Towards an Argument Knowledge Base for Autonomous Debating Systems

Charles Stubbs[1], Tommy Yuan[1]

[1]Department of Computer Science, University of York, Deramore Lane, York, YO10 5GH, UK

Abstract

This paper addresses the labour-intensive nature and expert annotator requirements of manual argument annotation by providing an initial sentence-level pipeline for argument identification and sentence relationship prediction. Before establishing our approach, we review current efforts to produce combined techniques and the available abstract pipelines. Next, we use the methods indicated in this review to produce standalone identification and relationship prediction methods. Finally, we apply these methods in a pipeline to produce a combined approach. The presented results demonstrate effective sentence identification but shortcomings in relationship detection. Overall, this paper suggests that combined approaches to argumentation are a promising area requiring further research.

Keywords

Argument Mining, Argument Identification, Sentence Relationship Prediction, Pipeline, Automation,

1. Introduction

Argumentation is a core part of human discourse, allowing us to convey "inclinations, attitudes or opinions"[1] regarding our views. It also involves our ability to identify "relevant assumptions and conclusions"[2] and supporting or conflicting arguments. From these, we create maps of these arguments and their evidence. This process is known as argument mining.

A promising development in the field is IBM's Project Debater[3], where connected modules are used for a system to debate humans on specific structured topics. An important module within this is the AKB (Argument Knowledge Base). It aims to "capture the commonalities between different debates"[3] in the form of a dataset. Overall, the model relies heavily on the AKB (produced manually) to provide general arguments in different areas.

Requiring expert annotators, it is intractable to keep pace with the rate of data generation. Automating this manual process is a valuable asset to discourse. A dataset could be created rapidly with minimal human interference through the automatic extraction of threads of arguments.

A major limiting factor in automation potential is the lack of public, large-scale, standardised datasets containing the original texts. A focus on creating such datasets and allowing the possibility to combine these to create increased volume is essential for machine learning success.

This paper proposes a new combined method for sentence-level argument mining that is successful using features from unannotated texts. Overall this paper contributes an initial step

CMNA'22: Workshop on Computational Models of Natural Argument, September 12, 2022, Cardiff

crs553@york.ac.uk (C. Stubbs); tommy.yuan@cs.york.ac.uk (T. Yuan)
https://crs553.github.io/ (C. Stubbs); https://www-users.cs.york.ac.uk/~tommy/ (T. Yuan)

to automation by combining state-of-the-art methods in a pipeline fashion, with texts requiring no annotation. Thus overall minimising human argument identification and relationship recognition between identified arguments.

2. Related Work

This section reviews common structures of pipelines and frameworks, followed by individual methods for argument mining, and finally, we look at the most promising combined method.

2.1. Argument Mining Pipelines

Argument mining follows a similar structure to manual analysis; Lawrence and Reed[4] define this in Figure 1. The figure presents the tasks in order of complexity, it suggests that mining is iterative. This iterative process attempts to address the issue of context-dependent evidence. They also represent the three critical areas of identification by dashed boxes.

Lippi and Torroni[5] present a simplified pipeline compared to the previous diagram. The stages within the pipeline are argumentative sentence detection, argument component boundary detection and argument structure prediction. Following this, they identify three options for argumentative sentence detection, which are:

1. Two binary classifiers in a pipeline differentiate arguments and non-arguments, then classify the component type.
2. A multi-class classifier assumes that a sentence contains at most one argument.
3. A set of binary classifiers for each argument component in the chosen model.

In the next step, Lippi and Torroni[5] recognise three overlapping instances for component boundary detection. The first is the case where a portion of a sentence is a component, the second is that more than one argument component can be in a sentence, and the third is the instance a component contains more than one sentence. Finally, Lippi and Torroni use argument structure prediction to establish links between components; they recognise that the underlying argument model will influence this step.

These methods contain similar sub-tasks. Lippi and Torroni[5] provide the most intuitive, clearly explaining each stage. Lawrence and Reed[4], whilst providing a comprehensive method, detail each stage's purpose with greater abstraction.

2.2. Standalone Methods

Moens et al.[6] detects argumentative sentences with each as a vector of features. Using a multinomial naïve Bayes classifier and a maximum entropy model, the features tested had a maximum accuracy of 74%; there is an accuracy reduction with legal text, speculated to be due to a lack of training examples. Palau and Moens[7] continue this work, moving on to proposition classification, achieving an F-score of 71%, 6% higher than a context-free grammar (CFG) when using maximum entropy and SVM model on the ECHR corpora. They then review a CFG for argument structure identification, obtaining a 60% accuracy and a 70% F1 score.

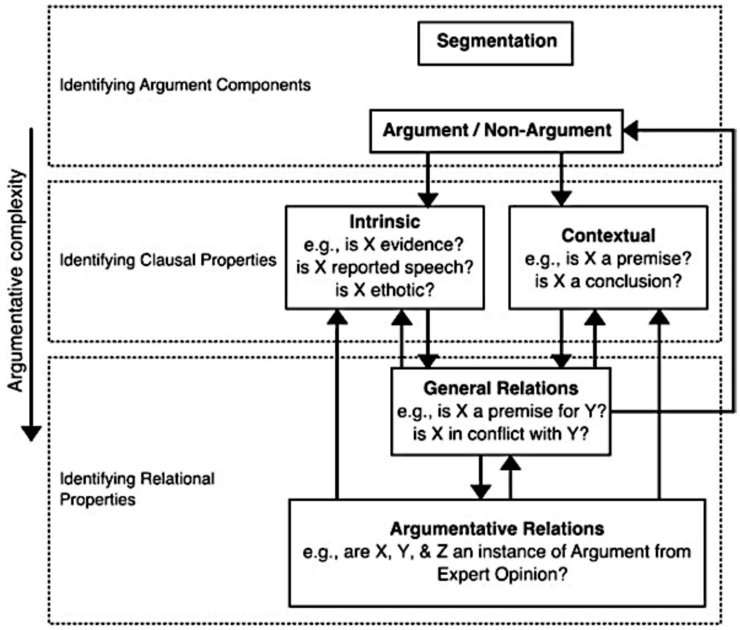

Figure 1: Argument Analysis Diagram[4]

Goudas et al.[8] evaluate classifiers by testing methods with "state-of-the-art features" and "new features". The new set of additional features, combined with state-of-the-art features, present an advantage in both argumentative section detection and claim and premise extraction.

Ajjour et al.[9] present a method for textual segmentation as a "token labelling problem" using three machine learning models. As a baseline, Ajjour re-implements Stab's method[10], reporting a worse F-score of 82.7 when compared to Stab's results. They then offer methods of type SVM, CRF and Bi-LSTM. Evaluating the results, Bi-LSTM produces significantly better results using all features compared to Stab's original implementation.

Cabrio and Villata[11] presents the most commonly used component detection and relations prediction methods. It outlines that in most cases, SVMs, parsing algorithms, and logistic regression are among the three most frequently used approaches. This popularity could indicate the most promising practices currently concerning argument mining.

Using the initial three stages by Lippi and Torroni[5] introduced in 2.1, a classifier of some form must be chosen in all cases. They state the most common, including SVMs, and that there is "no clear evidence to tell which classifier should be preferred", thereby inferring that all listed classifiers would be suitable.

Expanding on using SVMs, Mochales and Moens[12] use SVMs and a maximum entropy model to classify conclusions and premises on the ECHR corpus. They attain "68.1% and 74.0% F1 measure, respectively". This best-case scenario was found when using features such as combined words and text statistics (i.e. sentence length, average word length and the number of punctuation marks).

2.3. Combined Method

Discourse Indicators(DI) are used by Lawrence and Reed[13] to identify a connection between adjacent propositions and whether they support or attack the argument. According to them, DI are often used as a feature within argument mining; however, they instead present a standalone method. Webber et al.[14] defined DI as explicit linguistic expressions between statements. These either indicate support or conflict and can thus be used to infer general patterns regarding some arguments. In Lawrence and Reed's[13] results, this method of identification provides a high precision value of 0.89 but a low recall - 0.04. Consequently, the F1 score is low. They conclude it could be a "useful component" but is "inadequate" "unless supplemented by other methods".

The subsequent approach considered by Lawrence and Reed[13] is "Topical Similarity", which demonstrates how a change in topic relates tree structure formation. Their method performs best on non-directed edges. Whilst it has high precision, it fails to find most connectives falling victim to a low recall.

The final individual approach uses an argumentation scheme structure. They present two commonly used structures, followed by a naiv̈e Bayes classifier and a POS-tagger and add the tag frequency as an individual feature. This method gives F-scores of 0.93 and 0.75 - which consider the correct identification of one proposition type.

Lawrence and Reed[13] combine the aforementioned approaches in a supervised manner. Firstly by applying using discourse indicators, assuming that they are always correct, then identifying the scheme structure, and finally connecting any missed schemes through topical similarity. They obtain a better recall and F1 score from this combination than from any individual method.

Whilst their method provides good results, annotation is required as it uses already identified propositions. Building on their work, our combined method aims to identify sentence-level propositions and feed them into a relationship identification method.

3. Methods

This section introduces the dataset and the tools used to parse it. We then progress on to the individual methods and end with the combined method.

3.1. Dataset

Automating annotations using large generalised datasets presents a challenge due to the lack of standardisation. Dataset structures differ greatly, as some original texts can be missing. Lawrence and Reed[13] use a cleaned version of AIFdb's AraucariaDB[1]. Since it lacks most of the original texts, it is inadequate for our use case. Consequently, our results are not directly comparable to Lawrence and Reed's approach[13].

Stab[10] creates the Essay Corpus v2[2], which contains the initial text. Each text contains premises, claims and one major claim. This dataset is ideal due to the texts being available

[1] AraucariaDB: https://corpora.aifdb.org/araucaria

[2] Stab's Essay Corpus v2: https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2422

and being able to be split into training and testing sets. To load the dataset we use Németh's loader[3] and Milz's corpus parser [4] [15]. This corpus was chosen over the former by Lawrence and Reed[13], as it contains the original text for all annotations.

3.2. Discourse Indicators

The proposed method is to use Discourse Indicators similarly to Lawrence's method[13]. The implementation will be standalone; it will not differentiate between support and conflict and will be tested on three sets of indicators[5]. The first is Lawrence's original[13] (example: not and no). Another is called Premise indicators (example: in view of, as shown). And a combined set.

Before running the method, we apply pre-processing, where the data is split into sentences with punctuation removed. Figure 2 is a sample of the algorithm. Given a text, adjacent sentences are parsed to the lower case. At the end of argument 1 and the start of argument 2, the sentences are checked to ascertain if the connecting ends contain indicators. If either does, they are tagged as related.

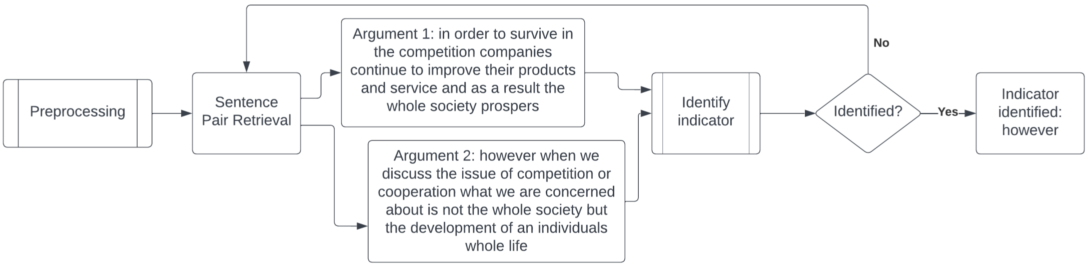
Preprocessing
Sentence Pair Retrieval
Argument 1: in order to survive in the competition companies continue to improve their products and service and as a result the whole society prospers
Argument 2: however when we discuss the issue of competition or cooperation what we are concerned about is not the whole society but the development of an individuals whole life
Identify indicator
Identified?
No
Yes
Indicator identified: however

Figure 2: Example of the algorithm using Lawrence's original indicators[13]

3.3. Machine Learning Models

SVMs are shown to be useful in classification tasks. As outlined in the research section, Lippi and Torroni[5] offer a general guideline for the stages at which classification can be used. The first of these is a binary classifier to discern which sentences contain arguments. In all cases listed in Lippi and Torroni's work, some form of classifier must be used. They indicate no clear preference but list SVMs amongst other state-of-the-art options.

3.3.1. Argumentative Classifier

Using the aforementioned guidelines to construct an SVM, we first aim to classify if a sentence is argumentative; if not, then it is discarded. A sentence is argumentative if it contains a premise, claim or major claim. Using features from Mochales' research [12] and adding some additional features, we create the features in Table 1. Table 2 provides an example of the features in Table 1.

[3] Németh's loader: https://github.com/negedng/argument_BERT/blob/master/preprocessing/data_loader.py
[4] Milz's parser: https://github.com/Milzi/arguEParser
[5] 3 Sets of Indicators: https://github.com/crs553/Towards-Automatic-Argument-Mining/tree/main/Models

Table 1
Features for Argumentative Classifier(AC) Dataset

Feature	Description
Sentence Vectors	Vectorisation with stop words removed
TriBefore	Vectorisation of last three words in the sentence before with stop words removed
TriAfter	Vectorisation of first three words in the sentence after with stop words removed
DocPosition	Sentence position in a document
PuncNumber	Number of punctuation marks a sentence contains

Table 2
Feature example of Table 1 (vectorisation not applied for readability)

Feature	Example
Sentence Vectors	point view firmly believe attach importance cooperation primary education
TriBefore	individuals whole life
TriAfter	first all cooperation
DocPosition	0.31
PuncNumber	1

The original sentence was “From this point of view, I firmly believe that we should attach more importance to cooperation during primary education”, the example is from Stab’s Corpus. These textual features are vectorised through TF-IDF vectorisation. When vectorising, unigrams and bigrams are included, and stop-words are removed before sentence processing. If a sentence is first or last in the text, “$$$” is used instead to indicate that there is no adjacent sentence - this is specifically for the TriBefore/TriAfter feature. To form the labels, the assumption is the sentence is argumentative if it has a relation within the dataset.

3.3.2. Relationship Occurrence Model

Moving on to another step in Lippi and Torroni’s guidelines[5], we present a method for relationship identification using an SVM. All sentence relations must be constructed for the document in preparation for the identification process. They are related if they contain a relation in the annotated dataset. When preparing the data, we create features as shown in Table 3. Relations are created between all argumentative sentences.

3.4. Combining into a pipeline

From the literature review, we find that Discourse Indicators were determined to be “inadequate”, when standalone, “for identifying even a small percentage of the argumentative structure”[13]. In order to obtain the best results, we now combine this method with the Machine Learning Model proposed in the previous subsection.

Figure 3 shows an abstract overview of how this combined pipeline will work. Exploring this diagram further, the original text files are inputted and prepared by creating the features

Table 3
Features for Relationship Occurrence Classifier (ROC)

Feature	Description
Sentence1	Vectorisation of first sentence
Sentence2	Vectorisation of second sentence
Three1	Vectorisation containing last three words of first sentence
Three2	Vectorisation of first three words of second sentence
Pos1	POS tagging of sentence 1
Pos2	POS tagging of sentence 2
Similarity	Cosine similarity between two vectorised sentences

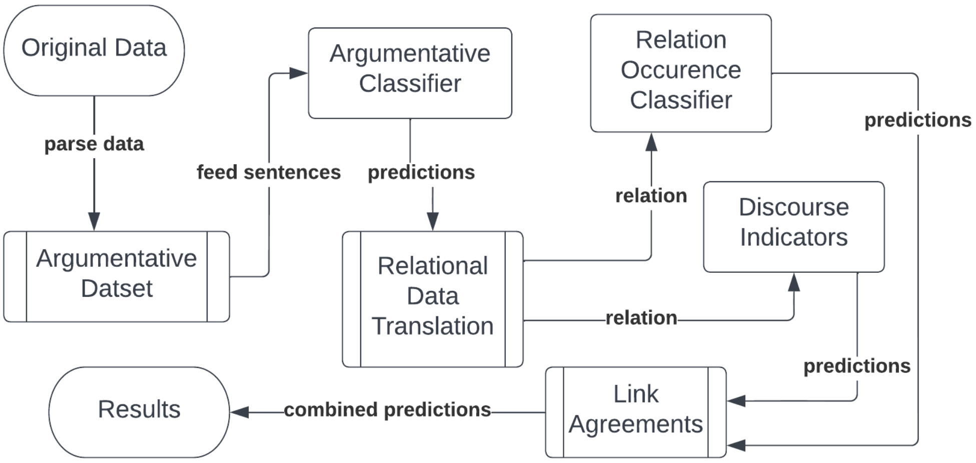

Figure 3: Combined Pipeline Abstract Component Overview

outlined in Table 1. The dataset is then fed into the Argumentative Classifier, where it is predicted on. These predictions are then passed to the relational data translation component, where only argumentative predictions are kept. These classes are then paired with all possible sentence combinations within their text file. Features are created from these classes, as shown in Table 3, and then the relational dataset is produced.

The final methods in Figure 3 are the Relationship Occurrence Classifier and the Discourse Indicators (DI) algorithm. The predictions from these results are combined with DI predictions being taken as always correct for the positive case. If it does not predict a link, the decision is then passed to the result of the link classifier. It should be noted that Discourse Indicators will be based on the best-performing indicators.

At each stage, selected features were constructed from the original text. The aim was to minimise the need for initial annotation and human influence other than correct labels for model training.

4. Results

4.1. Individual Components

4.1.1. Discourse Indicators

The Discourse Indicator (DI) results in Table 4. Reviewing the three sets of indicators shows that premise indicators have the best precision, whereas combined have the best recall and F1-score. Combined and premise indicators perform better because both contain indicators specifically developed for this dataset. In all cases, recall is low; this can indicate a high number of false negatives, possibly due to some arguments not containing the indicators listed. Since the premise indicators produce the best accuracy, we will use this in our combined method.

Table 4
Results Discourse Indicators (DI) for different sets of indicators

Type	Precision	Recall	F1
Discourse	0.60	0.02	0.04
Premise	0.74	0.01	0.03
Combined	0.65	0.03	0.06

4.1.2. Argumentative Classifier

Argumentative classifier, shown in Table 5, performs well in all testing metrics. This indicates its reliability in argument classification.

Table 5
Results report for Classifiers

	Precision	Recall	F1	Case Number
AC	0.77	0.97	0.86	1697
non-arg label	0.72	0.19	0.3	446
arg label	0.77	0.97	0.86	1251
ROC	0.60	0.62	0.15	30651
not-linked label	0.96	0.6	0.74	28928
linked label	0.08	0.62	0.15	1723

4.1.3. Relationship Occurrence Classifier

Relationship Occurrence Classifier (ROC) performs better in recall and F1 than DI at relationship prediction. The lower precision is down to a skew in the dataset towards the not-linked label. This is because data appears challenging to separate. This conclusion is supported by the fact that the number of sentences in the linked label is 16 times smaller than in the non-linked case.

4.2. Overall Pipeline

By combining these components to produce the pipeline, the results are shown in Table 6. We can conclude that when compared to the independent ROC and DI after AC indicated in the aforementioned table, the overall pipeline performs better than its individual counterparts in terms of recall in both cases and F1 in the DI case. It should be noted that in Table 6 overall recall and F1 are only for the positive case, whilst precision is averaged.

Table 6
Score report for pipeline

Stages	Precision	Recall	F1	-
AC	0.75	0.97	0.85	-
DI	0.78	0.23	0.12	-
ROC	0.73	0.40	0.15	-
Final pipeline				Support Cases
Overall	0.62	0.53	0.14	28427
not-linked	0.96	0.63	0.76	26771
linked	0.08	0.53	0.14	1656

Examining the Support Cases in Table 6 we can see a significant skew towards the not-linked case; this imbalance is likely due to the data not being easily separable. This issue affects relationship methods more when the AC stage is included than when operating individually.

5. Conclusion

The indicators experiment performs better on the premise and combined indicators than the discourse indicators. This is likely due to premise indicators being explicitly developed for Stab's dataset. Further work on more accurate general indicators could be valuable in producing a more reliable method.

Argumentative classification is the most reliable part of the pipeline and effectively identifies whether a sentence is argumentative. It should be noted that the "arg" label performs better than the "non-arg" label, with an F1 score difference of 0.83, indicating the SVM with the features outlined above can effectively identify arguments at a sentence level.

Link relationships are a different story, with a low F1-score compared to their accuracy and recall; this indicates a high rate of false positives. This is probably due to the imbalance towards the non-linked cases, with 14 out of 15 cases in this category.

A similar result is seen in the pipeline; AC performs similarly to the individual components and thus would be ideal for future use. Following this success, we prepare these identifications with all other combinations. For relationship prediction, both cases perform well in terms of precision and recall but perform worse in terms of F1-score. This is probably due to the original instances containing a randomised dataset of all possible outcomes, and the training set is limited to only propositions contained within it rather than all potential link cases.

Whilst the pipeline[6] does not produce the ideal result, we can conclude that for unannotated

[6] Code repository for all cases: https://github.com/crs553/Towards-Automatic-Argument-Mining

data, a process filtering at each stage is appropriate as an initial component towards an automatic argument mining system. How this pipeline should be formed requires more research, and full exploration of the stages outlined in Lippi and Torroni's guidelines[5] should be explored. In its current form, it cannot provide annotations for argument knowledge bases, such as the AKB in Project Debater[3], due to its overall accuracy and the unknown repeatability across different domains. Further analysis is required to determine the cause of classification error.

References

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

[2] P. Besnard, A. Hunter, *Elements of argumentation*, volume 47, MIT press Cambridge, 2008.

[3] N. Slonim, Y. Bilu, C. Alzate, R. Bar-Haim, B. Bogin, F. Bonin, L. Choshen, E. Cohen-Karlik, L. Dankin, L. Edelstein, et al., An autonomous debating system, *Nature* 591 (2021) 379–384.

[4] J. Lawrence, C. Reed, Argument mining: A survey, *Computational Linguistics* 45 (2020) 765–818.

[5] M. Lippi, P. Torroni, Argumentation mining: State of the art and emerging trends, *ACM Trans. Internet Technol.* 16 (2016).

[6] M.-F. Moens, E. Boiy, R. M. Palau, C. Reed, Automatic detection of arguments in legal texts, in: *Proceedings of the 11th international conference on Artificial intelligence and law*, Association for Computing Machinery, New York, NY, USA, 2007, pp. 225–230.

[7] R. M. Palau, M.-F. Moens, Argumentation mining: the detection, classification and structure of arguments in text, in: *Proceedings of the 12th international conference on artificial intelligence and law*, Barcelona, Spain, 2009, pp. 98–107.

[8] T. Goudas, C. Louizos, G. Petasis, V. Karkaletsis, Argument extraction from news, blogs, and social media, in: *Hellenic Conference on Artificial Intelligence*, Springer, 2014, pp. 287–299.

[9] Y. Ajjour, W.-F. Chen, J. Kiesel, H. Wachsmuth, B. Stein, Unit segmentation of argumentative texts, in: *Proceedings of the 4th Workshop on Argument Mining*, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 118–128.

[10] C. Stab, I. Gurevych, Parsing argumentation structures in persuasive essays, *Computational Linguistics* 43 (2017) 619–659.

[11] E. Cabrio, S. Villata, Five years of argument mining: a data-driven analysis., in: *International Joint Conferences on Artificial Intelligence IJCAI*, volume 18, 2018, pp. 5427–5433.

[12] R. Mochales, M.-F. Moens, Argumentation mining, *Artificial Intelligence and Law* 19 (2011) 1–22.

[13] J. Lawrence, C. Reed, Combining argument mining techniques, in: *Proceedings of the 2nd Workshop on Argumentation Mining*, Denver, Colorado, US, 2015, pp. 127–136.

[14] B. Webber, M. Egg, V. Kordoni, Discourse structure and language technology, *Natural Language Engineering* 18 (2012) 437–490.

[15] T. Milz, *Argue - an argumentation mining and modelling approach for classication of argumentative discourse units and structures.*, University of Passau, Master's thesis, 2017.