

Combining Classification-centered and Relation-based Argument Mining Methods

Andrew Henning^a, Anthony P. Young^a, Elizabeth Sklar^{b,c}, Simon Miles^a, and Elizabeth Black^a

^aDepartment of Informatics, King's College London, United Kingdom

^bDepartment of Engineering, King's College London, United Kingdom

^cLincoln Institute for Agri-Food Technology, University of Lincoln, United Kingdom

{andrew.henning,peter.young,simon.miles,elizabeth.black}@kcl.ac.uk,
esklar@lincoln.ac.uk

Abstract. Two key tasks in argument mining (AM) are classification of argument components and identification of relations between argument components. Approaches to solving the argument component classification problem typically take a supervised learning approach, however a lack of suitable datasets makes this a challenge for identification of argument component relations. We propose a pipeline with a recurrent, branched structure that combines supervised learning of argument component classifications with NLP approaches to identification of argument component relations, with the aim of improving both classification of argument components (i.e. premises and claims) and identification of support relationships between components.

Keywords: Argument mining · Computational argumentation

1 Introduction and Background

Argument mining (AM) is a relatively new field intersecting computational argumentation, natural language processing (NLP), and machine learning. While the primary goal of AM is simple – to extract arguments from raw text and identify the relationships between them – researchers currently deploy sophisticated systems as pipelines with stages that tackle relevant sub-tasks, like boundary detection, component classification, and relation prediction [7]. Recently, AM has garnered increased interest, with applications in fields such as law and medicine, education, and social media (e.g. [1,2,6,9,10,11]). We aim to take raw textual data from Wikipedia articles, classify its argument components as *claims*, *premises*, *both*, or *neither*, and predict the support relationships between them. To do this we propose a novel AM pipeline that leverages both supervised learning approaches to argument component classification and NLP techniques for identification of support relations in a branched, recurrent structure.

Classification-centered models typically use supervised machine learning algorithms to classify argument components as claims or premises (e.g. [7]). But these models often struggle with ambiguity when determining to which class an

argument component belongs, largely because an argument component’s classification is highly dependent on its relation to other statements. Relation-based models aim to predict relationships between argumentative statements (e.g. [3]), but are constrained by the granularity of the input data, such as sentences versus clauses. This suggests they may not be sufficiently sensitive to account for scenarios where argument components span multiple sentences, or where a single complex sentence may contain many different argument components. Further, relation-based models do not cope well with stand-alone argument components that cannot be easily related to others.

We describe ongoing work to develop an AM pipeline (Figure 1) that combines classification-centered models with relation-based models to address these problems. We claim that using relation-based methods to adjust preliminary classification likelihoods can improve argument component classifications made by machine learning algorithms alone. We also claim that a recurrent method of combining argument components can overcome the input constraint problem experienced by relation-based models. We argue that the branched, recurrent structure proposed in Figure 1 can better discriminate argument components over classification-centered designs and is more sensitive to the range of what can serve as input to current relation-based models.

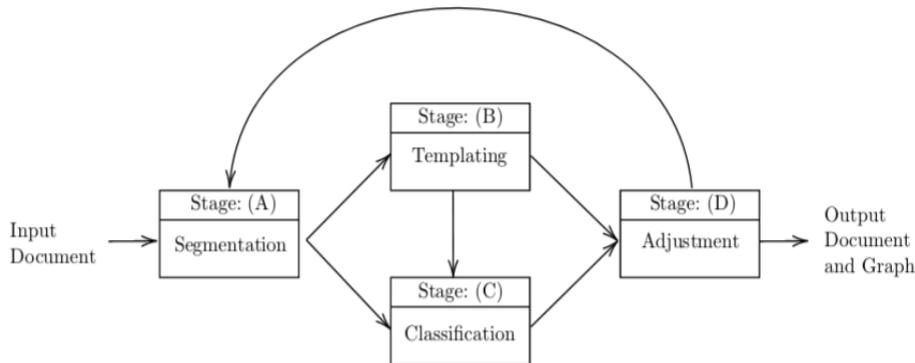


Fig. 1. The Recurrent Argument Mining Pipeline

While other works input text into a linear pipeline (e.g. [7,8]) and combine classification and relation-based methods (e.g. [5]), we believe our approach is the first to propose a branched, recurrent structure that aims to leverage benefits of both while reducing drawbacks. Further, we propose a four-stage process that provides: an extension to current shallow text classification methods called part-of-speech-tying for classifying argument components using both context and content-based features (Stage C); a novel method for creating argument relation templates (Stage B); a novel method to improve argument component classification by enriching their likelihood measures with additional relation-based information (Stage C); and a method for taking argument relation templates and adjusting them, given initial classifications made in Stage C (Stage D).

2 Pipeline Architecture

This section details the expected behaviour of different stages in our pipeline. Due to space considerations, we will briefly describe the pipeline’s input and output, but provide more detail of the critical stages.

An **Input Document** will come as raw textual data taken from a Wikipedia article. We will use the IBM Watson Debater dataset¹ for argument component classification training and overall evaluation, as it also uses Wikipedia data.

Stage A: Segmentation will perform clause tokenization, which is the smallest individual textual unit that argument components could possibly be. Later in Stage D, unused argument components that cannot be mapped using the argument relation template will be combined into new, disparate statements and returned to the end of this stage to be used as input back into Stages B and C, allowing consideration of argument components that span multiple clauses.

Stage B: Templating will take the segmented text from Stage A as input and output argument component classifications and an argument relation template, which is a graph whose nodes and edges correspond to the segmented text and their support relations, respectively.

The purpose of constructing an argument relation template is to extract classification information from the structure of the text, since edges represent support relationships between argument components, the argument relation template helps us to classify these components. Like Cocarascu and Toni [4], we aim to identify relations using LDA and sentiment analysis. However, we will extend this idea with additional NLP techniques. First, we will perform LDA and topic modelling to group statements by topic, since we assume related statements are contained in the same topic. Second, between each pair of statements in each topic, we will compute the mutual shared information and cosine similarity scores, which provide a measure of how similar the statements are, given their constituent words and usage. We assume related statements share similar sentiment values, so will calculate the similarity of sentiment scores for each pair of statements within a topic. Finally, we will track the distance between two statements in the text by how many clauses separate them. We will connect two statements together by taking the values from each step and combine them into single metric m . If the value of m is greater than some threshold T , a support relation exists between them. Although our assumptions may not hold in every case, our idea is that by combining each score into a single metric and setting an optimal threshold, two statements will still be linked. The direction of this edge will be determined by comparing the values calculated from topic modelling. The statement with the higher score will be considered a *claim* or *both* statement, as higher scores may suggest closer adherence to a given topic.

After we create the argument relation template, we will determine the classification to which each node belongs for use in Stage C. Each node’s incoming and outgoing support edges will indicate that node’s classification and will be

¹ Available at: https://www.research.ibm.com/haifa/dept/vst/debating_data.shtml, last accessed 19 August 2019.

determined through pre-established classification rules. For instance, a node N with $S_{inc} > 0$, where S_{inc} is the number of incoming support edges, is classified as either a *claim* or *both* statement.

Stage C: Classification will take the individual statements from Stage A and classify each statement by its role in the text as a *claim*, *premise*, *neither*, or *both*. We first apply a supervised learning approach, trained on the IBM Debater dataset, and aim to improve the classifications from this with information from the argument relation template from Stage B. We have developed our own shallow-learning based models called part-of-speech-tying (POST) that account for both content and context, from we will test and select the best classifier. POST models produce a likelihood measure for each class and express them as tuples: $t_S = (CL, PL, NL, BL)$, where CL , PL , NL , and BL , represent *claim*, *premise*, *neither* and *both* likelihoods, respectively. From the tuples, we will determine statements that are ambiguous, which we define as statements whose likelihoods are close to the same value. We will factor in the relation-based classifications output from Stage B by adding to or subtracting from the likelihoods of all statements based on the amount of support relations of that statement’s node in the argument relation template. For example, nodes with more incoming edges are likely to be claims, so for those nodes, we will increase CL by some weighted amount w . This should improve upon the ambiguous classification of previous AM models.

Stage D: Adjustment will take the argument relation template created in Stage B and the argument component classifications from Stage C as input. The template guides the identification of which classified statements from Stage C could be combined together and returned to Stage A as new statements for re-classification, provides a heuristic to determine which statements to evaluate first for connectivity, and determines when the stage proceeds to final output. We will first count the number of each type of argument component in each topic. Second, starting with the topic which contains the greatest cumulative number of argument components, we will compare different sub-trees in the template for structures that closely match the numbers described by the collection of argument component counts. When a match or close match is identified, we will label the nodes with the appropriate text segments by re-calculating the mutual information and cosine similarity scores, perform sentiment analysis, and factor in textual distance similar to the procedure described in Stage B; however, in this stage only relevant pairs are calculated *depending on their classification*. Combinations of statements with impossible connectivity (i.e. two claims or two premises) will be excluded. If any of the classified statements cannot be fit into the graph, pairs of those statements will be concatenated together to form a new statement and returned to Stage A, where the process repeats. We hope to be able to use classification information to identify the most appropriate text to feed back into Stage B. The pipeline terminates once a “best fit” has been found, or a recurrence threshold showing no further progress is reached. Best fit occurs when the template matches the classification information. This recurrent

approach should improve upon the input sensitivity problem experienced by relation-based models, which will be evaluated in future work.

Finally, an **Output Document and Graph** will be generated in a new mark-up document along with the final graph for visualisation.

3 Conclusion and Future Work

We have proposed an argument mining pipeline with a branched, recurrent structure that combines elements of both classification-centered and relation-based models. We believe that this structure will address ambiguity found in component classification and input sensitivity in relation-based models which are not sufficiently fine-grained; this claim will be evaluated fully in future work.

Additionally, we intend to expand our pipeline to include support and attack relations and plan to test the pipeline outside of the Wikipedia domain. Due to the growing body of research in attention mechanisms for text classification, we also intend to evaluate methods using recurrent neural networks with attention or gate recurrent unit mechanisms in Stage C. Finally, we intend to apply our pipeline to reasoning problems, such as finding winning arguments in text.

References

1. Bosc, T., Cabrio, E., Villata, S.: Tweeties Squabbling: Positive and Negative Results in Applying Argument Mining on Social Media. In: *Comp. Models of Arg.* pp. 21–32 (2016)
2. Boschi, G., Young, A.P., Joglekar, S., Cammarota, C., Sastry, N.: Having the Last Word: Understanding How to Sample Discussions Online. *ArXiv preprint arXiv:1906.04148* (2019)
3. Carstens, L., Toni, F.: Using Argumentation to improve classification in Natural Language problems. *ACM Trans. on Internet Technology* pp. 30 – 41 (2017)
4. Cocarascu, O., Toni, F.: Detecting Deceptive Reviews Using Argumentation. In: *Proc. of the 1st Int. Works. on AI for Privacy and Security.* pp. 9:1–9:8 (2016)
5. Galassi, A., Lippi, M., Torroni, P.: Argumentative Link Prediction using Residual Networks and Multi-Objective Learning. In: *Proc. of the 5th Works. on Arg. Mining.* pp. 1–10. *ACL* (2018)
6. Haddadan, S., Cabrio, E., Villata, S.: Yes, we can! Mining Arguments in 50 Years of US Presidential Campaign Debates. In: *Proc. of the 57th Conf. of the Assc. for Comp. Ling.* pp. 4684–4690 (2019)
7. Lippi, M., Torroni, P.: Argument mining: A machine learning perspective. In: *Int. Works. on Theory and App. of Formal Arg.* pp. 163–176 (2015)
8. Lippi, M., Torroni, P.: Margot: A web server for argumentation mining. *Exp. Sys. with App.* pp. 292–303 (2016)
9. Mayer, T., Cabrio, E., Lippi, M., Torroni, P., Villata, S.: Argument Mining on Clinical Trials p. 12 (2018)
10. Moens, M.F.: Argumentation Mining: Where Are We Now, Where Do We Want to Be and How Do We Get There? In: *Post-Proc. of the 4th and 5th Works. of the Forum for Inf. Ret. Eval.* (2017)
11. Stab, C., Gurevych, I.: Parsing argumentation structures in persuasive essays. *Journ. of Comp. Ling.* (2017)