac{1}{1+|\mathsf{Agg}(G) - \mathsf{ratingToAgg(b)}|^2}$. We then define a probability mass function for a graph G in graph space S as $P(G,b) = \frac{\mathsf{AggDist}(G,b)}{\sum_{G \in S} \mathsf{AggDist}(G,b)}$

Example 2.6. Table 1 shows two probability distributions for ratings in range $[1,10]$. In this example because there are more positive than negative arguments, and hence more graphs with a positive grounded extension, the probability mass is distributed across more graphs.

In this section we have defined a method for identifying a probability distribution for a view using the intuition that the rating provided in a view is a proxy for understanding the agent's belief in the polarity of the winning arguments.

3. Refining a Probability Distribution Using Impacts

In the previous section we identified a probability distribution for a view based on the rating alone. In this section we propose improving this distribution by incorporating real data about arguments derived from a set of views. We propose a simple measure which captures the general influence a particular argument has on a rating when it appears in a review.

Definition 3.1. Given a set of reviews *Rev*, and boundaries b^{Pos}_{max}, b^{Neg}_{min} an argument a, the set of reviews the argument appears in is given by $\mathsf{App}(a, Rev) = \{rev \in Rev | rev = (A, r) \,\&\, a \in A\}$. We denote the number of reviews it appears in as $N = |\mathsf{App}(a, Rev)|$. The sum of the ratings is then $\mathsf{sum}(a, Rev) = \sum_{(a,r) \in \mathsf{App(a,Rev)}} r - b^{\mathsf{Neg}}_{min}$. The impact of the argument is then given below.

$$\mathsf{Impact}(a, Rev) = \begin{cases} \dfrac{\mathsf{sum}(a, Rev)}{(b^{\mathsf{Pos}}_{max} - b^{\mathsf{Neg}}_{min}) \times N} & \text{if} \quad a \in A^{+} \\[2ex] 1 - \dfrac{\mathsf{sum}(a, Rev)}{(b^{\mathsf{Pos}}_{max} - b^{\mathsf{Neg}}_{min}) \times N} & \text{if} \quad a \in A^{-} \end{cases}$$

The impact of an argument tells us how much the argument caused the ratings of the reviews that it appeared in to move towards its polarity (positive or negative).

Example 3.1. Consider a set of reviews $Rev = \{(\{a,b,c\},9),(\{a,b,c\},8),(\{a,d\},7),(\{b,c\},2)\}$. where $A^+ = \{a,c\}$, $A^- = \{b,d\}$, $b^{\mathsf{Neg}}_{min} = 0$ and $b^{\mathsf{Pos}}_{max} = 10$. The impacts are then $\mathsf{Impact}(a,Rev) = 0.8$, $\mathsf{Impact}(b,Rev) = 0.63$, $\mathsf{Impact}(c,Rev) = 0.36$ and $\mathsf{Impact}(d,Rev) = 0.3$.

We interpret impact as a measure of relative strength of an argument. In the previous section we defined the relative strength of an argument using the normalised grade score. Hence in order to incorporate the impacts we weight those argument graphs whose normalised grade values resemble the impact values we have calculated.

Definition 3.2. Given a set of reviews Rev, a review $(A,r) \in Rev$, the corresponding graph space S for the review and a graph $G \in S$, the **similarity** between the impacts of the arguments A and their grades in graph G is given by $\mathsf{sim}(A,Rev,G) = \sqrt{\sum_{a \in A}(\mathsf{Impact}(a,Rev) - \mathsf{NormGrade}(G,a))^2}$.

Proposition 3.1. *For all* A,Rev,G, *it holds that* $0 \leq \mathsf{sim}(A,Rev) \leq \sqrt{|A|}$.

There is a natural correspondence between impact and graded score as they both are indicators of the degree of importance an argument plays in a graph/review. Hence when we find graphs where the difference between these values is small for all arguments we want to increase our probability assignment to such graphs.

Definition 3.3. Let (A,r) $Revs$ be a review, and S a graph space. Given a graph $G \in S$ we say that $d_G = \frac{1}{1+\mathsf{sim}(A,Rev,G))}$. The update weight associated with graph G is then $\mathsf{Weight}(G,r) = \frac{d_G \times P(G,r)}{\sum_{F \in S} d_F \times P(F,r)}$.

The weight assigned to each graph is thus the product of the probability of the graph and the inverse distance of the graph's grades to the argument's impacts. The normalising constant in the denominator ensures that the distribution of weights across the graph space is a probability distribution.

Example 3.2. Continuing from Example 3.1 if we now consider a review $(\{a,b,c\},9)$ we find that the largest weights are $\mathsf{Weight}(G_6,9) = 0.3$; this makes sense in this graph a has the highest grade followed by b and then c. We also see that $\mathsf{Weight}(G_6,3) = \mathsf{Weight}(G_6,3) = 0.10$ and that $\mathsf{Weight}(G_2,3) = \mathsf{Weight}(G_5,3) = 0.05$.

In this section we have proposed a method for incorporating data taken from sets of reviews to be able to identify probability distributions that better reflect the ratings in the reviews.

4. Experiment

In this section we demonstrate our framework using a set of reviews taken from the Drugs.com website. The dataset contains 601 reviews pertaining to the condition acne where each review contains a textual review and a rating between 1 and 10. In order to identify positive and negative arguments the first author identified arguments in the text for each review and assigned each argument a label (e.g. ‘bearable side effect’ etc) that best described it. Each label therefore denotes a different type of argument. In total 41 argument labels were used with a total of 2000 arguments being identified from all reviews. Following this we took 29 reviews and asked two annotators (neither of whom were authors) to provide an argument graph for each review using the identified arguments. We note that our paper is not intended as an argument mining framework and hence we are not focused on evaluating the quality of the argument labels, rather we want to evaluate our proposal for predicting an appropriate argument graph.

To report inter-annotator agreement we measured the degree of overlap between the grounded extensions of between the annotator's graphs. Popular inter-annotators agreement measures, such as Kappa-score, were not used as these measures are suitable for binary/categorical annotations and not graph structures.

Definition 4.1. For an actual graph graph G and a predicted graph $\hat{G}$, the **extension performance** is given by the function $\mathsf{GroundedPerformance} = \frac{|\mathsf{gr}(G)\setminus\mathsf{gr}(\hat{G})|+|\mathsf{gr}(\hat{G})\setminus\mathsf{gr}(G)|}{|\mathsf{gr}(G)|+|\mathsf{gr}(\hat{G})|}$

The function GroundedPerformance is 0 when the both graphs have exactly the same extension and 1 when they share no arguments in common. The average GroundedPerformance between the annotators was 0.16. To produce the final dataset annotators were asked to resolve conflicts in annotation between themselves.

We used this annotated data [1] to evaluate our approach in a two part experiment.. In the first part we trained a machine learning model to predict ratings for reviews using the full dataset and thus illustrate that it is possible to train such models to predict ratings for reviews that do not have any. In the second step we identified a probability distribution over the constellation of possible graphs for each of the 29 dual annotated reviews and sample from the distributions in order to predict a graph for each review. We measured the performance of our approach by comparing our predicted graphs to the graphs acquired through the annotators. Hence we required independent annotators for the argument graphs given a set arguments but not for the identification of those arguments.

4.1. Predicting Ratings for Reviews

We trained a 2-layer feed-forward multi-layer neural network to predict ratings for each review. We modelled each review as a binary vector of arguments. Our architecture consisted of 250 neurons in the hidden layers and a single neuron in the output layer. We chose a softmax activation function for the hidden layer and a linear activation function for the output layer. The model was trained using standard backpropogation with a mean-square-error loss function. All of our code was implemented using the Keras Python library. We used a training: validation split of 80:20 for our dataset.

After 150 iterations of training we achieved a mean absolute percentage error (MAPE) of 30.86 %. MAPE is a standard loss measurement when training regression models; it is defined as $\frac{100\%}{n}\sum_{t-1}^{n}\left|\frac{X_t-Y_t}{X_t}\right|$ where X_t is the ground truth, Y_t the predicted value and n the number of datapoints. Our reported MAPE suggests the model can generally predict near to the correct rating thus suggesting that their is a correlation between the polarity of arguments and the ratings. The hardest ratings to predict were the ratings between 4 and 6. This we believe is partly due to the quality of the original reviews; a number of times it was noted that the arguments in a review did not always match the rating provided.

[1] https://github.com/robienoor/constellationsDataReviews

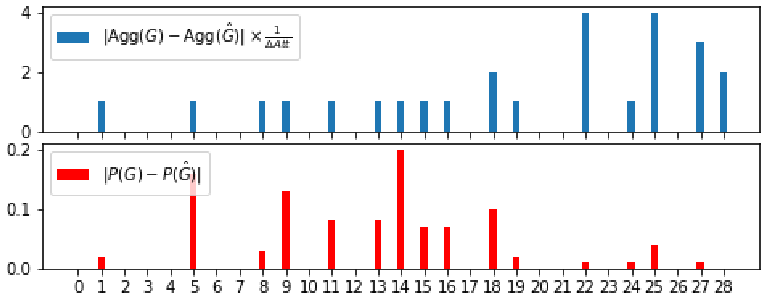


Figure 3. Performance results where each tick on the x-axis represents one of the 29 annotated graphs. The upper graph shows the aggregate distance between actual graph G and predicted graph $\hat{G}$. The bottom graph shows how far G was from $\hat{G}$ in terms of probabilities .

4.2. Predicting Graphs for Reviews

In this section we discuss the process of predicting argument graphs for reviews. We used the 29 argument graphs acquired from the annotators. For each review we identified a probability distribution using our approach and then sampled from this distribution in order to assign a graph to the review. For sampling we took the graph with the highest probability. In the case that we returned multiple graphs we simply randomly sample from the returned set of graphs.

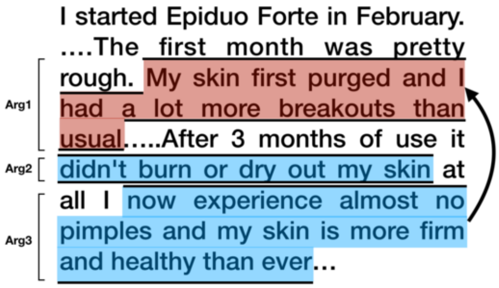
In order to measure the performance of our model we used two measurements in addition to GroundedPerfomance. For the first additional measure, we took the difference between the aggregate score of the predicted graphs and the actual graph. As per Definition 2.6 the aggregate scores for graphs in a graph space differ in units of ΔAtt. Graphs that share the same aggregate score are thus viewed as effectively belonging to the same equivalence class. This is captured by the following function that measures the number of equivalence classes by which the actual and predicted graph differ by.

Definition 4.2. Given a graph space S and set $Aggs = \{\mathrm{Agg}(G) | G \in S\}$ and a ground truth graph G and predicted graph $\hat{G}$ s.t. $G, \hat{G} \in S$ we define an **aggregate distance function** $\mathrm{AggDist}(G, \hat{G}) = \frac{|\mathrm{Agg}(G) - \mathrm{Agg}(\hat{G})|}{\Delta Att}$.

Example 4.1. Consider the example in Table 1 where $\Delta Att = 0.75$ and assume $G = G_1$ and $\hat{G} = G_{10}$. $\mathrm{AggDist}(G, \hat{G}) = 2.25/0.75 = 3$.

For the second additional measure we took the difference between the probability of the predicted graph and the actual graph. The results for the aggregate measurement and the probability measurement are depicted in Figure 3.

We found that the average GroundedPerformance was 0.30. In the cases where we identified an incorrect grounded extension we were either adding an additional argument or removing one and in other words we were not far off from the actual extension. In terms of aggregate distance we were never far off in terms of equivalence class as can be see in Figure 3 and likewise for the probability. Figure 4 depicts a review, in which three argument types where identified, and the attacks where assigned using our probabilistic model.



I started Epiduo Forte in February. ....The first month was pretty rough. My skin first purged and I had a lot more breakouts than usual.....After 3 months of use it didn't burn or dry out my skin at all I now experience almost no pimples and my skin is more firm and healthy than ever...

Arg1
Arg2
Arg3

Figure 4. A shortened review for the acne drug Epiduo with arguments annotated. Three arguments were identified. The graph sampled from the corresponding graph space is depicted above with Arg3 attacking Arg1

We have demonstrated in this section the end-to-end process of using our framework for predicting argument graphs for reviews. We started by demonstrating that ratings could be reasonably predicted for reviews by using off the shelf machine learning algorithms. We then used our framework to identify probability distributions for each review before finally sampling from this distribution to predict the correct graph for the review.

5. Related Works

In another proposal for generating probability distributions over constellations of arguments graph [1] it is assumed that an agent(s) specifies a belief in the acceptability status of arguments. Using this data the paper proposes methods for aggregating, combining and summarising these beliefs. Whilst related to this paper, we have a different starting point which is that we do not have access to such beliefs directly rather we have access to ratings which we process to produce a distribution over a set of argument graphs.

There have been a few proposals for argument graphs learning algorithms when in/out/un labellings are provided by agents. In [3] a learning algorithm is proposed which takes as input a probability distribution over a set of in/un/out labellings. The algorithm is an on-the-fly algorithm to aggregate these labellings into a weighted argumentation graph. In our case we deal with a setting in which we do not have access to such labellings and furthermore we produce a distribution over a constellation of argument graphs. Likewise [4] makes a similar starting assumption in that the algorithm begins which a set of labellings for each argument. A Bayesian approach is proposed in order to learn from these labellings a posterior distribution for a set of arguments being in an extension. Both of these papers differ from our approach in that we do not assume we have such labelled data rather. Another proposal in [2] provides a method for extracting bipolar argument frameworks from a set of movie reviews. Each review contains a textual review and a binary rating indicating whether the reviewer thought the movie was good or bad. The proposed algorithm produces a quantitative bipolar argument per review which differs from our probabilistic output.

Various proposals for capturing and aggregating views taken from the social web have also been made [6][5]. These proposals use judgement aggregation and voting mechanisms to produce the aggregation which differs from our approach which produces probabilistic interpretation of views.

In summary our proposal differs primarily from the existing literature in that it is driven by our interpretation of ratings. The notion of rating is not dealt with explicitly in the literature and certainly not in a probabilistic context.

6. Discussion

In this paper we have proposed a methodology for identifying a probability distribution for a review. In the *identification step* this is done by exploiting the relationship between the rating and the accepted arguments in that review. We considered a situation in which we deal with bipartite argument graphs but this could be generalised to handle multi-partite graphs. We further provided a *refinement step* for utilising information extracted from a set of reviews so as to enrich the identified probability distribution. We illustrated our approach using an annotated dataset and highlighted how machine learning models can be employed to provide ratings for reviews without ratings.

In future work we wish to ensure that our proposal is scalable given that the constellations approach can be computationally challenging [11]. We intend to do this by developing an understanding of the underlying combinatorics as well as the potential of approximation techniques. We also wish to experiment with other implementations of the grading function to see if we can improve the distribution. Likewise we wish to explore the use of additional acceptability semantics in order to enrich our function for partitioning the graph space based on polarity.

References

[1] Hunter, Anthony, and Kawsar Noor. "Aggregation of Perspectives Using the Constellations Approach to Probabilistic Argumentation." Proceedings of the AAAI '20.

[2] Cocarascu, Oana, Antonio Rago, and Francesca Toni. "Extracting dialogical explanations for review aggregations with argumentative dialogical agents." Proceedings of AAMAS '19.

[3] Riveret, Régis, and Guido Governatori. "On learning attacks in probabilistic abstract argumentation." Proceedings of AAMAS '16.

[4] Kido, Hiroyuki, and Keishi Okamoto. "A Bayesian Approach to Argument-Based Reasoning for Attack Estimation." Proceedings of IJCAI '17.

[5] Leite, Joao, and Joao Martins. "Social abstract argumentation." Proceedings of IJCAI '11.

[6] Noor, Kawsar, Anthony Hunter, and Astrid Mayer. "Analysis of medical arguments from patient experiences expressed on the social web." Proceedings of IEA/AIE '17.

[7] Dung, Phan Minh. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games." Artificial intelligence 77.2 (1995): 321-357.

[8] Li, Hengfei, Nir Oren, and Timothy J. Norman. "Probabilistic argumentation frameworks." Proceedings of TAFA '11.

[9] Hunter, Anthony. "Some foundations for probabilistic abstract argumentation." Proceedings of COMMA '12.

[10] Bonzon, Elise, et al. "A comparative study of ranking-based semantics for abstract argumentation." Proceeding of AAAI '16.

[11] Fazzinga, Bettina, Sergio Flesca, and Francesco Parisi. "Efficiently estimating the probability of extensions in abstract argumentation." Proceedings of SUM '13.