

# Towards relation based Argumentation Mining

**Lucas Carstens**

Imperial College London  
United Kingdom  
SW2 7AZ, London  
lc1310@imperial.ac.uk

**Francesca Toni**

Imperial College London  
United Kingdom  
SW2 7AZ, London  
ft@imperial.ac.uk

## Abstract

We advocate a relation based approach to Argumentation Mining. Our focus lies on the extraction of argumentative relations instead of the identification of arguments, themselves. By classifying pairs of sentences according to the relation that holds between them we are able to identify sentences that may be factual when considered in isolation, but carry argumentative meaning when read in context. We describe scenarios in which this is useful, as well as a corpus of annotated sentence pairs we are developing to provide a testbed for this approach.

## 1 Introduction

Arguments form an integral part of human discourse. Whether we argue in dialogue with another person or advocate the merits of a product in a review, arguments are ubiquitous, in real life as much as in the world wide web. The ever increasing amounts of data on the web mean that manual analysis of this content, including debates and arguments, seems to become increasingly infeasible. Among other problems Argumentation Mining addresses this issue by developing solutions that automate, or at least facilitate, the process of building *Argument Frameworks (AFs)* (Amgoud et al., 2008; Dung, 1995) from free text. To build AFs we are generally concerned with two problems, (1) the identification of arguments and (2) the identification of relations between arguments. With this paper we highlight the intricate link between those two tasks and argue that treating them separately

raises a number of issues. On the back of this we propose a relation based way of performing Argumentation Mining. Instead of treating the identification of arguments and their relations to each other as two problems we define it as a single task. We do this by classifying sentences according to whether they stand in an *argumentative* relation to other sentences. We consider any sentence which supports or attacks another sentence to be argumentative. This includes cases such as the one shown in section 3.1, where a sentence contains only parts of an arguments (premises or a conclusion) and the remainder of the argument is left implicit for the reader to infer.

The remainder of this paper is organised as follows. We discuss related work in section 2. In section 3 we discuss three issues that arise when trying to decouple the process of identifying arguments and finding relations between them. Following this we describe a relation based approach to perform Argumentation Mining for the creation of AFs in section 4. We discuss an application in section 5.1, as well as a corpus design to help us build such applications in section 5.2. We conclude the paper in section 6.

## 2 Related work

Work on Argumentation Mining has addressed a number of tasks crucial to the problem, including the automatic construction of *Argument Frameworks (AFs)* (Cabrio and Villata, 2012; Feng and Hirst, 2011) and the creation of resources such as annotated corpora (Mochales and Moens, 2008; Stab and Gurevych, 2014; Walker et al., 2012). Amidst the increasing interest in Argumentation Mining various types of online content have been the target of anal-

ysis. (Park and Cardie, 2014) use *multi-class Support Vector Machines (SVM)* (Crammer and Singer, 2002) to identify different classes of argumentative propositions in online user comments. (Ghosh et al., 2014) use SVM to analyse multilogue, instead, classifying relations between user comments. (Boltuzic and Šnajder, 2014) use Textual Entailment to identify support relations between posts in discussion fora. Other application areas for Argumentation Mining have been the biomedical (Faiz and Mercer, 2014; Green, 2014; Hounbo and Mercer, 2014) and legal domains, where the well-structured nature of legal text and the development of corpora such as the *ECHR* corpus (Mochales and Moens, 2008) have sparked development in this area (Palau and Moens, 2009; Wyner et al., 2010).

### 3 Motivation

The separation of identifying arguments and the relations between them raises a number of problems, three of which are highlighted here to motivate our approach.

#### 3.1 This is just a fact - so why does it attack this other sentence?

The context in which a sentence appears can change its meaning significantly, and with it a sentence's *argumentativeness*. Consider the following statement:

(1) Nigel Farage<sup>1</sup> has attended private school and used to work as a banker in the City.

This is a simple enough fact and, on its own, conveys no particular attitude towards Nigel Farage, his education, or his professional past. If however, we consider the above sentence in relation to the one below, the situation changes:

(2) Nigel Farage understands the common folks; he is the face of UKIP, the people's army!

It now becomes quite possible that sentence (1) is meant to be an attack on sentence (2) and the notion of Nigel Farage being the leader of a *people's army*. After all, how could someone who went to

<sup>1</sup>Nigel Farage is the leader of the UK Independence Party (UKIP), see [www.ukip.org](http://www.ukip.org)

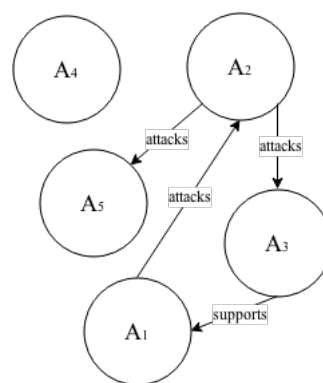


Figure 1: Example Argument Framework.

private school and has a history as a banker possibly understand the common people? This conclusion is not stated explicitly, but one may easily infer it. Trying to identify arguments in isolation may hence lead us to discard factual sentences such as sentence (1), even though, when considered in context with sentence (2), we should arguably consider it to be argumentative.

#### 3.2 I have found the arguments - relating them is still a three-class problem!

Let us consider again the task of identifying a sentence as argumentative or non-argumentative. Say we have built a model that provides us with a good split between the two classes, so that we can reliably discard non-argumentative sentences (though, as discussed in section 3.1, this concept may be questionable, as well). We now need to find relations, i.e. attacks and supports between the sentences that have been classified as argumentative. In spite of our knowledge of all sentences in question being arguments, we are still faced with a three-class problem, as three scenarios need to be accounted for. A sentence may attack another, it may supports another, and, lastly, both sentences may be arguments, but otherwise unrelated. By discarding non-argumentative sentences we thus simply limit the search space for the construction of an AF, the complexity of the problem itself remains unchanged.

#### 3.3 This is an argument - but is it relevant?

While in section 3.1 we argue that, by trying to identify sentences as argumentative or non-argumentative, we may discard potentially valuable

input to our AF, we may also end up retaining sentences that are of little use. Though often enough toy examples of AFs contain isolated arguments, as shown in figure 1, such arguments may arguably not be useful in real life applications. In the example AF, argument  $A_4$  does not offer us any insight either as to whether it is viable/acceptable or in what way it may contribute to identifying *good* arguments, by whichever measure this may be.

#### 4 Relation based Argumentation Mining

Based on the issues we describe in section 3 we have set out to offer an alternative, relation based view on Argumentation Mining. We hope that this will offer new ways of building AFs from text that may be useful on their own, but also complementary to other approaches. Instead of identifying sentences or other text snippets as (non)argumentative we classify *pairs* of sentences according to their relation. If this relation is classified as an *attack* or *support* relation we consider *both* sentences to be argumentative, irrespective of their individual quality. Accordingly we classify sentence pairs as belonging to one of three classes,  $A = Attack$ ,  $S = Support$ , or  $N = Neither$ , where *Neither* includes both cases where the two sentences are unrelated and those where they are related, but not in an argumentative manner. To construct pairs and build AFs from them we currently consider two options. On the one hand, we create a *root node*, a sentence to be compared to a set of other sentences. Consider, for example, a journalist who is in the process of composing an article on UKIP. To gather insights on the attitude towards UKIP he or she may want to test a claim against an existing body of articles. A claim here is a sentence conveying a hypothesis, such as:

$C =$  "UKIP's proposed immigration policies effectively discriminate against migrants from specific European countries, thereby undermining the inclusiveness and unity of the European Union."

To evaluate this claim we take a set of relevant sentences  $S = \{s_1, s_2, \dots, s_n\}$ , for example other news articles on UKIP. We then construct a set of sentence pairs  $P = \{(C, s_1), (C, s_2), \dots, (C, s_n)\}$ , where each  $p \in P$  needs to be assigned a class label

$L \in \{A, S, N\}$ . We can then determine which sentences from the articles attack or support the journalist's claim and can iteratively establish further connections between the sentences related to the original claim. On the other hand we may want to create an AF from a single piece of text. If the text is not very large and/or we have the computing power available we can simply create sentence pairs by matching every sentence with every other sentence appearing in the article. This, however, means we have to classify an exponentially growing number of pairs as we consider larger texts. It may hence be prudent to preselect pairs, e.g. by matching sentences containing the same entities. Once we have constructed pairs in some way we need to represent them in a manner that lets us classify them. To achieve this we represent each sentence pair as a single feature vector. The vector is comprised of a set of features, of which some characterise the sentences themselves and others describe the relation between the sentences. We describe preliminary work on building a corpus of such vectors, each annotated with a class label, in section 5.2.

#### 5 Putting theory into practice

Based on the ideas described in section 4 we have defined a number of use cases, one of which we discuss here, and have also developed a first annotated corpus of sentence pairs.

##### 5.1 Application

The first application we are developing following our approach offers a way of evaluating claims against a body of text, as described in section 4. As a first step, this provides us with a gauge of what proportion of a text argues for or against our claim. In a second step we can then discard sentences which do not appear to have an argumentative relation to our claim and try to establish further connections between those sentences that do, giving us a preliminary AF. At this stage the result will not be a fully fledged AF that reflects the argumentative structure of the text itself, simply because it relates to an external claim. To test our approach *in real life* we have teamed up with the BBC News Labs<sup>2</sup> to define a use case, for which figure 2 provides an overview. One

<sup>2</sup>[www.BBCNewsLabs.co.uk](http://www.BBCNewsLabs.co.uk)

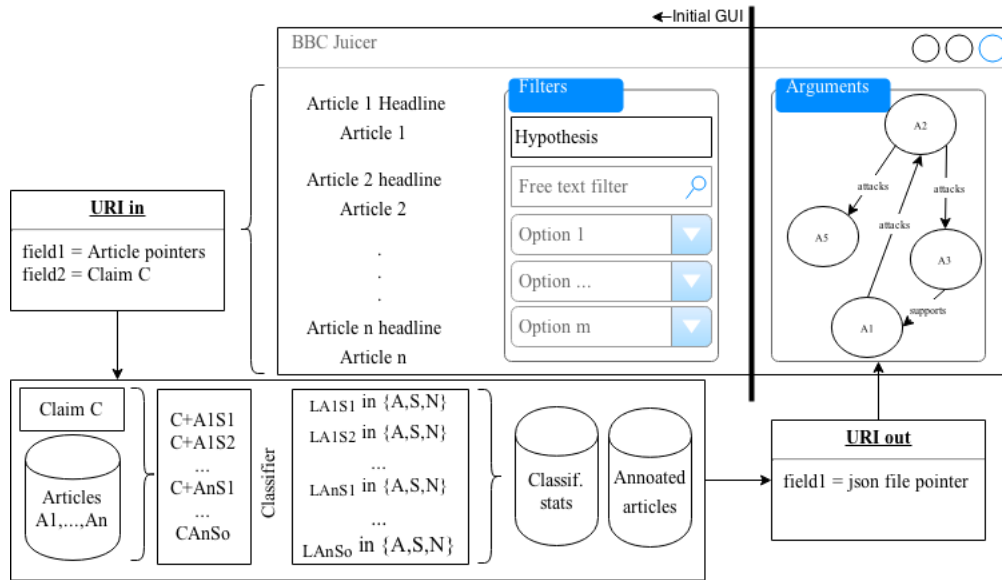


Figure 2: Mock Up of the Juicer, including the way the API interacts with it to retrieve & classify contents and then feed it back to the Juicer. The user enters a Claim and chooses a set of articles via the available filters. Pointers to the articles, as well as the Claim are then processed via an API. The classification output is fed back into the Juicer.

of the applications developed by the News Labs is *The Juicer*<sup>3</sup>, a platform used to facilitate the semantic tagging of BBC content. The Juicer provides an interface to a large repository of news articles and social media posts from various sources, such as the BBC websites and Twitter. Each article stored in the repository is assigned to various categories, such as topic and source, and is then semantically tagged for people, places, events, etc. We are currently developing an API to integrate a concrete realisation of relation based Argumentation Mining, to be used as an additional semantic filter in the Juicer. This will allow us to utilise the existing filters of the BBC Juicer to select the articles we want to compare with the claim. Pointers to the articles retrieved using the filters, as well as the provided claim are sent to be processed via the API. The content of the articles are then compared to the provided claim, as described in section 4. We are considering a number of options for how the resulting classifications may be presented to the user:

1. He or she may access simple statistics on the resulting classifications, e.g. the proportion of sentences attacking or supporting the claim.

<sup>3</sup>[www.bbc.co.uk/partnersandsuppliers/connectedstudio/newslabs/projects/juicer.html](http://www.bbc.co.uk/partnersandsuppliers/connectedstudio/newslabs/projects/juicer.html)

2. Alternatively the user may access the full articles, with sentences highlighted for argumentative contents.
3. Another option is to just view argumentative sentences directly, without the articles in which they appear. These sentence may be represented in graph form, as shown in figure 2.

## 5.2 Corpus development

To develop applications such as the one described in section 5.1 we need to build solid classification models. In turn, to build such models, we need a sizeable corpus of labeled examples, in our case sentence pairs that are labeled with  $L \in \{A, S, N\}$ . To identify the challenges in this we have built a preliminary corpus of 854 annotated sentence pairs<sup>4</sup>, examples of which are shown in table 1. Based on the insights gained from annotating a reasonable amount of sentence pairs we are now in the process of building a larger corpus in which each instance will be labeled by at least two annotators. The annotators are either native or fully proficient English speakers. We summarise the main points of the setup below.

Firstly, we do not ask annotators to identify arguments. This is based on the issues this raises, as

<sup>4</sup>Available at [www.doc.ic.ac.uk/~lc1310/](http://www.doc.ic.ac.uk/~lc1310/)

Parent	Child	Class
UKIP doesn't understand that young people are, by and large, progressive.	But UKIP claims to be winning support among younger voters and students.	a
It's a protest vote because (most) people know that UKIP will never net in power.	Emma Lewell-Buck made history becoming the constituency's first female MP.	n
It is because of UKIP that we are finally discussing the European question and about immigration and thank goodness for that.	I believe that what UKIP is doing is vital for this country.	s

Table 1: Example sentence pairs, labeled according to the relation pointing from the Child to the Parent

explained in section 3. Instead we ask annotators to focus on the relation, taking into account whatever may be implied in the sentences to then decide whether one attacks or supports the other. We will also ask annotators to provide qualitative feedback on whether they can pinpoint why they have classified pairs the way they have. This will be achieved via free text feedback or the completion of templates and will be used as a basis for further exploration on how we may represent and identify arguments.

This leads to the second challenge in building models that we can use in our applications: We need to decide how to represent the sentence pairs. Here, we have two options. We may either choose a *Bag-of-Words (BOW)* approach or develop a set of features that are representative of a sentence pair. The BOW approach is straight forward and has proven to yield reasonable results for many NLP problems, e.g. (Maas et al., 2011; Sayeedunnissa et al., 2013). We will hence use it as one of two baselines, the other being random classification. To see whether we can improve on both these baselines we have set out to collect a set of features that give us numerical representation of a sentence pair. Most broadly we distinguish two types of features, *Relational features* and *Sentential features*. Relational features will be comprised of any type of features that represent how the two sentences that make up the pair relate to each other. Features we have been experimenting with on our preliminary corpus include WordNet based similarity (Miller, 1995), Edit Distance measures (Navarro, 2001), and Textual Entailment measures (Dagan et al., 2006). The second category includes a set of features that characterise the individual sentences. Here we are considering various word lists, e.g. keeping count of discourse markers, sen-

timent scores, e.g. using SentiWordNet (Esuli and Sebastiani, 2006) or the Stanford Sentiment library (Socher et al., 2013), and other features. All features are then pooled together to create the feature vector representing a sentence pair. Experiments on the preliminary corpus, representing sentence pairs using all features described, show promising results on our approach, with classification accuracy of up to 77.5% when training Random Forests (Breiman, 2001) on the corpus.

## 6 Conclusion

We have advocated a relation based approach to performing Argumentation Mining. We focus on the determination of argumentative relations, foregoing the decision on whether an isolated piece of text is an argument. We do this arguing that often times the relation to other text is what lends text its argumentative quality. To illustrate the usefulness of this approach we have described a use case we are developing, as well as a corpus of annotated sentence pairs. Alongside the developments proposed in section 5 we need to conduct experiments to track the quality of data and classification output. For the construction of our corpus this means collecting multiple annotations, not just for a subset of the corpus, but for its entirety. This will allow us to monitor the quality of our annotations more reliably. Next to introducing features to represent sentence pairs we must determine the optimal feature combination at all stages of development. We need to avoid features that are detrimental to performance and those which do not contribute to it and waste computational resources.

## References

- Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasque-Schiex, and Pierre Livet. 2008. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23(10):1062–1093.
- Filip Boltuzic and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58.
- Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.
- Elena Cabrio and Serena Villata. 2012. Generating abstract arguments: A natural language approach. In *COMMA*, pages 454–461.
- Koby Crammer and Yoram Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292.
- Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising textual entailment*, pages 177–190. Springer.
- Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357.
- Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer.
- Syed Ibn Faiz and Robert E Mercer. 2014. Extracting higher order relations from biomedical text. *ACL 2014*, page 100.
- Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 987–996. Association for Computational Linguistics.
- Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 39–48.
- Nancy L Green. 2014. Towards creation of a corpus for argumentation mining the biomedical genetics research literature. *ACL 2014*, page 11.
- Hospice Hougbo and Robert E Mercer. 2014. An automated method to build a corpus of rhetorically-classified sentences in biomedical texts. *ACL 2014*, page 19.
- Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Raquel Mochales and Marie-Francine Moens. 2008. Study on the structure of argumentation in case law. *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems*, pages 11–20.
- Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88.
- Raquel Mochales 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*, pages 98–107. ACM.
- Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. *ACL 2014*, page 29.
- S Fouzia Sayeedunnissa, Adnan Rashid Hussain, and Mohd Abdul Hameed. 2013. Supervised opinion mining of social network data using a bag-of-words approach on the cloud. In *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, pages 299–309. Springer.
- 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)*, volume 1631, page 1642. Citeseer.
- Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1501–1510.
- Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *LREC*, pages 812–817.
- Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. *Approaches to text mining arguments from legal cases*. Springer.