Towards Robust Argument Mining: An analysis of datasets, labelling schemes and models

Group ID: 15

Abstract

Argument Component Identification (ACI) is an important task within the rapidly developing NLP field of Argument Mining. One of its most exciting applications is within Automated Writing Evaluation tools to enhance students' writing skills in an accessible and democratised way. Unfortunately though, to reach classroom applications, ACI needs to reach exceptional levels of robustness. Working towards this goal, here we show how, when applying transformers on ACI, different segment representation schemes can affect models' performance in a way that is dataset-specific. Further, we present a set of simple yet powerful adversarial attacks which can challenge even the best performing models in our experiments. Applying such attacks as data augmentations will conceivably increase model reliability.

1 Introduction

Writing is an indispensable ability in modern society, predicting success and achievement in school, work and everyday life (Graham, 2019). Far from an innate ability, the acquisition of solid writing skills involves a wide complex of cognitive processes (Kellogg, 2008) and is entirely dependent on education. Research robustly shows that frequent feedback on writing exercises and assessments is a crucial component in the development of writing competence in students at all stages of education (Biber et al., 2011). However, providing frequent feedback can be time-consuming for teachers, especially when working with younger students and larger classes, effectively limiting the benefits of this instructional practice.

Fortunately, modern developments in Natural Language Processing (NLP) are producing robust Automated Writing Evaluation (AWE) systems capable of scoring, commenting and correcting essays, providing students with feedback that is as effective in enhancing writing quality as teacher feedback (Graham et al., 2015; Nunes et al., 2022). Despite the proven effectiveness of AWE in the classroom, it must be recognised that writing skills are highly multi-faceted. While it is easier for an AWE system to provide feedback, for example, on grammar and syntax, the problem becomes substantially harder when we want to analyse and evaluate the argumentative structure and soundness of a piece of text. Precisely at this point recent developments in the field of Argument Mining (AM) come into play.
Argument Mining can be defined as "the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language" (Lawrence and Reed, 2019). Within AM, it is also useful to our discussion to distinguish different tasks, of increasing complexity and mutual dependence: Argument Component Identification, Clausal Properties Identification and Relational Properties Identification.
Even though some recent advances gathered considerable excitement towards AM (Slonim et al., 2021), AM is still a novel field within NLP, showing as a result a lot of gaps and opportunities for research within its methodologies. In the current work, we are going to focus exclusively on the task of Argument Component Identification (ACI), the fundamental building block of any downstream AM system, of particular relevance to AWE.

Only recently (Park and Cardie, 2014), this task has been approached from a sequence labelling perspective at the token level. It remains unclear however which segment representation (SR) is best suited in this scenario. The first subject of our investigation will therefore be how different

SR schemes used in Named Entity Recognition will affect the task. We hypothesis that the often discarded "IO" scheme is not poorly suited in this scenario. Furthermore, motivated by the idea that AWE systems need to be especially robust when applied in the classroom, we devise and perform a set of adversarial attacks on our ACI systems. Given a known issue with model robustness in the ACI field (Reimers et al., 2019), we hypothesise that even the simplest attacks will have a non-negligible effect on models' performance.

Our contributions will be the following: we first survey the current literature on AM, with a specific focus on the sub-task of ACI, to show how gaps in the literature can be turned into opportunities for research and potential improvement of the current state of the art. For the first time in the ACI task, we then compare different SR schemes to show how they have a non-trivial influence on the models' final performance. We further perform a set of adversarial attacks on our best performing models, suggesting how these could be used as task-specific data augmentations in future work. We release all the code used in our experiments in `ArgMiner`[1], a package for processing SOTA AM datasets using standardised methodology.

2 Background

Argument Component Identification was traditionally tackled by taking the simplifying assumption that sentences are the smallest argumentative unit, and by also dividing the task into two classification problems, namely first separating argumentative from non-argumentative text, and then further identifying the type of arguments within argumentative sentences (e.g. Moens et al., 2007; Florou et al., 2013). More recently, the "argument component = phrase" simplification has been dropped by a body of studies trying to detect argumentative units at the clause level (Park and Cardie, 2014; Stab and Gurevych, 2017; Eger et al., 2017; Sardianos et al., 2015). These latter explorations approached ACI with a method reminiscent of Named Entity Recognition (NER), namely by doing sequence labelling at the token level in one single classification step (Petasis, 2019).

A first unexplored issue in the literature is worth mentioning at this point. When the state of the art in ACI moved toward a NER-like approach, the popular IOB segment representation (SR) scheme has been adopted (Ramshaw and Marcus, 1995). Even if this choice of standard can easily go unnoticed and be accepted uncritically, especially given that it has been widely adopted across NLP (in Tjong Kim Sang and De Meulder, 2003, as a notable example), studies in NER show that the particular choice of segment representations, especially when dealing with multi-token named entities (as it is the case in ACI) has influences on downstream models' performances both in terms of accuracy and training speed (Cho et al., 2013). Furthermore, results on the use of different segment representations are discordant when different datasets and languages are used (Alshammari and Alanazi, 2021), suggesting that a choice of SR should be made on a case by case basis to maximise performance. We therefore take this as a first point for our exploration, further considering the fact that, to our knowledge, no studies on the impact of segment representation schemes on ACI have been done in the past.

Continuing our survey on the AM literature on ACI, it must be noted that it was established practice to approach the task with a heavy feature-engineering component, even when using neural models such as biLSTMs + CRF (binary Long-Short Term Memory networks with Conditional Random Fields; Eger et al., 2017; Ajjour et al., 2017). More recently, following a general trend within NLP research, it has been appreciated that neural models alone can achieve state of the art performances in the right circumstances, eliminating the need for heavy feature engineering (Petasis, 2019). This is especially true when using transformers (Vaswani et al., 2017), as they have been just shown to reach a new state-of-the-art in ACI (Alhindi and Ghosh, 2021). Given how late this result is (relative to established practices in ACI), we sought as our second main line of exploration the effect of different transformers architectures on the task at hand.

Our final investigations are inspired by sizeable research efforts on adversarial attacks on Deep

[1] Note that the package is in an early access release stage. It can be found here: https://github.com/namiyousef/argument-mining

Learning models, originating in Computer vision (Akhtar and Mian, 2018) and now rapidly developing in NLP (Zhang et al., 2020). Given that our aim, rendered explicit also in our choice of datasets, is to make ACI systems robustly usable for AWE, it is only natural to try to devise adversarial-like attacks to test said robustness. The reason for this is twofold. Firstly, it has been shown that ACI systems are particularly vulnerable to idiosyncrasies in the training corpora (due to, for example, annotation strategies, type of text annotated, age and style of the writers; Reimers et al., 2019), which make their generalisation performance poorer than expected by evaluation on the test set only. This is also the motivation behind extensive efforts into developing larger and more varied corpora (Stab and Gurevych, 2017; Stab et al., 2018; Alhindi and Ghosh, 2021), and why we are using two of the largest available in our project.

Secondly, looking at AWE usage in the classroom, it has already been shown how students can occasionally devise strategies to exploit the underlying systems in their favour (Powers et al., 2002; Nunes et al., 2022). To preserve the didactic purpose of AWE, we must therefore focus on making ACI robust.

3 Methods

3.1 Data

We base our work on two different datasets, which can motivate different types of investigations. The first one is the popular Argument Annotated Essays corpus (AAE; Stab and Gurevych, 2017). This contains 402 essays written by college students, which have been annotated for premises, major claims and claims, all at the clause level. We use this dataset mainly to be able to compare our results with previous work in the field. However, it is argued in the literature that essays written by college students can already be a simplification of the ACI problem, given the level of education and structure often present in such texts (Alhindi and Ghosh, 2021). It is also more useful from an AWE perspective to have a system that can give feedback to those that need it the most, namely students that still need to learn how to structure argumentative texts (Alhindi and Ghosh, 2021). This is why we trained our models separately also on the newly released PERSUADE (PER) corpus [2]. This rich dataset includes over 15,000 argumentative essays written by U.S. 6-12 graders, which have been manually annotated by expert annotators for the following categories: claim, counterclaim, rebuttal, evidence, lead, position, and concluding statement. It is easy to see how this more elaborate annotation scheme can benefit an AWE mechanism.

For both datasets, we respected the original train/test set split provided by the authors. We also created a 30% validation set from the training set in both cases, stratifying the split so that the essays in both folds contained roughly the same total number of argument components, as measured by tokens.

3.2 Models

As mentioned Section 2, following the current state of the art in ACI we approached the problem as a sequence labelling task. We decided to use a Transformer architecture to tackle this, mainly for two reasons.

First of all, even if LSTM networks have long been the baseline in these type of tasks (Huang et al., 2015), it has recently been shown that transformers are superior in the ACI setting, especially when dealing with essays written by younger, more argumentatively disorganised students (Alhindi and Ghosh, 2021). Using transformers has also the added advantage of eliminating the feature engineering process from the pipeline. Secondly, the “HuggingFace Transformers” library (Wolf et al., 2020) allows fast and efficient manipulation of large pre-trained transformer models to readily do fine-tuning on a variety of downstream tasks. Using the library allowed us to speed up implementations while also cutting on computational costs.

We specifically compared two transformer implementations. The first one is a pre-trained RoBERTa (Liu et al., 2019, `roberta-base` on Huggingface). The second is Big Bird (Zaheer et al., 2021, `bigbird-roberta-base` on Huggingface). Facilitating our analysis, the two models (as found in the “HuggingFace” library) were trained on the same corpora, with Big Bird warm-started from a RoBERTa checkpoint. The main difference between the two is the implementation of the attention mechanism. While RoBERTa uses full attention in its attention blocks, resulting in computations that scale quadratically with sequence length, Big Bird

[2] The corpus will be soon publicly accessible at the following link: `https://github.com/scrosseye/PERSUADE_corpus`. We have written confirmation from the authors to use the dataset in the present work.

uses a mixture of sparse and stochastic attention to attain a linear dependency on sequence length. To allow this difference to stand out during training, expecting better results in the task when using a longer sequence length, we set the maximum sequence length to 512 for RoBERTa and to 1024 for Big Bird.
Further details on the training procedure are schematised in Figure 1.

3.3 Segment representation

We adopted three different segment representation strategies, as present in Alshammari and Alanazi (2021) and Cho et al. (2013). These are, in order of increasing complexity: IO, BIO, BIEO. These schemes assign each token a label if it is in the beginning (B), inside (I), end (E) or outside (O) a given sequence.

3.4 Adversarial attacks

We are performing adversarial attacks (in the loose sense of the term) on our models by using largely semantic-invariant transformations of the essay texts. The motivation behind the approach is that the argumentative structure of the text will be unaffected by the specific words used to express a stance. Not only that, but the form of the argument will be also unaltered by the specific position taken, which allows us to use also transformations that can reasonably alter the semantics of passages. We used the `nlpaug` ((Ma, 2019)) package to construct our attacks, which can broadly be divided in word substitutions and word/expression insertions. These are, ordered by the expected negative effect on models' performance:

1. Random synonym substitution. Synonyms are drawn from wordnet (Fellbaum, 1998).

2. Random spelling mistakes. Random words were substituted for misspelled ones at random. Given that both datasets deal with essays that are handwritten, the mistakes are selected from a pool of common mis-spellings in the English language (see Ma, 2019).

3. Random antonym substitution. Also drawn from wordnet (Fellbaum, 1998).

4. Random keyword insertion. A random, filler, argumentative keyword y was added after the subject of a phrase. $y \in \{$ therefore, actually, basically, seriously, highly, really, totally, absolutely$\}$

5. Random filler expression insertion. Common filler expression, as defined and listed in the AAE manual for annotators (Stab and Gurevych, 2014), were added at random at the beginning of phrases.

Further note that only one attack strategy was applied at a time. In every phrase and for each strategy, a minimum of 1 and a maximum of 10 augmentations were applied, with an attack probability of $p = .3$ for/after each word in a phrase.

Devising these attacks was an important part of our exploration: if the model is severely affected by them, they could be used as data augmentation strategies during training. However, this would also mean that the model is not performing as we would expect, defying our intuition of what it is actually learning. Furthermore, these are also important in the context of AWE systems. A good AWE system should be robust to spelling errors made by students (e.g. it should not severely impact their argumentation scores because of spelling errors) and should not be susceptible to bad actors trying to pad their essays with 'filler words'.

4 Experiments

4.1 Experimental setup

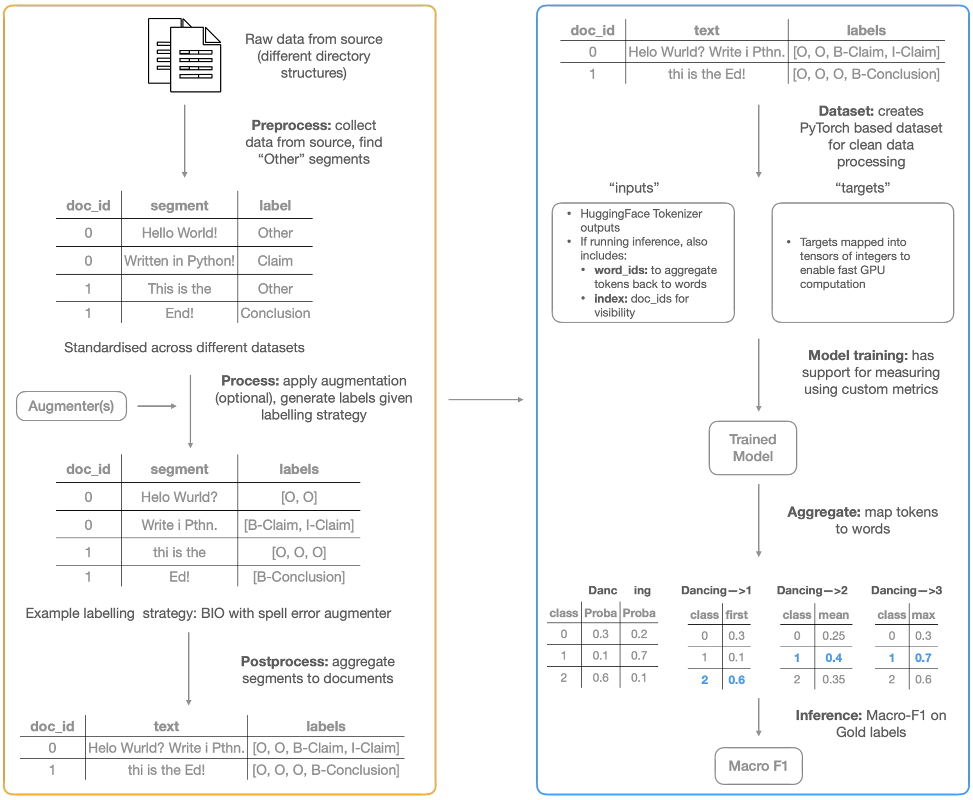
To reiterate, for ACI in written essays, we systematically study the different behaviour of two popular NLP models, namely Big Bird and RoBERTa on two selected datasets. We do this by investigating their performance with different segment representations, namely IO, BIO and BIEO. This will ultimately give us a total of six models to compare.

The following table contains the hyperparameters we used to train all our models.

Table 1: Model hyperparameters

Parameter	BigBird	RoBERTa
Batch size	4	4
Optimiser	Adam	Adam
Learning rate	2.5×10^{-5}	2.5×10^{-5}
Max length	1024	514
No.of epoch	20	20
loss	CrossEntropy	CrossEntropy

After training the models, we compared them with regards to a number of evaluating metrics on the test datasets.


Raw data from source (different directory structures)
Preprocess: collect data from source, find "Other" segments
Standardised across different datasets
Augmenter(s)
Process: apply augmentation (optional), generate labels given labelling strategy
Example labelling strategy: BIO with spell error augmenter
Postprocess: aggregate segments to documents
Dataset: creates PyTorch based dataset for clean data processing
"inputs"
"targets"
HuggingFace Tokenizer outputs
If running inference, also includes:
word_ids: to aggregate tokens back to words
index: doc_ids for visibility
Targets mapped into tensors of integers to enable fast GPU computation
Model training: has support for measuring using custom metrics
Trained Model
Aggregate: map tokens to words
Danc
ing
Dancing—>1
Dancing—>2
Dancing—>3
Inference: Macro-F1 on Gold labels
Macro F1

doc_id	segment	label
0	Hello World!	Other
0	Written in Python!	Claim
1	This is the	Other
1	End!	Conclusion

doc_id	segment	labels
0	Helo Wurld?	[O, O]
0	Write i Pthn.	[B-Claim, I-Claim]
1	thi is the	[O, O, O]
1	Ed!	[B-Conclusion]

doc_id	text	labels
0	Helo Wurld? Write i Pthn.	[O, O, B-Claim, I-Claim]
1	thi is the Ed!	[O, O, O, B-Conclusion]

doc_id	text	labels
0	Helo Wurld? Write i Pthn.	[O, O, B-Claim, I-Claim]
1	thi is the Ed!	[O, O, O, B-Conclusion]

class	Proba	Proba
0	0.3	0.2
1	0.1	0.7
2	0.6	0.1

class	first
0	0.3
1	0.1
2	0.6

class	mean
0	0.25
1	0.4
2	0.35

class	max
0	0.3
1	0.7
2	0.6

Figure 1: End-to-end training procedure for all of our models using our custom `Argminer` package

4.2 Evaluating metrics

We implemented a number of common intrinsic and extrinsic metrics to evaluate the trained NLP systems. They are as follows:

- Macro-F1: This macro-f1 score is evaluated on the Gold labels, and not on the prediction tensors. This involves mapping subtokens back to words after prediction and finding segment intersections against the original text (for a given class and document). Matches that exceed a certain threshold[3] are considered true positives (tp). Unmatched ground truths are false negatives (fn) and unmatched predictions are false positives (fp) (GTU, 2022). The macro-F1 score is the average F1 score across classes, given by: $\bar{F}1 = \sum_c tp_i/(tp_i + 0.5(fn_i + fp_i))$

- Recall: recall is given by the following $recall = tp/(tp + fn)$. A macro score can be calculated exactly as above.

- Precision: precision is given by the following $precision = tp/(tp + fp)$. A macro score can be calculated as described above.

The F1 score is of particular relevance in this NLP classification problem since it is proven to be a robust metric when classes are not balanced. Benefiting from its combined nature of recall and precision, F1 score is selected as the main evaluating metric to be used in this project.

4.3 Adversarial attacks protocol

The best trained models on the AAE dataset for each architecture (so one for Big Bird and one for RoBERTa) will be further evaluated using our adversarial examples. The Macro F1 score as a result of application of the adversarial examples will then be compared with that of original test datasets to conclude the efficacy of such attacks. We chose to run the adversarial attacks on the AAE dataset only for practical purposes.

5 Results

5.1 F1 score across SR schemes

Results after training the models on both the PERSUADE and AAE corpus with different SR strategies are summarised in the following tables.

After 20 epochs, RoBERTa and Big Bird are observed to converge to a steady sliding average F1 score on both datasets. As shown above, RoBERTa

[3] For this experiment a threshold of 0.5 was used

Table 2: Model F1 score: RoBERTa

	io	bio	bieo
AAE	0.784	0.789	0.742
PERSUADE	0.499	0.493	0.493

Table 3: Model F1 score: Big Bird

	io	bio	bieo
AAE	0.697	0.762	0.712
PERSUADE	0.494	0.471	0.474

has shown slightly better F1 score when the trained model is tested on both datasets, achieving 0.789 at its highest.

5.2 Adversarial attacks results

As mentioned, we conducted 5 adversarial attacks on our best-performing models on the AAE dataset, as discussed in subsection 4.3. These two models were RoBERTa (BIO) and Big Bird (BIO). The findings are summarised in tables 4 and 5. It can be seen that custom filler and spelling errors are associated with notably high attack success rates.

Table 4: F1 score with adversarial attacks and relative reduction [RoBERTa - bio; AAE dataset]

Attack Name	After attack	Reduction(%)
Custom filler	0.401	49%
Synonym	0.651	17%
Spelling error	0.569	28%
Key word change	0.688	12%
Antonym	0.701	11%

The relative efficacy of the adversarial attacks can be ordered as: custom filler > spelling error > synonym > key word change > antonym.

6 Discussion

Interestingly, results are mostly in line with our initial hypotheses. SR schemes have a non-negligible effect on model performance, with the often discarded IO scheme being a viable option in some cases. Furthermore, adversarial attacks are shown to have varying degrees of impact, though all of them severely affect model scores.

Let us expand on the implications of these results, framing them in the wider problem of model generalisation in ACI.

Table 5: F1 score with adversarial attacks and relative reduction [BigBird - bio; AAE dataset]

Attack Name	After attack	Reduction(%)
Custom filler	0.448	41%
Synonym	0.698	8.4%
Spelling error	0.569	25%
Key word change	0.762	0%
Antonym	0.695	8.8%

6.1 SR schemes matter

The findings of model training clearly show that Segment Representation schemes have an effect on final performance. This is perhaps unsurprising, given that the number of labels that the final classification layer of the model has to output changes with SR strategies. Though often overlooked by practitioners, these results are in line with previous findings in the NER literature (Alshammari and Alanazi, 2021; Cho et al., 2013), which concluded that the choice of optimal SR strategy is dependent on the model and task at hand. Our work extends these conclusions and underlines their importance in ACI.

It is worth drawing our attention specifically on the performance of the IO scheme. This scheme is often discarded in NER tasks because, lacking a boundary tag, it does not have the ability to distinguish between successive named entities of the same type, or even named entities that are interlaced with one another. Note, however, that none of these two problems apply in the ACI task: successive argument components of the same type are often classified as the same by annotators, and nested arguments are rare, at most appearing in the forms of parenthetical clauses. Therefore, it is provocative to see IO performing as the best labelling scheme for both models on the PERSUADE dataset, but not on AAE (see tables 2 and 3). We hypothesise that this occurs because PERSUADE has more argument labels to predict. Using IO on it will therefore reduce the total number of effective labels needed to be predicted by the model (e.g. for a claim: only {`I-CLAIM` , `O`} and not {`B-CLAIM`, `I-CLAIM`, `O`}), possibly reducing the overall complexity of the classification task. Using a similar line of reasoning we interpret BIEO as being the worse performing scheme in almost all comparisons.

As far as practical implications are concerned, we encourage future ACI practitioners to include the evaluation of SR schemes in their pipelines.

6.2 Understanding adversarial attacks success

Even given the known issues in the ACI field regarding model generalisation (Reimers et al., 2019), results of our adversarial attacks on the models look striking. Even the simplest strategies, namely synonym substitution and spelling mistakes substitutions, affected the models' predictions as measured by the F1 score by over 10% .

The addition of custom filler expressions reduces performance by a massive 40-50% , which is remarkable given the fact that these expressions were specifically taken from the ones the AAE annotators were told to ignore (Stab and Gurevych, 2014). A possible explanation for this, however, is that this prior knowledge given to the annotators could not be acquired by the model given the data alone. This opens up the possibility that models could be further improved by a more specific encoding of prior knowledge, a prime example of which is the annotation guidelines used in the creation of the datasets.

A surprising result, defying our initial expectations and somehow showing a degree of robustness in the models, is the fact that random antonym substitution is among the least effective adversarial attacks for both models. This can be taken to imply that the model is really focusing on argumentative structure rather than argument sentiment or content, with a desirable degree of invariance to those.

Taken together, these findings reveal the potential of adversarial attacks on the ACI task, as in many other fields in Deep Learning (Zhang et al., 2020; Akhtar and Mian, 2018). Given our implementation of these adversarial examples, it is straightforward to think of incorporating them in the training process as data augmentations. We hypothesise that this would greatly benefit the robustness of the training process, and plan to analyse this in future work.

6.3 The effect of attentional window size

We purposely deferred our discussion of the Big Bird - RoBERTa comparison to this point. So which one is better here? Sparser, larger attentional windows or having smaller but full attention blocks?

On the one hand, one might answer that RoBERTa is the better performing model for the task at hand, showing a macro F1 score higher than Big Bird on the test set of both datasets, on close to all SR schemes. We would therefore conclude that having a larger, sparser, attention window size does not have a substantial impact on the ACI task.

On the other hand, however, we cannot neglect the adversarial attacks results. Even if Big Bird performed worse on the plain AAE test set, it was far more robust than RoBERTa to adversarial perturbations in terms of percentage performance reduction. In almost all attacks, this allowed Big Bird to surpass RoBERTa in raw performance. Under this perspective, we would conclude that having larger, sparser, attention blocks renders the model less susceptible to local information (especially when it can be perturbed), and generally more robust for this type of task.

What is preferable then? We will leave the reader to draw their conclusions, limiting ourselves to note how effective adversarial perturbations of these type can prove to be even in the model selection process.

6.4 Limitations

We recognise some limitations in the present work, which can hopefully be addressed by future research.

Firstly, given the low computational resources at our disposal, we did not have a chance to perform extensive hyperparameter tuning on our models, conceivably not reaching their full potential despite having used recommended values[4].

Secondly, with regards to the datasets, it is a difficult pursuit to measure how well 'learning' transfers from one dataset to another, which was one of our original objectives. In particular this is because the mappings are not straightforward. For instance if comparing the AAE dataset with PERSUADE [5], it is plausible that the following label maps are valid $(Claim) \rightarrow (Claim, Counterclaim)$ and $(MajorClaim) \rightarrow (Position)$ and vice versa. However this means that all the other examples are to be labelled as 'Other' which severely decreases dataset power. Further, the mappings described above are also not correct all of the time, leading

[4] For completeness, it must be stated that our models failed to reach SOTA in the ACI task. See (GTU, 2022) for some examples

[5] Our work looked into the state-of-the-art datasets for ACI. However, another dataset (Alhindi and Ghosh, 2021, ARG2020) was made public towards the end of this project. Future work should also consider how ARG2020 can be mapped to the AAE and PERSUADE datasets.

to substantial ambiguity. Further work should look into using domain expert knowledge to bridge the gap between labels across multiple datasets.

6.5 Outlook: the future of generalisation in ACI

A lot of directions for future work are present in the field of Argument Mining and ACI specifically, some of which were expanded by our present contributions.

Investigating the performance of ACI models across datasets and with adversarial examples is an especially pressing direction, given the importance of ACI for numerous applications in argument mining and AWE specifically. Our approach to creating adversarial examples can be enhanced with more specific, prior-informed augmentations, specialised to the domain of model application. The hope is that then, the same augmentations can be used to make model training robust.

Given that transformers have been shown to be especially flexible for this task, both here and elsewhere (Alhindi and Ghosh, 2021), further work should go into determining what architectural details are important. Here we show that the attentional window size can significantly impact robustness, but more efforts are needed to corroborate our findings.

Finally, sizeable effort should also be devoted to creating larger corpora. Given how widely different corpora can change between each other, a unified corpus of argumentative essays coming from writers of different ages and countries would be of immense benefit to the ACI field.

7 Conclusion

Argument Component Identification is an important task within the rapidly developing NLP field of Argument Mining. One of its most exciting applications is within Automated Writing Evaluation tools, to enhance students' writing skills in an accessible and democratised way. Unfortunately though, to reach classroom applications, ACI needs to reach exceptional levels of robustness.

In the present work we made some efforts towards this, showing how, when applying transformers on this task, using in concert different datasets and different segment representation schemes can give better insights into model performance. Furthermore, we present a set of adversarial attacks which can challenge the best performing models in our experiments. We hope this can serve as inspiration for ACI practitioners trying to build reliable models.

Acknowledgments

We would like to thank Yihong Chen for her guidance and advice. We would also like to thank Scott Crossley and the Learning Agency for providing permission to use the PERSUADE dataset for this project. Finally, while we didn't get to including it in our experiments due to time constraints, we'd like to thank Debanjan Ghosh for expediting the cleaning and end release of the ARG2020 dataset.

References

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit Segmentation of Argumentative Texts. In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128, Copenhagen, Denmark. Association for Computational Linguistics.

Naveed Akhtar and Ajmal Mian. 2018. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *arXiv:1801.00553 [cs]*. ArXiv: 1801.00553.

Tariq Alhindi and Debanjan Ghosh. 2021. ”Sharks are not the threat humans are”: Argument Component Segmentation in School Student Essays. Publisher: arXiv Version Number: 1.

Nasser Alshammari and Saad Alanazi. 2021. The impact of using different annotation schemes on named entity recognition. *Egyptian Informatics Journal*, 22(3):295–302.

Douglas Biber, Tatiana Nekrasova, and Brad Horn. 2011. The Effectiveness of Feedback for L1-English and L2-Writing Development: A Meta-Analysis. *ETS Research Report Series*, 2011(1):i–99. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2333-8504.2011.tb02241.x.

Han-Cheol Cho, Naoaki Okazaki, Makoto Miwa, and Jun'ichi Tsujii. 2013. Named entity recognition with multiple segment representations. *Information Processing & Management*, 49(4):954–965.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural End-to-End Learning for Computational Argumentation Mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Eirini Florou, Stasinos Konstantopoulos, Antonis Koukourikos, and Pythagoras Karampiperis. 2013. Argument extraction for supporting public policy formulation. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 49–54, Sofia, Bulgaria. Association for Computational Linguistics.

Steve Graham. 2019. Changing How Writing Is Taught. *Review of Research in Education*, 43(1):277–303. Publisher: American Educational Research Association.

Steve Graham, Michael Hebert, and Karen R. Harris. 2015. Formative Assessment and Writing: A Meta-Analysis. *Elementary School Journal*, 115(4):523–547. Publisher: University of Chicago Press.

GTU. 2022. Feedback prize - evaluating student writing.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv:1508.01991 [cs]*. ArXiv: 1508.01991.

R. T. Kellogg. 2008. Training writing skills: A cognitive developmental perspective.

John Lawrence and Chris Reed. 2019. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818. Place: Cambridge, MA Publisher: MIT Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*. ArXiv: 1907.11692.

Edward Ma. 2019. NLP Augmentation.

Marie-Francine Moens, Erik Boiy, Raquel Mochales 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 '07, pages 225–230, New York, NY, USA. Association for Computing Machinery.

Andreia Nunes, Carolina Cordeiro, Teresa Limpo, and São Luís Castro. 2022. Effectiveness of automated writing evaluation systems in school settings: A systematic review of studies from 2000 to 2020. *Journal of Computer Assisted Learning*, 38(2):599–620. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jcal.12635.

Joonsuk Park and Claire Cardie. 2014. Identifying Appropriate Support for Propositions in Online User Comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Georgios Petasis. 2019. Segmentation of Argumentative Texts with Contextualised Word Representations. In *Proceedings of the 6th Workshop on Argument Mining*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Donald E. Powers, Jill C. Burstein, Martin Chodorow, Mary E. Fowles, and Karen Kukich. 2002. Stumping e-rater:challenging the validity of automated essay scoring. *Computers in Human Behavior*, 18(2):103–134.

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking using Transformation-Based Learning. In *Third Workshop on Very Large Corpora*.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and Clustering of Arguments with Contextualized Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. 2015. Argument Extraction from News. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 56–66, Denver, CO. Association for Computational Linguistics.

Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, Liat Ein-Dor, Roni Friedman-Melamed, Assaf Gavron, Ariel Gera, Martin Gleize, Shai Gretz, Dan Gutfreund, Alon Halfon, Daniel Hershcovich, Ron Hoory, Yufang Hou, Shay Hummel, Michal Jacovi, Charles Jochim, Yoav Kantor, Yoav Katz, David Konopnicki, Zvi Kons, Lili Kotlerman, Dalia Krieger, Dan Lahav, Tamar Lavee, Ran Levy, Naftali Liberman, Yosi Mass, Amir Menczel, Shachar Mirkin, Guy Moshkowich, Shila Ofek-Koifman, Matan Orbach, Ella Rabinovich, Ruty Rinott, Slava Shechtman, Dafna Sheinwald, Eyal Shnarch, Ilya Shnayderman, Aya Soffer, Artem Spector, Benjamin Sznajder, Assaf Toledo, Orith Toledo-Ronen, Elad Venezian, and Ranit Aharonov. 2021. An autonomous debating system. *Nature*, 591(7850):379–384. Number: 7850 Publisher: Nature Publishing Group.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. Parsing Argumentation Structures in Persuasive Essays. *Computational Linguistics*, 43(3):619–659. Place: Cambridge, MA Publisher: MIT Press.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic Argument Mining from Heterogeneous Sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674, Brussels, Belgium. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*. ArXiv: 1910.03771.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big Bird: Transformers for Longer Sequences. *arXiv:2007.14062 [cs, stat]*. ArXiv: 2007.14062.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial Attacks on Deep-learning Models in Natural Language Processing: A Survey. *ACM Transactions on Intelligent Systems and Technology*, 11(3):24:1–24:41.