Using Argumentation to improve classification in Natural Language problems

LUCAS CARSTENS, Imperial College London
FRANCESCA TONI, Imperial College London

Argumentation has proven successful in a number of domains, including Multi-Agent Systems and decision support in medicine and engineering. We propose its application to a domain yet largely unexplored by argumentation research: Computational linguistics. We have developed a novel classification methodology that incorporates reasoning through argumentation with supervised learning. We train classifiers and then *argue* about the validity of their output. To do so we identify arguments that formalise prototypical knowledge of a problem and use them to correct misclassifications. We illustrate our methodology on two tasks. On the one hand we address *cross-domain sentiment polarity classification*, where we train classifiers on one corpus, e.g. Tweets, to identify positive/negative polarity, and classify instances from another corpus, e.g. sentences from movie reviews. On the other hand we address a form of argumentation mining that we call *Relation-based Argumentation Mining*, where we classify pairs of sentences based on whether the first sentence attacks or supports the second, or whether it does neither. Whenever we find that one sentence attacks/supports the other we consider both to be argumentative, irrespective of their stand-alone argumentativeness. For both tasks we improve classification performance when using our methodology, compared to using standard classifiers only.

Additional Key Words and Phrases: Argumentation, Abstract Argumentation, Quantitative Argumentation, Computational Linguistics, Argumentation Mining, Sentiment Analysis

ACM Reference Format:
Lucas Carstens and Francesca Toni, 2016. Using Argumentation to improve classification in Natural Language problems. *ACM Trans. Embedd. Comput. Syst.* V, N, Article A (March 2016), 22 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

We argue every day; we attempt to impart our views and understanding of things on others, and we use language as the prime vehicle of doing so. Accordingly, argumentation has been studied extensively by a variety of scholars, not least by philosophers and computational linguists, who, amongst other problems, have studied the way arguments are communicated, and how they interact in language, e.g. as in [Toulmin 2003; Van Eemeren and Grootendorst 2003]. Over the past decade or so advances in computational linguistics have commonly relied on *data-driven* solutions, as in e.g. [Collobert et al. 2011; Mikolov et al. 2013; Socher et al. 2013] The field of *computational argumentation* (henceforth called just *argumentation*), on the other hand, has emerged as a discipline concerned with developing formal, logic-based accounts of arguments, e.g. as in [Dung 1995; Kowalski and Toni 1996; Modgil et al. 2013]. Recently, as illustrated by e.g. [Hogenboom et al. 2015], there appears to be growing consent that, in order to achieve significant advances in artificial intelligence, in general, and computational linguistics, in particular, we may need to consider data- and knowledge-driven approaches in unison. To this end we propose a novel classification methodology, called ***Cl****assification* ***e****nhanced with* ***Ar****guments (CleAr)*, in which we use argumentation, together with supervised learning, to consolidate these streams of research on arguments into a single classification procedure. This allows us to integrate data-driven approaches to computational linguistics, i.e. machine learning, with more knowledge-heavy approaches, i.e. argumentation. We consult knowledge specific to the problem at hand to *argue* about the validity of class labels proposed by a classifier. Whenever we identify sufficiently strong arguments, arguing against a class label suggested by a classifier or for a different label, we overrule the classifier and instead base our classification decision on what the arguments tell us.

In this paper, in addition to presenting CleAr, we also describe an instantiation thereof to tackle *Relation-based Argumentation Mining (RbAM)*, an approach to argumentation mining where we classify sentence pairs based on whether or not we identify argumentative relations between sentences. As part of our work on RbAM we describe a corpus of around $2,200$ hand-annotated sentence pairs. Finally we describe an instantiation of CleAr for *cross-domain sentiment polarity classification* to incorporate generic sentiment knowledge as arguments to decrease the negative impact of domain dependence in classification. We evaluate CleAr for RbAM and polarity classification on a variety of domains, ranging from news articles to movie reviews and Tweets. In summary, with the work presented in this paper we aim to verify or disprove the following hypotheses:

— The use of CleAr improves classification performance when applied to cross-domain sentiment polarity classification
— The use of CleAr improves classification performance when applied to Relation-based Argumentation Mining

We test each of these hypotheses for two different versions of CleAr, where we use different argumentation frameworks, *Extended Social Abstract Argumentation (ESAA)* [Evripidou and Toni 2014] and the *Discontinuity-Free Quantitative Argumentation Debate (DF-QuAD)* [Rago et al. 2016], to instantiate our approach.

This paper builds upon and extends preliminary work as follows. The concepts underpinning CleAr were introduced in [Carstens and Toni 2015a], but in an informal manner. Here we also describe initial results of applying CleAr to sentiment polarity classification. We conceptually described RbAM in [Carstens and Toni 2015b], focusing on motivation for developing RbAM as an alternative to conventional argumentation mining approaches. With this paper we introduce formal definitions for both CleAr and RbAM, building on the underpinnings of [Carstens and Toni 2015b], and we provide extensive evaluations of the two.

The remainder of this paper is organised as follows. In section 2 we first review relevant background in argumentation and corpus-based computational linguistics. We then formally define CleAr as a four-step classification procedure in section 3. Based on this we describe instantiations and evaluations of CleAr for RbAM and Polarity classification in sections 4 and 5, respectively. Before concluding the paper in section 7 we give a brief summary of related work in section 6.

2. BACKGROUND

2.1. Argumentation

An *Abstract Argumentation Framework (AF)* [Dung 1995] consists of a set, whose elements are referred to as arguments, and a binary *Attack* relation between arguments, representing conflicts. Formally, an AF is a pair $< Args, Attack >$ where $Attack \subseteq Args \times Args$ and $(A, B) \in Attack$ is read 'A *attacks* B'. *Bipolar Argument Frameworks (BAF)* [Amgoud et al. 2008; Cayrol and Lagasquie-Schiex 2005] additionally use a support relation to complement the original definition by Dung. Formally, a BAF is a triple $< Args, Attack, Support >$ where $Support \subseteq Args \times Args$ is an additional binary relation compared to Dung-style AFs. Both AFs and BAFs can be represented as directed graphs, in which nodes correspond to arguments and directed arcs correspond to attacks, as well as supports in the case of BAFs.

Quantitative BAFs, e.g. the *Extended Social Abstract Argumentation (ESAA) Framework* [Evripidou and Toni 2014] and the *Discontinuity-Free Quantitative Argumentation Debate (DF-QuAD) Framework* [Rago et al. 2016], assume that arguments are equipped with a *base score*, i.e. a real number in $[0, 1]$. The base score amounts to an intrinsic (non-dialectical) strength of arguments, which is then altered to give the final (dialectical) strength, based on the (dialectical) strength of attacking and supporting arguments. The resulting strength of

arguments is determined by aggregating the strength of attackers against and supporters for these arguments, for restricted types of frameworks that can be represented as trees.

Formally an ESAA framework is a tuple $< Args, Attack, Support, S^0 >$ where $< Args, Attack, Support >$ is a BAF, extended with a function $S^0 : Args \rightarrow [0, 1]$, assigning a base score to arguments. A DF-QuAD framework is a tuple $< Ans, Pro, Con, R, S^0 >$ where Ans is a set of possible answers to a question and Pro and Con are two types of arguments, attacks and supports, respectively. R is a relation between arguments $R \subseteq (Pro \cup Con) \times (Pro \cup Con \cup Ans)$ and $S^0 : Ans \cup Pro \cup Con \rightarrow [0, 1]$ is a function that associates a base score with answers and arguments. A DF-QuAD framework $< Ans, Pro, Con, R, S^0 >$ can be understood as an ESAA framework $< Args, Attack, Support, S^0 >$ by setting $Args = Ans \cup Pro \cup Con$, $Attack = \{(X, Y)|X \in Con, (X, Y) \in R\}$ and $Support = \{(X, Y)|X \in Pro, (X, Y) \in R\}$. Both approaches then define strength $S : X \rightarrow [0, 1]$ where X is $Args$ in ESAA and $Ans \cup Pro \cup Con$ in DF-QuAD, assigning a score to answers and/or arguments, taking S^0, as well as supports and attacks, into account. Let v_0 be the base score of an argument A, and v_{att}, v_{sup} the combined strength of all attackers against A and supporters for A, respectively. Then the dialectical strength of A is given, in ESAA, by $g(v_0, v_{att}, v_{sup})$ defined as follows:

$$g(v_0, v_{att}, v_{sup}) = \begin{cases} v_{att} \text{ if } v_{sup} = nil \text{ and } v_{att} \neq nil; \\ v_{sup} \text{ if } v_{att} = nil \text{ and } v_{sup} \neq nil; \\ v_0 \text{ if } v_{att} = v_{sup} = nil; \\ \frac{(v_{att} + v_{sup})}{2} \text{otherwise.} \end{cases}$$

Here $v_{att} = nil$/ $v_{sup} = nil$ if there are no attackers against/supporters for (respectively) A or all such attackers/supporters have strength 0 (and are thus ineffective [Baroni et al. 2015]). The combined strength v_{att}/v_{sup} of (the sequence S of the strength of) all attacks against/supports for A is computed as $F_*(v_0, S)$, for $* = att$ or $* = sup$ (respectively), defined recursively as follows, in ESAA:

$$\begin{aligned} \text{if } S \text{ is ineffective} &: \quad \mathcal{F}_*(v_0, S) = nil \\ \text{if } S = (v) &: \quad \mathcal{F}_*(v_0, S) = f_*(v_0, v) \\ \text{if } S = (v_1, \ldots, v_n) &: \quad \mathcal{F}_*(v_0, (v_1, \ldots, v_n)) = \\ & \qquad f_*(\mathcal{F}_*(v_0, (v_1, \ldots, v_{n-1})), v_n) \end{aligned}$$

where[1]

$$\begin{aligned} f_{att}(v_0, v) &= v_0 - v_0 \cdot v = v_0 \cdot (1 - v) \\ f_{sup}(v_0, v) &= min(v_0 + (v_0 - v_0 \cdot (1 - v)), 1) \end{aligned}$$

Whereas in ESAA the strengths of attacks (F_{att}) and supports (F_{sup}) are aggregated using separate functions, DF-QuAD defines a single strength aggregation function:

$$\begin{aligned} \text{if } n = 0 &: \quad \mathcal{F}(S) = 0 \\ \text{if } n = 1 &: \quad \mathcal{F}(S) = v_1 \\ \text{if } n = 2 &: \quad \mathcal{F}(S) = f(v_1, v_2) \\ \text{if } n > 2 &: \quad \mathcal{F}(S) = f(\mathcal{F}(v_1, ..., v_{n-1}), v_n) \end{aligned}$$

where

$$f(v_1, v_2) = v_1 + (1 - v_1) \cdot v_2 = v_1 + v_2 - v_1 \cdot v_2.$$

[1]Here f_{sup} is as in [Carstens et al. 2015], correcting a typo in the original publication [Evripidou and Toni 2014]

Once strengths of arguments for and against an argument have been aggregated using $\mathcal{F}$, the dialectical strength of an argument with base score S^0 and combined strengths of attacks and supports v_{att} and v_{sup}, respectively, is calculated as follows:

$$g(v_0, v_{att}, v_{sup}) = v_0 - v_0 \cdot |v_{sup} - v_{att}| \quad if \;\; v_{att} \geq v_{sup}$$
$$g(v_0, v_{att}, v_{sup}) = v_0 + (1 - v_0) \cdot |v_{sup} - v_{att}| \quad if \;\; v_{att} < v_{sup}$$

Refer to [Carstens et al. 2015] for details and example calculations for ESAA and to [Rago et al. 2016] for the same on DF-QuAD.

2.2. Corpus-based Computational Linguistics

In corpus-based computational linguistics we focus on solving natural language problems through supervised learning approaches. This broadly means that we take a corpus of instances of text, e.g. sentences, each assigned to one of a predefined set of classes, and train a statistical classification model on this corpus. This model is then used to classify unseen instances according to patterns that were identified to occur more often in one class than any of the other classes in the training corpus. In argumentation mining, for example, this may mean building solutions based on corpora where instances are labeled to be either arguments or non-arguments, as in e.g. [Moens et al. 2007].

To classify such instances, using common classifiers like *Random Forests (RF)* [Breiman 2001], *Support Vector Machines (SVM)* [Cortes and Vapnik 1995] and *Naive Bayes (NB) classifiers* [John and Langley 1995], we need to represent them numerically. A common approach to doing so is to use a *Bag-of-Words (BOW)*. For example, with *binary BOW* [Salton and McGill 1986] representation of text we simply track the occurrence of words in each instance. Every instance is represented by the same amount of features, one for each word appearing in the vocabulary of the training corpus used. If, for example, our training corpus contains $10,000$ unique words, the feature vector representing each instance will be comprised of $10,000$ features, also. Each feature is instantiated to a value of either 1 or 0, depending on whether the word represented by a feature occurs in an instance or whether it does not, respectively. Alternatives to this representation include *term frequency - inverse document frequency (tf-idf)* and representations based on *feature engineering*, as in e.g. [Moens et al. 2007; Rooney et al. 2012], where we aim to identify a set of semantic and/or syntactic characteristics of text that are conducive towards distinguishing classes and that we can represent numerically.

3. CLASSIFICATION ENHANCED WITH ARGUMENTATION (CLEAR)

From its early days in the 1940's until well into the 1990's, computational linguistics was dominated by logic- and rule-based solutions [Cambria and White 2014; Nadkarni et al. 2011]. With the advent of ever increasing computing power, however, more data-driven approaches came into focus. Building statistical models based on ever larger datasets gained traction and more knowledge-heavy solutions began to fall out of favour. More recently, however, a rekindling with the methods of old has been taking place [Hogenboom et al. 2015], indicating that, after a prolonged period of placing data at the centre of computational linguistics, certain problems may call for a consolidation of the two streams of research.

To this end we have developed a novel classification methodology that incorporates reasoning through argumentation with corpus-based computational linguistics: ***Cl**assification **e**nhanced with **Ar**guments (CleAr)*. Broadly we take a supervised classifier, such as RF, and use argumentation to enhance its performance. As a first step we classify an instance, using the model built by the classifier, e.g. determining a sentence to exhibit negative or positive polarity. We then use Argumentation to *argue* on the validity of this class label. We do this by consulting a collection of arguments, an *argument base (AB)*, for those applicable to the instance in question. To classify an instance we first apply a conventional classification model,

trained on an annotated corpus, which provides us with a class label. We then argue about whether the suggested class label is appropriate by identifying arguments that apply to the instance in question. These arguments may then strengthen the class label suggestion, they may weaken it, or they may strengthen/weaken other class labels. If the suggested class label is sufficiently weakened, or another class label suggestion strengthened, we overrule the suggestion made by the classifier and base the classification on what the arguments tell us. The classification procedure is comprised of the following four steps, described in detail below, for a given instance I:

(1) Classify I and assign scores to possible class labels, using a given classification model
(2) Identify arguments and relations *applicable* to instance I within a given *argument base* AB
(3) Calculate dialectical strength of class labels according to an Argumentation algorithm, e.g. ESAA or DF-QuAD, for the AF built in step (2) by instantiating AB for I
(4) Make final classification decision, based on dialectical strength calculated in step (3)

3.1. Step (1): Instance classification

To make a classification decision for a particular instance I we take as a starting point the output of a classifier C for I. We assume that the output is the set of all possible class labels for the instance at hand: $A = \{a_1, ..., a_n\}$, where $n \geq 2$. We further assume that each a_i in A is assigned a score P_{a_i} by C, with $\sum_{i=1}^{n} P_{a_i} = 1$. If C is a probabilistic classifier we use the posterior probabilities provided by the classifier as scores. Though classifiers such as SVM are non-probabilistic, ways of assigning posteriors to their output have been proposed, e.g. by [Gönen et al. 2008; Platt 1999]. As a simple, ad-hoc alternative we may assign predefined scores to class labels where the label chosen by the classifier receives a higher initial score than the other possible labels in A. We may, for example, use F_1 measures obtained during the training of the classifier to inform the choice of of these initial scores. In this case we assign $P_{a_i} = F_1$ to the a_i selected by the classifier and $P_{a_j} = \frac{1-F_1}{n-1}$ to each a_j not selected by the classifier. The output of step (1) is thus the set of all possible class labels, each of which has a score assigned.

To exemplify, imagine we are considering whether or not to book a room at a hotel h. We then try to classify instance $I = h$, where the set of all possible class labels is $A = \{yes, no\}$. Say our classifier determines that we should book h, and it does so with a confidence 0.6. We would then assign scores to our possible labels as $P_{yes(h)} = 0.6$ and $P_{no(h)} = 0.4$. Here we use $yes(h)$ and $no(h)$ to indicate that labels yes/no are under consideration for instance $I = h$.

3.2. Step (2): Argument identification

In step (2) we use arguments to deliberate on whether or not class labels suggested by the supervised classifier in step (1) are correct. The argument base AB is a collection of the arguments we use to do so, where each argument formalises some sort of knowledge conducive to solving the problem in question. We assume that AB is comprised of arguments of the form

$$id(I, I') : Premise(I, I') \Rightarrow Conclusion(I, I')$$

where I' refers to optional additional information, $id(I, I')$ is an identifier univocally identifying the argument, $Premise(I, I')$ formalises some domain knowledge about I and $Conclusion(I, I')$ may be one of the possible class labels in I or another argument's premise. In addition to its premise and conclusion, each argument is defined by three metrics:

(i) A base score $S^0_{id(I,I')} \in [0, 1]$
(ii) An argument type $T_{id(I,I')} \in \{Pro, Con\}$
(iii) A relation $R_{id(I,I')}$ that determines which class label or other arguments it relates to

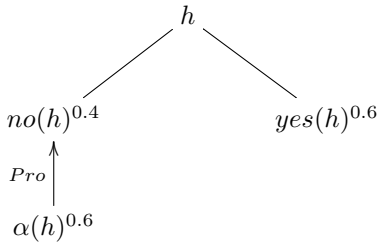


Fig. 1: Tree view of the output of step (2) for the hotel example

Note that these arguments are templates to be instantiated when I and possibly I', if present, are known. We consider an argument defined this way to be *applicable* to an instance I whenever its premise holds for I. To continue our example, say we have an AB concerned with hotels, and three of its arguments go as follows:

$$\alpha(I): \ I \ has \ star \ rating \ of \ 1 \Rightarrow I \ is \ labeled \ as \ no$$
$$\beta(I): \ I \ is \ in \ a \ good \ neighbourhood \Rightarrow I \ is \ labeled \ as \ yes$$
$$\gamma(I, SR): \ Expert \ gave \ star \ rating \ SR \ of \ I \ \Rightarrow I \ has \ star \ rating \ SR$$

Assume also that the argument metrics have the below values

$$S^0_{\alpha(I)} = 0.6; \ T_{\alpha(I)} = Pro; \ R_{\alpha(I)} = \{(\alpha(I), no(I))\}$$
$$S^0_{\beta(I)} = 0.3; \ T_{\beta(I)} = Pro; \ R_{\beta(I)} = \{(\beta(I), yes(I))\}$$
$$S^0_{\gamma(I,SR)} = 0.3; \ T_{\gamma(I,1)} = Pro; \ T_{\gamma(I,SR)} = Con \ if \ SR \neq 1; \ R_{\gamma(I,SR)} = \{(\gamma(I, SR), \alpha(I))\}$$

where *no(I)*/*yes(I)* are shorthand for *I is labeled as no*/*yes*, respectively. The scores S^0 here may, for example, reflect that it is more important to us that a hotel is highly rated than its being well located. However, note that the general problem of choosing scores for arguments is an on-going research question [Baroni et al. 2015]. SR in γ modulates the argument type based on the additional information provided. To determine the arguments' applicability to $I = h$ we need to match their premises to information about h. If h indeed has a one star rating, but we know nothing about its neighbourhood, nor do we know who provided the rating, we select and instantiate only $\alpha(I)$ from AB. $R_{\alpha(h)}$ tells us that $\alpha(h)$ is related to the label *no*. $T_{\alpha(h)} : Pro$, in turn, tells us that the relation to the label is one of support. The result of step (2) can be viewed graphically in tree form, as shown in figure 1. Note that the root node denotes the instance in question, the second layer represents possible class labels, each assigned their score, and subsequent nodes denote arguments and their base score.

The output of step (2) is thus an AF $< Ans, Pro, Con, R, S^0 >$ where $Ans = A$ and Pro/Con is the set of all applicable arguments in AB with $Pro \in T_{id(I,I')}/Con \in T_{id(I,I')}$, respectively. R is the set of all pairs (X, Y) such that $X, Y \in Pro \cup Con \cup Ans$ and (X, Y) is an instance, with respect to I, of some $R_{id}(I, I')$. For $X \in Pro \cup Con$ let S^0 be defined as $S^0(X) = S_{id(I,I')}$, where $id(I, I')$ is the unique identifier of argument X if $X \in Pro \cup Con$ and $S^0(X) = P_X$ if $X \in Ans$. This DF-QuAD framework can be represented alternatively as an ESAA framework, as we have seen in section 2.1.

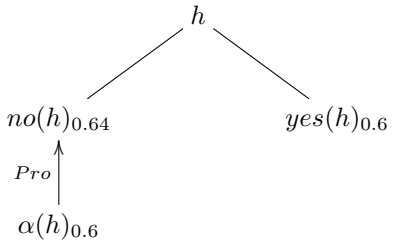


Fig. 2: Tree view of the output after executing step (3) for the hotel example, where only α is applicable to h. The subscripts denote the dialectical strengths, calculated using ESAA.

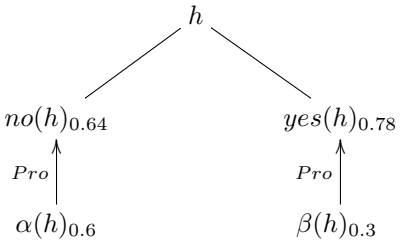


Fig. 3: Tree view of the output after executing step (3) for the hotel example, where both α and β are applicable to h. The subscripts denote the dialectical strengths, calculated using ESAA.

3.3. Step (3): Answer strength calculation

In this step we calculate the dialectical strength of each class label, based on the output of step (2). Figure 2 shows the result of calculating this strength for our example using the ESAA algorithm, and thus treating the output of step (2) as an ESAA framework, but note that we may use any other algorithm to execute step (3), as long as there is a possible mapping of the relevant metrics as we have given from DF-QuAD to ESAA. Also note that the strength S_{a_i} of a class label a_i should not be interpreted as a confidence value. Indeed, in general $\sum_{i=1}^{n} S_{a_i} \neq 1$. Also, we adopt a convention to show the dialectical strength of the arguments and class labels as subscripts to distinguish them from base score, which are shown as superscripts.

To continue our illustration, say we also knew that the hotel was in a good neighbourhood. Then the output of step (3) would be as shown in figure 3, where we have added the appropriate support $\beta(h)$ to $yes(h)$. We may also encounter arguments that are in relation with other arguments, instead of class labels, such as $\gamma(I, SR)$. If, for our example, we knew $SR = 1$ was provided for h by an expert, $\gamma(h, 1)$ would also apply and we would obtain as output of step (3) figure 4. Any further applicable arguments may be incorporated in the same manner.

3.4. Step (4): Final classification decision

To obtain our final classification we only need to consider the strength of all class labels in A after executing step (3). Let $S = \{S_{a_1}, ..., S_{a_n}\}$ where S_{a_i} is the strength of a_i resulting from step (3). For the instantiations of CleAr described in sections 4 and 5 the final classification decision is made by choosing the maximal element of S, but any other criteria, e.g. by setting thresholds, may be employed to choose the final class label. Should two or more class labels have the same strength after step (3) we use the label suggested by the classifier in step (1)

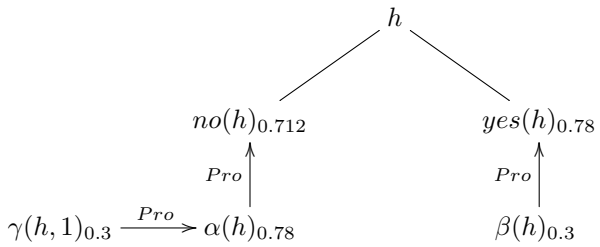


Fig. 4: Tree view of the output after executing step (3) for the hotel example, where α, β and γ are applicable to h. The subscripts denote the dialectical strengths, calculated using ESAA.

as a tie breaker. In our example we would hence choose no if arguments were applicable as in figure 2, thus overruling the original recommendation by the classifier, and yes if the situation were as shown in figures 3 and 4, thus upholding the original classification.

4. CLEAR FOR RELATION-BASED ARGUMENTATION MINING

Argumentation Mining is generally understood as a problem of (1) identifying arguments in text, as in e.g. [Boltuzic and Šnajder 2015; Moens et al. 2007; Palau and Moens 2009], and (2) connecting said arguments through attack and support relations, as in e.g. [Boltuzic and Šnajder 2014; Kirschner et al. 2015; Park and Cardie 2014]. With *Relation-based Argumentation Mining (RbAM)* [Carstens et al. 2014] we propose an alternative view, in which we focus solely on relations between sentences. In RbAM we classify pairs of sentences, a *Parent* and a *Child*, as belonging to one of three classes, *Attack (Att)*, *Support (Sup)* or *Neither (N)*, where Neither denotes everything other than an attack or support relation. Whenever we identify the Child to be an attack or support on the Parent, we consider both sentences to be argumentative, irrespective of their individual argumentative quality. Here we give an instantiation of CleAr for RbAM, describing each step in turn below.

4.1. Step (1): Instance classification

We first classify instance I, e.g. using a model trained on one of the corpora described in section 4.5, where I is a pair of sentences (P, C), such as the following pair $pc = (p, c)$:

> *p*= With her cautious attitude Mrs. M. will struggle as party leader!
> *c*= And we can all agree that her reserved character will not please voters!

For three-class RbAM the class labels are $A = \{Att, Sup, N\}$ while for two-class RbAM we we have $A = \{Att, Sup\}$.

4.2. Step (2): Argument identification

Once we have classified I we identify arguments and relations that are applicable to the instance in question, as described in section 3.2. We have built an argument base AB_{RbAM} that currently consists of 172 arguments and contains arguments from 5 categories:

(1) Keyword (73 arguments)
(2) First word (11 arguments)
(3) Sentiment (3 arguments)
(4) Similarity (12 arguments)
(5) Negation (73 arguments)

Category	id	Premise	Conc.	S^0	T	R
FirstWord	$\phi_1(I)$	C in I *starts with disagree*	$I \in Att$	0.5	Pro	$\{(\phi_1(I), \kappa_1(I))\}$
	$\phi_2(I)$	C in I *starts with and*	$I \in Att$	0.5	Pro	$\{(\phi_2(I), \kappa_2(I))\}$
Keyword	$\kappa_1(I)$	C in I *contains disagree*	$I \in Att$	0.5	Pro	$\{(\kappa_1(I), Att)\}$
	$\kappa_2(I)$	C in I *contains and*	$I \in Sup$	0.5	Pro	$\{(\kappa_2(I), Sup)\}$
Sentiment	$\sigma_1(I)$	C in I has neg. sentiment	$I \in Att$	0.5	Pro	$\{(\sigma_1(I), Att)\}$
	$\sigma_2(I)$	C in I has pos. sentiment	$I \in Sup$	0.5	Pro	$\{(\sigma_2(I), Sup)\}$
	$\sigma_3(I)$	P in I has neg. sentiment	$Pr(\sigma_1(I))$	0.5	Con	$\{(\sigma_3(I), \sigma_1(I))\}$
Similarity	$\theta_1(I)$	Sim $>.1$	$I \in Att$	0.5	Pro	$\{(\theta_1(I), Att)\}$
	$\theta_2(I)$	Sim $=<.1$	$I \in Att$	0.5	Con	$\{(\theta_2(I), Att)\}$
	$\theta_3(I)$	Sim $=<.1$	$Pr(\sigma_1(I))$	0.5	Con	$\{(\theta_3(I), \sigma_1(I))\}$
	$\theta_4(I)$	Sim mainly stopwords	$Pr(\theta_1(I))$	0.5	Con	$\{(\theta_4(I), \theta_1(I))\}$
	$\theta_5(I)$	Sim mainly nonStopwords	$Pr(\theta_1(I))$	0.5	Con	$\{(\theta_5(I), \theta_1(I)))\}$
Negation	$\nu_1(I)$	*disagree* is negated in I	$Pr(\kappa_1(I))$	0.5	Con	$\{(\sigma_1(I), \kappa_1(I)\}$

Table I: Example arguments taken from the argument base AB_{RbAM}, where $I \in Att/Sup$ stands for instance I belonging to Att/Sup, $Pr(X)$ stands for the premise of argument X and Sim stands for $Similarity$ between P and C

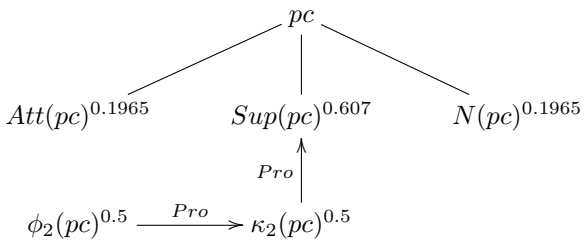


Fig. 5: Tree view of output of step (2) for RbAM for example pair pc

We show examples from each category in table I. Arguments may support/attack either the class labels Att and Sup or other arguments from AB_{RbAM}. When arguments support/attack another argument they do so by supporting/attacking the other argument's premise. Note that at present there are no arguments in AB_{RbAM} attacking or supporting the N class label.

Arguments from the *Keyword* category are concerned with discourse markers appearing in either the Parent P or the Child C. With the *FirstWord* category we additionally check whether the discourse marker is the first word of the Child sentence, further supporting the notion of a relation being present. In the above example pair pc, the keyword *And* appears as the first word in C. This renders two arguments applicable, namely $\kappa 2(I)$ and $\phi 2(I)$ shown in table I. Indeed the output of step (2) for $I = pc$ is shown in figure 5. To exemplify we assume that in step (1) the instance was labeled with $A = Sup$ and that $P_{Sup(pc)}$ was 0.607.

For the *Sentiment* category we use the Stanford CoreNLP sentiment library [Socher et al. 2013] to classify the individual sentiment of C and P. We have found that the sentiment of C is a good indicator of the argumentative relation to another sentence, if one is present, which may be mediated by P's sentiment. The *Similarity* category contains various arguments reflecting different types of relatedness between P and C. The degree of similarity is measured using Wu and Palmer's WordNet similarity measure [Wu and Palmer 1994] and we argue for or against relatedness depending on whether this similarity is above or below a threshold

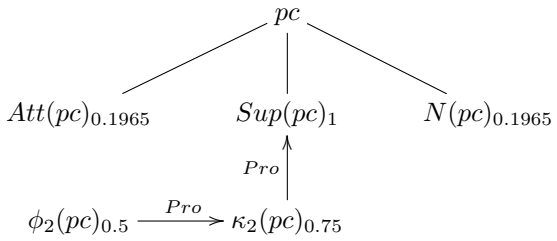


Fig. 6: Tree view of output of step (3) for RbAM for example pair pc

of 0.1, respectively (see $\theta_1(I), \theta_2(I), \theta_3(I)$). We have chosen this threshold by empirically determining the average relatedness measure for a held out collection of sentence pairs, split by class. Whenever the similarity surpasses the determined threshold arguments apply that support both the labels Att and Sup, indicating that some relation is present. When the similarity is below the threshold, on the other hand, arguments that attack the same class labels apply. We further argue about where the similarity between P and C stems from, e.g. stop words or non-stop words (see $\theta_4(I), \theta_5(I)$). Whenever the majority of words that contribute to a high degree of relatedness are stop words an argument applies that attacks the similarity arguments. Finally we have arguments dealing with negation, such as $\nu 1(I)$ in table I. A keywords's sentiment will broadly either be neutralised (e.g. *not awful*) or reversed (e.g. *not good*) when negated. To reflect this, the presence of negations is formalised as an attack on applicable keyword arguments. Note that currently all arguments in AB_{RbAM} have the same score of 0.5. This is not because we believe that they should be, but rather a choice owed to the current lack of ways to make informed choices for scores. How we may render score assignment more flexible remains an open research question; one possible option may be to extend existing approaches to determining the strength of arguments in abstract argumentation frameworks without requiring any initial scores, as, for example, in [Matt and Toni 2008], so that support is also taken into account. Another direction may be to study relationships between quantitative semantics for argumentation and fuzzy logic, as, for example, in [Alsinet et al. 2008].

4.3. Step (3): Answer strength calculation

In the third step of CleAr we determine the strength of each possible class label by including the applicable arguments identified in step (2), as described in section 3.3. The output of step (3) for our running example is shown in figure 6, where the dialectical strength has been calculated for all nodes by applying the ESAA method. We compare two ways of calculating answer strength for RbAM, where on the one hand we use the ESAA algorithm and on the other hand we use DF-QuAD, described in section 2, as part of our evaluation in section 4.5.

4.4. Step (4): Final classification decision

In the final step we need to make a classification choice based on the dialectical strengths of the labels that we have calculated in step (3). For example, if we use the method outlined in section 3.4, we simply choose whichever of S_{Sup}, S_{Att}, S_N has the highest dialectical strength.

4.5. Corpora & Evaluation

4.5.1. Corpora. Below we describe the following corpora, each of which we have used to build classification models for our evaluation of CleAr for RbAM:

(1) Internet Argument Corpus (IAC) [Walker et al. 2012]

(2) AIFdb corpus [Lawrence et al. 2012; Lawrence and Reed 2014]
(3) News articles corpus [Carstens and Toni 2015b]

While corpora (1) and (2) are based on third party corpora, we have purpose built corpus (3) as part of our developments. Both corpora (2) and (3) consist of pairs of sentences or sentence like structures; the IAC constituents that make up each pair may be comprised of multiple sentences. While corpora (1) and (2) are comprised of instances from the *Att* and *Sup* classes, only, corpus (3) also contains instances assigned to the *N* class.

The Internet Argument Corpus (IAC) [Walker et al. 2012] is comprised of text extracted from the debating website *4forums*[2]. As part of their posts users of 4forums have the option of quoting other users and replying to the quoted text. From posts that contain quotes Walker and colleagues [Walker et al. 2012] have created annotated pairs of quotes and responses to the quotes, *Q-R pairs*, as well as chains of related posts from the same source. As these text passages are clearly marked in the forum, they are easily identifiable. The structure of the Q-R pairs aligns well with that needed in RbAM, where we treat the quote of the pair to be the Parent and the response to be the Child.

Based on the Argument Interchange Format [Bex et al. 2013; Chesnevar et al. 2006; Rahwan and Reed 2009] Lawrence and colleagues [2012; 2014] have built the *Argument Interchange Format corpus*, or *AIFdb*[3], an online repository of corpora annotated according to AIF specifications. It is comprised of 54 sub-corpora, concerned with a variety of topics, ranging from discussions that have taken place on the *BBC Moral Maze*[4] to argumentative microtexts. Each corpus is subdivided into a collection of tree structured *argument maps*. Of the 54 corpora we have chosen a subset of 18 because the remaining 36 corpora are either small (below 20 argument maps) or the content was in a language other than English. AIF defines five node types [Bex et al. 2013; Chesnevar et al. 2006], from which we create a five-class corpus, using the original types as labels, and a two-class corpus where we use only pairs that are connected through node types equivalent to support (RA-nodes) or attack relations (CA-nodes).

To fully evaluate RbAM it was necessary to have a corpus at hand that was comprised of instances not only assigned to the Att and Sup classes, but to the N class, as well. To this end we chose to create a new corpus from scratch, rather than artificially creating instances for the third class and adding them to an existing corpus. We chose the domain of news articles, focusing on a collection of publications concerned with the *UK Independence Party (UKIP)*. The rise of this eurosceptic party in the UK was a common and contentious topic at the time of our annotation work and thus seemed a natural choice. The corpus resulting from these annotations is comprised of $2,274$ instances where the split between classes is $Att = 413$, $Sup = 456$ and $N = 1,385$. We selected a random subset of 200 instances to be labeled by three annotators to test for inter-annotator agreement. We calculated agreement using *Fleiss' kappa*, which provides an agreement measure that takes the probability of chance agreement into consideration. Agreement on the 200 instances lay at $\kappa = .4287$ (Standard Error $= 0.0705$, 95% Confidence Interval = 0.2905 to 0.5670). From the annotations we created two corpora, a two-class (Att, Sup) corpus and a three-class (Att, Sup, N) corpus, and balanced the class distribution in each, retaining 413 instances per class, a total 826 for the two-class corpus and $1,239$ for the three-class corpus.

4.5.2. Evaluation. To evaluate the performance of applying CleAr to RbAM we conducted a two-tiered evaluation. In a first step we trained three different models on each corpus described above, exploiting RF, SVM and NB classifiers, respectively. After initial experiments

[2]www.4forums.com
[3]www.arg.dundee.ac.uk/aif-corpora/
[4]www.bbc.co.uk/programmes/b006qk11

Table II: Comparing classification performance for the *Two-class News articles corpus*, representing instances using features and BOW, respectively

Representation	Classifier	Accuracy	Precision	Recall	$\mathbf{F}_1$
BOW	RF	0.695	0.695	0.695	0.695
	SVM	0.691	0.692	0.691	0.691
	NB	0.663	0.663	0.663	0.663
Features	RF	0.684	0.684	0.684	0.684
	SVM	0.57	0.57	0.57	0.57
	NB	0.561	0.562	0.562	0.562

to gauge of performance we opted to train the Support Vector Machines using *polynomial* kernels (with $C = 1$ and $\epsilon = 2^{-12}$), rather than, for example, *radial basis function* or *linear* kernels. For Random Forests Oshiro and colleagues [2012] point out that little gain in classification performance seems to be made when using any more than 128 trees in a forest. We have found, however, that increasing the tree count from 128 to 256 helps performance for some corpus combinations and we have hence opted to report results from building Random Forests with 256 trees (with unlimited maximum tree depth and the Weka[5] default number of features $K = log_2(num\ features) + 1$). We chose parameter values based on a simple grid search for the baseline and used the same parameter values when applying CleAr. To select a representation of instances we conducted some preliminary experiments across different corpus and classifier combinations, omitted here for space reasons. These showed that representing instances with a set of features, formalising the same knowledge as the arguments in table I do, produced slightly inferior results compared to using a BOW representation, only. Table II compares the results of training classifiers for the two-class News corpus where instances are represented by features and BOW, respectively, illustrating how the simpler BOW representation compares favourably. Accordingly we apply CleAr using classifiers trained on the BOW representation, as described in section 2.2, and report results below. For each of the corpora described above we then chose the best performing classification model out of these three, according to F_1 measures, as the baselines, and attempted to improve upon it by applying CleAr. We compare the performance of using classifiers, only, with that of using CleAr, applying either ESAA or DF-QuAD for argument strength calculation. We conduct a total of 12 experiments, 4 (corpora) $\times 1$ (model per corpus) $\times 3$ (baseline vs. two Argumentation algorithms with CleAr). We describe the experiments, split by corpora, in turn below. A summary of all evaluations can be found in figure 7, where we show the F_1 measures across experiments.

For the first run of experiments we take the best performing model trained on the IAC, RF, and use this model as part of CleAr. The aim is then to identify whether, by using ESAA or DF-QuAD, we improve upon the baseline provided by the RF. Table IIIa shows the baseline classification performance in comparison to the two applications of CleAr. We find that applying CleAr with ESAA yields no improvements on the baseline. Using DF-QuAD, on the other hand, gives us an improvement on the baseline F_1 measure of 0.017 points.

For the application of CleAr to the AIFdb corpora we limit ourselves to the two-class version. This is due to the fact that our argument base is concerned with determining attack and support relations, whereas in the 5-class AIFdb corpus instances are labeled according to the RST node types connecting the I-nodes. Table IIIb shows the results of applying CleAr in comparison to the baseline. Again, applying CleAr with ESAA produces no performance improvement, while DF-QuAD yields an improvement in F_1 scores by 0.007 points.

[5]We have used the Weka toolbox [Hall et al. 2009] throughout for the experiments reported here. In other experiments we have also used other libraries, such as libSVM [Chang and Lin 2011]

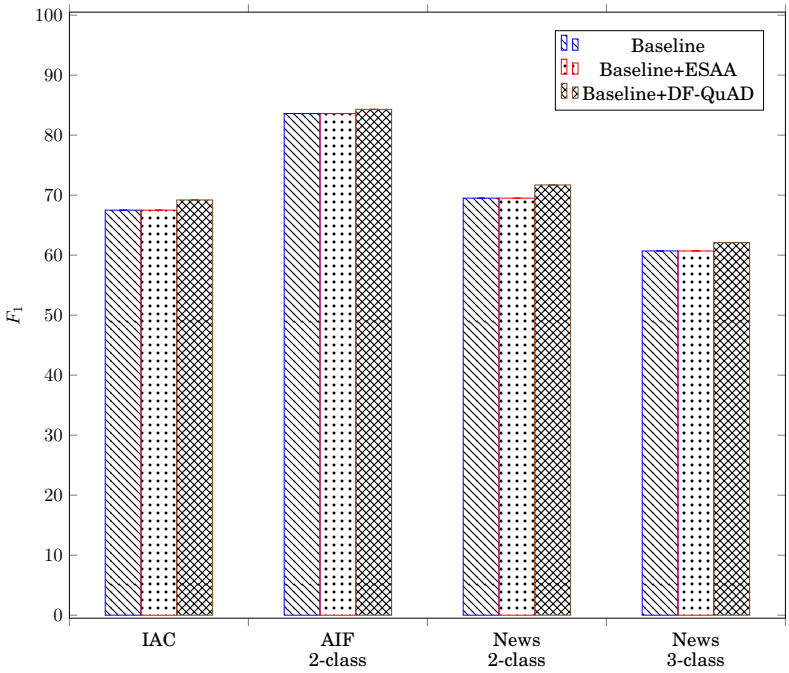


Fig. 7: Summary of evaluation results of applying CleAR to RbAM, where we compare the F_1 score of the baseline with the application of CleAr using ESAA or DF-QuAD for each corpus

For the final run of experiments we compare the baseline obtained from training RF on the two- and three-class News articles corpora. Table IIIc shows the results for the two-class corpus, while table IIId shows the results for the three-class corpus. Again, in both cases, applying CleAr with ESAA has no impact on classification, while applying it using DF-QuAD yields an improvement in both cases. For the two-class corpus the F_1 score is improved by 0.022 points, for the three-class corpus we achieve an improvement of 0.014 points.

With the application of CleAr with DF-QuAD we improve on all four baselines. We do not achieve statistically significant differences ($p = 0.686$[6]). Note, however, that the statistical power of the significance tests is low, due to the small number of experiments we compare. Yeh [2000] also points out that standard tests for statistical significance often underestimate the significance of differences in classification for natural language problems. We believe that more tests are needed to confirm or disprove the results provided here, but we see it as an encouraging sign that, using DF-QuAD semantics, we obtain improvements on all corpora.

5. CLEAR FOR CROSS-DOMAIN SENTIMENT POLARITY CLASSIFICATION

Sentiment Analysis is generally concerned with (1) identifying opinionated text, e.g. [Wiebe et al. 2004; Zhang and Ye 2008], and (2) identifying the polarity of text that is deemed to be opinionated, e.g. [Pang et al. 2002; Taboada et al. 2011; Turney 2002]. These tasks may be defined as a three-class problem, classifying instances in a single pass as *positive*, *negative* or *neutral*, as in e.g. [Kouloumpis et al. 2011]. Another popular approach has been to split the problem into two binary classification problems, where we first extract opinionated text and then classify its polarity, as in e.g. [Jiang et al. 2011]. We focus on task (2), which is

[6] p is calculated using a *paired-sample t-test*, see www.mathworks.com/help/stats/ttest.html for details

Table III: Classification results for RbAM with and without the application of CleAr

(a) Results for the *IAC*; RF baseline vs. CleAr

Classifier	Accuracy	Precision	Recall	$\mathbf{F}_1$
RF baseline	.675	.675	.675	.675
RF baseline & ESAA	.675	.675	.675	.675
RF baseline & DF-QuAD	.692	.692	.692	**.692**

(b) Results for the *two-class AIFdb corpus*; RF baseline vs. CleAr

Classifier	Accuracy	Precision	Recall	$\mathbf{F}_1$
RF baseline	.836	.836	.836	.836
RF baseline & ESAA	.836	.836	.836	.836
RF baseline & DF-QuAD	.843	.843	.843	**.843**

(c) Results for the *two-class News articles corpus*; RF baseline vs. CleAr

Classifier	Accuracy	Precision	Recall	$\mathbf{F}_1$
RF baseline	.695	.695	.695	.695
RF baseline & ESAA	.695	.695	.695	.695
RF baseline & DF-QuAD	.717	.717	.717	**.717**

(d) Results for the *three-class News articles corpus*; SVM baseline vs. CleAr

Classifier	Accuracy	Precision	Recall	$\mathbf{F}_1$
SVM baseline	.605	.609	.605	.607
SVM baseline & ESAA	.605	.609	.605	.607
SVM baseline & DF-QuAD	.618	.624	.618	**.621**

referred to as *polarity classification*. More specifically we address *binary cross-domain polarity classification*, where we train models on corpora that assign sentences (or Tweets) to one of two classes: positive or negative. We then use this model to classify unseen instances from a different domain; we may, for example, train a model on a corpus of Tweets and then use this model to classify sentences from movie reviews. Below we formally define an instantiation of CleAr for sentiment polarity classification.

5.1. Step (1): Instance classification

We first classify instance I , using a model trained on one of the corpora described in section 5.5. Each instance I is either a sentence or a Tweet, such as the following sentence m:

He was not capable of resolving disputes within his party.

For binary polarity classification the set of class labels is $A = \{Neg, Pos\}$ where Pos stands for $Positive$ and Neg stands for $Negative$.

5.2. Step (2): Argument identification

Once we have classified instance I we identify arguments and relations that are applicable to the instance in question, as described in section 3.2. The argument base AB_{SA} we have built for polarity classification is comprised of two types of arguments:

(1) Keyword ($6,815$ arguments)
(2) Negation ($6,815$ argument)

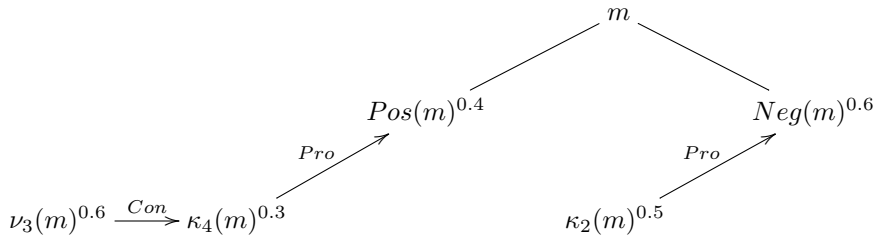


Fig. 8: Tree view of output of step (2) for Polarity classification for example sentence m

To construct keyword arguments we have used existing resources, namely word lists of positive and negative words. Each word in such a word list is used to create an argument, where the word itself is considered the premise, and its polarity is treated as the conclusion. All arguments have a syntax analogous to the arguments shown for RbAM in table I. To create keyword arguments we have opted to use a list compiled by Hu and Liu [2004], comprised of a total of $6,815$ keywords, $2,014$ whose class is *Positive* and $4,801$ whose class is *Negative*. In addition to keyword arguments we use a class of arguments to handle negation, defined analogously to the one described in section 4.2. For our example sentence m the following three arguments from AB_{SA} would be applicable:

— $\kappa_2(m)$: *dispute* occurs in m
— $\kappa_4(m)$: *capable* occurs in m
— $\nu_1(m)$: *capable* occurs negated in m

Arguments $\kappa_2(m)$ and $\kappa_4(m)$ argue for the labels Neg and Pos respectively, due to certain keywords occurring in m, and $\nu_1(m)$ attacks $\kappa_4(m)$, reflecting that the keyword is negated. The output of step (2) for this example, $I = m$, is shown in figure 8. To exemplify we assume that in step (1) the instance was labeled with $A = Neg$ and that $P_{Neg(m)}$ was 0.6. Note that, while the argument dealing with negation is assigned a score of 0.6, all arguments supporting positivity are assigned the same score, 0.5, while all those supporting negativity are assigned a score of 0.3. We assign different scores to the two groups of keyword arguments because we have a grand total of $2,019$ arguments supporting positivity in our argument base, compared to $4,810$ arguments supporting negativity. As for RbAM, assigning a base score to arguments systematically is an important question for future work.

5.3. Step (3): Answer strength calculation

In the third step of CleAr we determine the strength of each possible class label, including the applicable arguments identified in step (2), as described in section 3.3. The output of step (3) is shown in figure 9, where we have applied the ESAA method to calculate dialectical strength.

5.4. Step (4): Final classification decision

During the final step of CleAr we need to make a classification choice based on the final scores of the labels that we have calculated in step (3). For our example, we can determine the maximum of the scores $S_{Pos(m)}, S_{Neg(m)}$. If $S_{Pos(m)}$ is larger than $S_{Neg(m)}$ we label the sentence in question as *Positive*, if $S_{Neg(m)}$ is larger we label the sentence as *Negative*. Again, whenever $S_{Pos(m)} = S_{Neg(m)}$ we revert back to the original classification proposed in step (1).

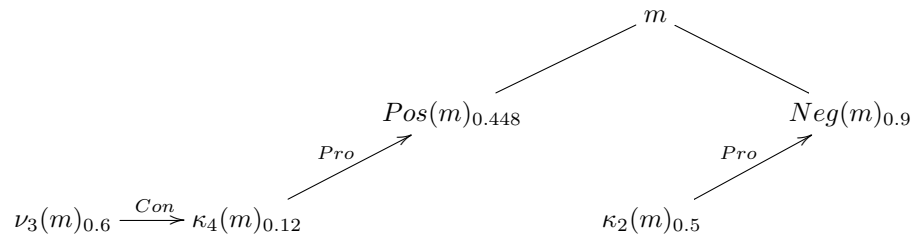


Fig. 9: Tree view of output of step (3) for Polarity classification for example sentence m

5.5. Corpora & Evaluation

Below we briefly describe the corpora based on which we have built CleAr for cross-domain polarity classification, as well as a summary of our evaluations. A more in-depth description of both can be found in [Carstens and Toni 2015a]. As with RbAM we have selected three corpora on which to run our experiments. Here, however, the way we evaluate performance differs somewhat. Because we are interested in *cross-domain* classification we were provided with a natural split between training and test data. For each experiment we have simply trained our models on one corpus and classified *all* instances from another corpus. This means that we conduct *six* sets of experiments, where in each we have used one of the corpora described below to train a classification model and another to test the classifier on.

5.5.1. Corpora. We have used two corpora comprised of Tweets, the *Sanders corpus* [Sanders 2011] and the *STS corpus* [Saif et al. 2013], while the third corpus [Pang and Lee 2005] is made up of positive and negative sentences, taken from movie reviews. The Sanders corpus, developed by Sanders Analytics[7], is comprised of $5,513$ manually annotated Tweets. Each Tweet belongs to one of four categories, *Apple, Google, Microsoft* or *Twitter*. The STS corpus has been developed with a focus on annotating entities, such as *Obama* or *Taylor Swift*, in Tweets alongside their sentiment. To ensure that the corpus only contains Tweets on topics that are different from those that make up the Sanders corpus we removed all Tweets from the *Technology* category, giving us a total of $1,799$ Tweets. Of those Tweets $1,211$ are negative and the remaining 588 are positive. The movie reviews corpus is comprised of $10,662$ sentences, with $5,331$ sentences labeled as positive and the other $5,331$ labeled as negative.

5.5.2. Evaluation. The results of training classification models on each of these corpora and testing these models on the remaining two corpora are summarised in figure 10, showing the F_1 measures of all experiments. For each corpus combination we show three baselines, one each for classifiers trained using RF, SVM and NB classifiers. For RF we use again 256 trees, as well as the Weka default number of features. For SVM, we employ polynomial kernels (with $C = 1$ and $\epsilon = 1^{-12}$). Every baseline is shown in comparison to the results of applying CleAr, as described above, to that baseline. For space reasons here we report only the application of CleAr with the ESAA Framework, which outperforms the application of CleAr with DF-QuAD for every combination of corpora and classifiers. Note that in all cases the application of CleAr produces significant ($p < 0.01$[8]) improvements on the baseline. A more detailed breakdown is shown in table IV. This means that, through the application of CleAr, we have been able to improve commonly poor cross-domain classification to more promising levels, showing what may prove to be an alternative to the often costly construction of annotated resources.

[7] www.sananalytics.com/lab/twitter-sentiment/
[8] p was calculated using a *one-way analysis of variance (ANOVA)*

Table IV: Detailed results for experiments described in section 5.5.2, using a paired t-test for to compare each classifier's baselines to the results of using of CleAr

	p	Mean		SD	
		Baseline	CleAr	Baseline	CleAr
RF	$< .0001$	.590	.646	.038	.033
SVM	.0001	.590	.651	.043	.035
NB	.0006	.610	.657	.042	.031

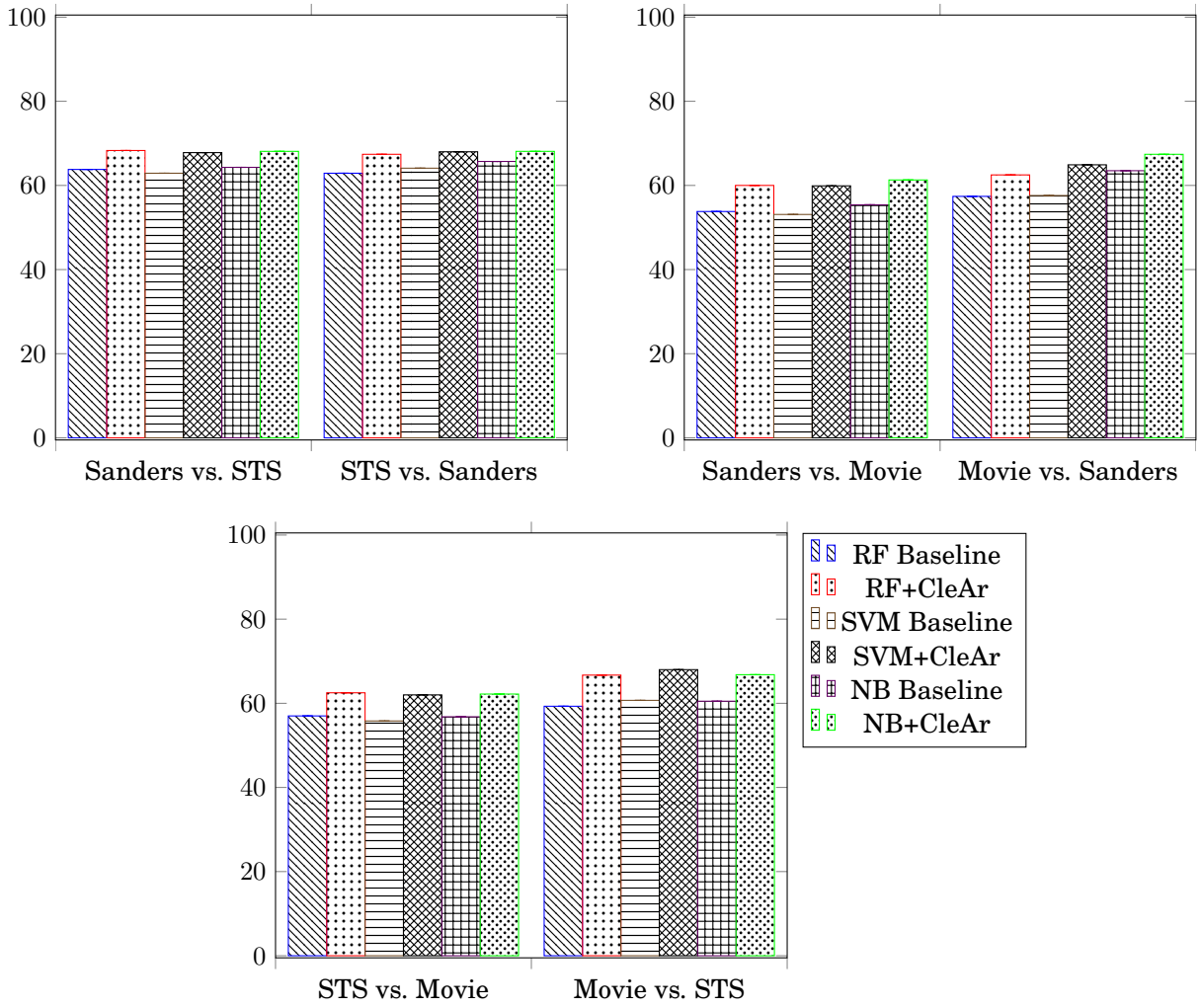


Fig. 10: Summary of evaluation results of applying CleAr to cross-domain Sentiment Polarity classification, where we compare the F_1 scores across all three baseline classifiers for each corpus combination with the results of applying ESAA, where the x-axis labels corpus combinations as *training corpus vs. test corpus*

6. RELATED WORK

Beyond the work we describe in this paper we find research based on synergies between Machine Learning and Argumentation in a number of areas, as e.g. summarised in [Cocarascu and Toni 2016]. Gao and colleagues [Gao and Toni 2014] integrate Argumentation with *Reinforcement Learning (RL)* through *Argumentation Accelerated Reinforcement Learning (AARL)*, while Mozina and colleagues focus on improving supervised learning with *Argument-based Machine Learning (ABML)* [Modgil et al. 2013; Mozina et al. 2005; Mozina et al. 2007]. Our work is similar in spirit to both AARL and ABML. Like the latter we integrate argumentation with supervised learning, but additionally make use of the dialectical power of argumentation by allowing argumentation frameworks corresponding to trees of potentially unlimited depth and using algorithms for determining strength. The former does so, as well, but instead uses reinforcement learning. In contrast to both we also use a numerical representation of argument strength.

In *argumentation mining* computational linguistics is used to extract arguments from text. Much work here has focused on the identification of individual arguments, or parts of arguments, in text, e.g. [Boltuzic and Šnajder 2015; Lippi and Torroni 2015b], as opposed to considering relations the way we have. *AraucariaDB* [Reed and Rowe 2004], which constitutes a significant chunk of the AIFdb corpus described in section 4.5, has been a popular target for argumentative analysis. Rooney and colleagues [2012] train SVM on the AraucariaDB corpus, based on words, word lemmata and part-of-speech tags. Another development on AraucariaDB is presented by Moens and colleagues [2007], who focus on identifying argumentative sentences using *multinomial Naive Bayes* classifiers. Some research has also addressed the challenges of trying to identify relations between arguments [Kirschner et al. 2015; Park and Cardie 2014; Petasis and Karkaletsis 2016]. Boltuzic and Šnajder [2014] use textual entailment [Dagan et al. 2006] to identify support relations between posts in discussion fora. Furthermore, recent developments of graphical representations of argumentative text, such as in [Barker and Gaizauskas 2016; Toledo-Ronen et al. 2016], illustrate possible paths towards identifying argumentative relations. Amgoud and colleagues [2015] extrapolate a formal representation of arguments and their internal structure as part of a framework to represent arguments that have been mined from natural language text. This constitutes a first attempt at intertwining argumentation mining with formal representations of arguments. Amgoud and Serrurier [2008a; 2008b] use training examples, as conventionally used to build classification models, to construct arguments for assigning instances to classes, instead. Despite these, and other, developments, to our knowledge, no implementation has been proposed to date that fully automates the process of building argument frameworks from text. For more in-depth surveys of the field refer, for example, to [Palau and Moens 2009; Peldszus and Stede 2013; Lippi and Torroni 2015a].

There is a broad interest in developing solutions for cross-domain classification, particularly Sentiment Analysis, illustrated by e.g. [Bollegala et al. 2011; 2013; Dragoni et al. 2015; Tsai et al. 2013; Weichselbraun et al. 2014; Xia et al. 2013]. For example, Bollegala and colleagues [2011; 2013] construct a *sentiment sensitive* lexicon whose contents are used to expand feature sets that represent text. Tsai and colleagues [2013] build a concept-level sentiment lexicon based entirely on common sense knowledge. We may use such thesauri to try and create argument bases and thus incorporate the knowledge in our classification methodology. Xia and colleagues [2013] propose the *feature ensemble plus sample selection (SS-FE)* method; they also provide an overview of domain adaptation work in sentiment analysis.

Finally there appears to be a critical link between the two phenomena we have studied in our work, sentiment and argumentation. It seems reasonable to assume that human argumentation is not conducted purely based on rational insights, but rather that sentiment plays an important role in how arguments are put forth. This has been corroborated in an empir-

ical study by Benlamine and colleagues [2015] and, as illustrated in table I, has motivated us to formalise the presence of sentiment, positive or negative, as arguments that help us to determine the presence of arguments.

7. CONCLUSION

We have described our work at the intersection of argumentation and supervised learning, and how we may bring the two fields together to help us in solving computational linguistics challenges. Specifically we have described *Classification enhanced with Argumentation (CleAr)*, a novel classification methodology that integrates knowledge in the form of arguments with supervised learning for computational linguistics. We have developed two instantiations of CleAr to address specific computational linguistics tasks:

(1) *Relation-based Argumentation Mining (RbAM)*
(2) *Cross-domain Sentiment Polarity Classification*

RbAM is an approach to argumentation mining that capitalises on argumentative relations between sentences. While for sentiment polarity classification we have used existing resources, we have developed resources for performing RbAM from scratch. This has included the construction of a tailor-made corpus of annotated sentence pairs. We believe that our developments illustrate a novel and promising application area for argumentation, namely the improvement of corpus-based classifiers for computational linguistics problems. We have shown that by applying the CleAr methodology we can improve performance both in (cross-domain) sentiment polarity classification and in (relation-based) argumentation mining. Additionally we have highlighted the importance of considering text in relation to what surrounds it to identify arguments and argumentative structures in text. By applying argumentation to computational linguistics problems and by using computational linguistics to identify argumentative text we have highlighted the reciprocal link between the two fields, illustrating how we stand to benefit from considering insights from both fields in unison.

Future work will need to address a variety of challenges. In sentiment analysis the most pressing shortcoming of our work is arguably that we have limited ourselves to polarity classification, and we aim to apply our methodology to sentiment detection, too. To improve RbAM, amongst other tasks, we will need to identify more sophisticated ways of constructing sentence pairs from free text. For both tasks, as well as any other we may apply CleAr to, we will need to identify effective ways of extending and constructing new argument bases, as well as making more informed choices of base scores for said arguments. To further evaluate the viability and scope of our work we will need to test it on more corpora, either existing ones or purpose-built ones tailored to our problems. Existing corpora that offer valuable data include, but are not limited to, those described in [Cabrio and Villata 2014; Rosenthal et al. 2015].

REFERENCES

Teresa Alsinet, Carlos Iván Chesñevar, Lluis Godo, and Guillermo Ricardo Simari. 2008. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems* 159 (2008), 1208–1228.

Leila Amgoud, Philippe Besnard, and Anthony Hunter. 2015. Representing and Reasoning about Arguments Mined from Texts and Dialogues. In *Proceedings of European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (LNCS) volume 9161*. Springer, 60–71.

Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and Pierre Livet. 2008. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems* 23, 10 (2008), 1062–1093.

Leila Amgoud and Mathieu Serrurier. 2008a. Agents that argue and explain classifications. *Autonomous Agents and Multi-Agent Systems* 16, 2 (2008), 187–209.

Leila Amgoud and Mathieu Serrurier. 2008b. Arguing and explaining classifications. In *Argumentation in Multi-Agent Systems*. Springer, 164–177.

Emma Barker and Robert Gaizauskas. 2016. Summarizing multi-party argumentative conversations in reader comment on news. *Proceedings of the Third Workshop on Argumentation Mining* (2016), 12.

Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. 2015. Automatic evaluation of design alternatives with quantitative argumentation. *Argument & Computation* 6, 1 (2015), 24–49.

Sahbi Benlamine, Maher Chaouachi, Serena Villata, Elena Cabrio, Claude Frasson, and Fabien Gandon. 2015. Emotions in argumentation: an empirical evaluation. In *International Joint Conference on Artificial Intelligence, IJCAI 2015*. 156–163.

Floris Bex, Sanjay Modgil, Henry Prakken, and Chris Reed. 2013. On logical specifications of the Argument Interchange Format. *Journal of Logic and Computing* 23 (2013), 951–989.

Danushka Bollegala, David Weir, and John Carroll. 2011. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies (ACL-HLT)-Volume 1*. ACL, 132–141.

Danushka Bollegala, David Weir, and John Carroll. 2013. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Transactions on Knowledge and Data Engineering* 25, 8 (2013), 1719–1731.

Filip Boltuzic and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*. 49–58.

Filip Boltuzic and Jan Šnajder. 2015. Identifying Prominent Arguments in Online Debates Using Semantic Textual Similarity. *Proceedings of the Second Workshop on Argumentation Mining* (2015), 110–115.

Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

Elena Cabrio and Serena Villata. 2014. NoDE: A Benchmark of Natural Language Arguments. *COMMA* 266 (2014), 449–450.

Erik Cambria and Bruce White. 2014. Jumping NLP curves: a review of natural language processing research [review article]. *IEEE Computational Intelligence Magazine* 9, 2 (2014), 48–57.

Lucas Carstens, Xiuyi Fan, Yang Gao, and Francesca Toni. 2015. An Overview of Argumentation Frameworks for Decision Support. In *Graph Structures for Knowledge Representation and Reasoning - 4th International Workshop, GKR*. 32–49.

Lucas Carstens and Francesca Toni. 2015a. Improving Out-of-domain Sentiment Polarity Classification using Argumentation. In *SENTIRE*. IEEE, 1294–1301.

Lucas Carstens and Francesca Toni. 2015b. Towards relation based Argumentation Mining. In *Proceedings of the Second Workshop on Argumentation Mining*. 29–34.

Lucas Carstens, Francesca Toni, and Valentinos Evripidou. 2014. Argument Mining and Social Debates. *COMMA* (2014).

Claudette. Cayrol and Marie-Christine Lagasquie-Schiex. 2005. On the Acceptability of Arguments in Bipolar Argumentation Frameworks. In *ECSQARU*. 378–389.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 3 (2011), 27.

Carlos Ivan Chesnevar, Jarred McGinnis, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Ricardo Simari, Matthew South, Gerard Vreeswijk, and Steven Willmott. 2006. Towards an argument interchange format. *Knowledge Engineering Review* 21, 4 (2006), 293–316.

Oana Cocarascu and Francesca Toni. 2016. Argumentation for Machine Learning: A Survey. *COMMA* (2016), 219.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12 (2011), 2493–2537.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment* (2006), 177–190.

Mauro Dragoni, Andrea GB Tettamanzi, and Célia da Costa Pereira. 2015. Propagating and aggregating fuzzy polarities for concept-level sentiment analysis. *Cognitive Computation* 7, 2 (2015), 186–197.

Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and¡ i¿ n¡/i¿-person games. *Artificial intelligence* 77, 2 (1995), 321–357.

Valentinos Evripidou and Francesca Toni. 2014. Quaestio-it. com: a social intelligent debating platform. *Journal of Decision Systems* 23, 3 (2014), 333–349.

Yang Gao and Francesca Toni. 2014. Argumentation Accelerated Reinforcement Learning for Cooperative Multi-Agent Systems. In *Proceedings of European Conference on Artificial Intelligence (ECAI)*. IOS Press, 333.

Mehmet Gönen, Ayse Gönül Tanugur, and Ethem Alpaydin. 2008. Multiclass posterior probability support vector machines. *Neural Networks, IEEE Transactions on* 19, 1 (2008), 130–139.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18.

Alexander Hogenboom, Flavius Frasincar, Franciska de Jong, and Uzay Kaymak. 2015. Using rhetorical structure in sentiment analysis. *Commun. ACM* 58, 7 (2015), 69–77.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international Conference on Knowledge discovery and data mining*. ACM, 168–177.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies (ACL-HLT)-Volume 1*. ACL, 151–160.

George H John and Pat Langley. 1995. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in artificial intelligence (UAI)*. Morgan Kaufmann Publishers Inc., 338–345.

Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2015. Linking the Thoughts: Analysis of Argumentation Structures in Scientific Publications. *Proceedings of the Second Workshop on Argumentation Mining* (2015), 1–11.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg!. In *Proceedings of the AAAI Conference on Web and Social Media (ICWSM)*, Vol. 11. AAAI, 538–541.

Robert A Kowalski and Francesca Toni. 1996. Abstract argumentation. *Artificial intelligence and law* 4, 3-4 (1996), 275–296.

John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. 2012. AIFdb: Infrastructure for the Argument Web.. In *Proceedings of Computational Models of Argument (COMMA)*. IOS Press, 515–516.

John Lawrence and Chris Reed. 2014. AIFdb Corpora. *Proceedings of Computational Models of Argument (COMMA)* 266 (2014), 465.

Marco Lippi and Paolo Torroni. 2015a. Argument Mining: A Machine Learning Perspective. In *Theory and Applications of Formal Argumentation*. Springer, 163–176.

Marco Lippi and Paolo Torroni. 2015b. Context-independent claim detection for argument mining. In *Proceedings of the 24th International Conference on Artificial Intelligence (ICAI)*. AAAI Press, 185–191.

Paul-Amaury Matt and Francesca Toni. 2008. A Game-Theoretic Measure of Argument Strength for Abstract Argumentation. In *Logics in Artificial Intelligence, 11th European Conference, JELIA 2008, Dresden, Germany, September 28 - October 1, 2008. Proceedings*. 285–297. DOI:http://dx.doi.org/10.1007/978-3-540-87803-2_24

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. Citeseer, 3111–3119.

Sanjay Modgil, Francesca Toni, Floris Bex, Ivan Bratko, Carlos I Chesñevar, Wolfgang Dvořák, Marcelo A Falappa, Xiuyi Fan, Sarah Alice Gaggl, Alejandro J García, and others. 2013. The added value of argumentation. In *Agreement Technologies*. Springer, 357–403.

Marie-Francine Moens, Erik Boiy, Raquel Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th international Conference on Artificial intelligence and law (ICAIL)*. ACM, 225–230.

Martin Mozina, Jure Zabkar, Trevor Bench-Capon, and Ivan Bratko. 2005. Argument based machine learning applied to law. *Artificial Intelligence and Law* 13, 1 (2005), 53–73.

Martin Mozina, Jure Zabkar, and Ivan Bratko. 2007. Argument based machine learning. *Artificial Intelligence* 171, 10-15 (2007), 922–937.

Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. 2011. Natural language processing: an introduction. *Journal of the American Medical Informatics Association* 18, 5 (2011), 544–551.

Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. 2012. How many trees in a random forest?. In *Proceedings of the International Conference on Machine Learning and Data Mining (MLDM)*. Springer, 154–168.

Raquel Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international Conference on Artificial Intelligence and Law (ICAIL)*. ACM, 98–107.

Bo Pang and Lilian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on ACL (ACL)*. ACL, 115–124.

Bo Pang, Lilian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)-Volume 10*. ACL, 79–86.

Joonsuk Park and Claire Cardie. 2014. Identifying Appropriate Support for Propositions in Online User Comments. *Proceedings of The 52nd Annual Meeting of the ACL* (2014), 29.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* 7, 1 (2013), 1–31.

Georgios Petasis and Vangelis Karkaletsis. 2016. Identifying Argument Components through TextRank. *Proceedings of the Third Workshop on Argumentation Mining* (2016), 94.

John C. Platt. 1999. Probabilities for SV Machines. In *Advances in Large Margin Classifiers*. MIT Press, 61–74. http://research.microsoft.com/apps/pubs/default.aspx?id=69187

Antonio Rago, Francesca Toni, Marco Aurisicchio, and Pietro Baroni. 2016. Discontinuity-Free Decision Support with Quantitative Argumentation Debates. In *Proceedings of the 15th International Conference on Principles and Knowledge Representation and Reasoning (KR)*. AAAI, 63–73.

Iyad Rahwan and Chris Reed. 2009. The argument interchange format. In *Argumentation in artificial intelligence*. Springer, 383–402.

Chris Reed and Glenn Rowe. 2004. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools* 13, 04 (2004), 961–979.

Niall Rooney, Hui Wang, and Fiona Browne. 2012. Applying Kernel Methods to Argumentation Mining. In *Proceedings of the 25th Florida Artificial Intelligence Research Society Conference (FLAIRS)*. AAAI.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of SemEval-2015* (2015).

Hassan Saif, Miriam Fernandez, Yulan He, and Harith Alani. 2013. Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold. In *1st Interantional Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013)*. AI*IA, 9–21.

Gerard Salton and Michael J McGill. 1986. *Introduction to modern information retrieval*. McGraw-Hill, Inc.

Niek J Sanders. 2011. *Sanders-Twitter Sentiment Corpus*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on empirical methods in natural language processing (EMNLP)*, Vol. 1631. ACL, 1642.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37, 2 (2011), 267–307.

Orith Toledo-Ronen, Roy Bar-Haim, and Noam Slonim. 2016. Expert Stance Graphs for Computational Argumentation. *Proceedings of the Third Workshop on Argumentation Mining* (2016), 119.

Stephen Toulmin. 2003. *The uses of argument*. Cambridge University Press.

Angela Charng-Rurng Tsai, Chi-En Wu, Richard Tzong-Han Tsai, and Jane Yung-jen Hsu. 2013. Building a concept-level sentiment dictionary based on commonsense knowledge. *IEEE Intelligent Systems* (2013), 22–30.

Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on ACL (ACL)*. ACL, 417–424.

Frans H. Van Eemeren and Rob Grootendorst. 2003. *A systematic theory of argumentation: The pragma-dialectical approach*. Vol. 14. Cambridge University Press.

Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A Corpus for Research on Deliberation and Debate.. In *Proceedings of the 8th international Conference on Language Resources and Evaluation (LREC)*. Citeseer, 812–817.

Albert Weichselbraun, Stefan Gindl, and Arno Scharl. 2014. Enriching semantic knowledge bases for opinion mining in big data applications. *Knowledge-Based Systems* 69 (2014), 78–85.

Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. 2004. Learning subjective language. *Computational linguistics* 30, 3 (2004), 277–308.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on ACL*. 133–138.

Rui Xia, Chengqing Zong, Xuelei Hu, and Erik Cambria. 2013. Feature ensemble plus sample selection: domain adaptation for sentiment classification. *IEEE Intelligent Systems* 28, 3 (2013), 10–18.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational linguistics-Volume 2*. ACL, 947–953.

Min Zhang and Xingyao Ye. 2008. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval. In *Proceedings of the 31st annual international ACM SIGIR Conference on Research and development in information retrieval*. ACM, 411–418.

Received January 2016; revised July 2016; accepted