4-2021

BERT Argues: How Attention Informs Argument Mining

Ting Chen
University of Richmond

BERT Argues: How Attention Informs Argument Mining

Ting Chen

Honors Thesis

Under supervision of: Dr. Joonsuk Park

Submitted to:

Department of Math and Computer Science

University of Richmond

Richmond, VA 23173, USA

April 2021

Abstract

Argument mining is the automatic identification and extraction of structure from argumentative language. Previous works have constrained the argument structure to forming strictly trees in order to utilize efficient tree-specific techniques. However, arguments in the wild are unlikely to exhibit this constrained structure. Given the recent trend of fine-tuning large pre-trained models to reach state of the art performance on a variety of Natural Language Processing (NLP) tasks, we look to leverage the power of these deep contextualized word embeddings towards the task of non-tree argument mining. In this paper, we introduce a new pipeline which utilizes pre-trained BERT based models as well as Proposition Level Biaffine Attention and Weighted Cross Entropy Loss for predicting arguments where the structure forms a directed acyclic graph. Our experiments demonstrate the efficacy of using deep contextualized word embedding from BERT based models while also suggesting future directions involving recurrence for modelling hierarchical relationships.

Acknowledgements

I thank Dr. Joonsuk Park for taking the chance to work with me and continuing to do so through the end of my undergraduate studies. His presence, guidance, and encouragement have helped me become the researcher as well as the person I am today.

I also thank Dr. Prateek Bhakta for being my first research professor in Computer Science and for continually giving me advice on research, academics, and life since then.

A nonexhaustive list of the professors I've had over the years that I'd like to thank that have influenced my educational as well as research experience are Dr. Jory Denny, Dr. Lawrence Leemis, Dr. Barry Lawson, Dr. Taylor Arnold, Dr. Cindy Bukach, Dr. Heather Russell, Dr. Iain Murray, Dr. Sharon Goldwater, and Dr. William Ross.

Of course thank you to my family and friends for giving me support and love through these years at Richmond.

Finally I'd like to thank all my classmates as well as the staff at the Math and Computer Science department in Richmond, especially Ms. Kathy Rothert for her work behind the scenes that keeps everything running smoothly.

List of Figures

List of Figures

Table of Contents

Chapter 1

Introduction

The fundamental components of human communication include debate and argument. As the proliferation of online communities spread, more and more users are engaging with one another in describing and justifying their beliefs. It is now apparent that the primary stage for these kinds of interactions in the foreseeable future will be online media. In order to analyze this growing source of data, we look to the automatic identification and extraction of the structure of arguments in natural language, known as argument mining (Lawrence and Reed, 2019). Despite the numerous successes of applying machine learning methods to natural language processing (NLP) tasks, so far current methods have been unable to reliably identify relationships between different argument structures.

While the theory of argumentation, the use of logical reasoning to justify claims and reach conclusions, has a long and storied history (van Eemeren et al., 2019), the field of argument mining is relatively young. One simplifying assumption common to earlier argument mining works is constraining the argument structure to forming one or more trees (Peldszus and Stede, 2015; Stab and Gurevych, 2017). This greatly improves ease of computation as it enables the use of maximum spanning tree-style parsers.

However, arguments commonly found online are unlikely to exhibit exact tree

structures. This observation has led to many approaches for argument mining in the wild with argument structures that are not necessarily constrained to trees. Beginning with structured support vector machines and recurrent neural networks (Niculae et al., 2017), non-tree argument mining has attracted a variety of methods in order to solve the problem of identifying argumentative components as well as the relationships between those components in a way such that the overall structure forms a directed acyclic graph (DAG). Most recently, Bidirectional long short-term memory networks (LSTM) augmented with proposition level biaffine attention (PLBA) inspired by dependency parsing (Dozat and Manning, 2018) have been leveraged to achieve the state of the art (SOTA) performance (Morio et al., 2020).

The most recent and prevalent trend in the field of NLP has been fine tuning large pre-trained Transformer based models, yielding remarkable performance gains on a wide variety of different tasks. Research on how transformers work, in particular the BERT model (Devlin et al., 2019; Rogers et al., 2020), has garnered a great deal of interest in recent years. However, the extent to which we understand how and why Transformer based models performs so well is lacking and remains a promising future research direction.

In this thesis our contribution is the application of transformers in the form of BERT-based models to the task of identifying argument components and predicting the links between the components when they form non tree structures. To this end we utilize the Cornell eRulemaking Corpus (CDCP) (Park and Cardie, 2018), a collection of argument annotations on comments from an eRule-making discussion forum, where the argumentative structures do not necessarily form trees. While using pre-trained BERT models did indeed improved the state of the art on identification of the argumentative components, they struggled with link prediction, suggesting the utility of hybrid models that incorporate recurrence (Tran et al., 2018).

In Chapter 2 we discuss related works and compare our model to existing argument mining models. Chapter 3 explains the formal problem and the approaches we use. Chapter 4 details the experiments, implementation details, and results. Finally in Chapter 5 we discuss conclusions as well as future directions.

Chapter 2

Related Works

2.1 Argument mining

Generally, the process of argument mining can be broken down into several tasks: distinguishing argumentative text from non-argumentative text, identifying the functional components of an argument, and linking the different components based on how they support each other. While many approaches have been proposed for the first two tasks that enjoy relative success (Palau and Moens, 2009; Niculae et al., 2017; Morio et al., 2020; Stab and Gurevych, 2017), the task of predicting relations between argumentative components is extremely challenging as it requires high level representation and reasoning that has alluded most machine learning methods to this day (Cabrio and Villata, 2018).

Argument mining has shown to be useful in many diverse domains. Early notable works include applications towards newspaper articles (Reed et al., 2008), legal documents (Palau and Moens, 2009), and political discussions on online forums (Abbott et al., 2016). With each of these domains, researchers had to create and annotate their own datasets, a costly process resulting in smaller datasets when compared with other standard NLP datasets. One prominent dataset that has been continuously used and analyzed with various argument

mining approaches is Essay (Stab and Gurevych, 2014). Research on this dataset focuses on argument component identification as well as relation identification of persuasive student essays (Persing and Ng, 2016; Potash et al., 2017; Eger et al., 2017). This dataset however has constrained the arguments to forming tree structures.

Niculae et al. (2017) proposed the first non-tree argument mining approaches with a factor graph model along with structured SVMs and bidirectional LSTMs on the CDCP dataset. Galassi et al. (2018) explored the same dataset using LSTMs along with residual network connections in order to focus on link prediction between argument components. The most recent work on the CDCP dataset is Morio et al. (2020), which most closely resembles our work. Inspired by models utilized in semantic dependency (Dozat and Manning, 2018), this work employs task-specific parameterization that uniquely encode argument proposition sequences for each task as well as PLBA for edge prediction as well as edge classification.

Several recent works have also utilized transformer based models towards argument mining tasks. Reimers et al. (2019) utilize contextual word embeddings in the form of BERT and ELMo in order to greatly improve argument/no argument classification as well as propose methods for argument clustering. Importantly, this work did not tackle the challenging task of argument component link prediction using pre-trained language models as we do. Chakrabarty et al. (2019) propose a model based on BERT towards argument component classification as well as relation detection in persuasive online discussions. Specifically, they annotate and analyze argumentative relations in threads from the Change My View (CMV) subreddit.

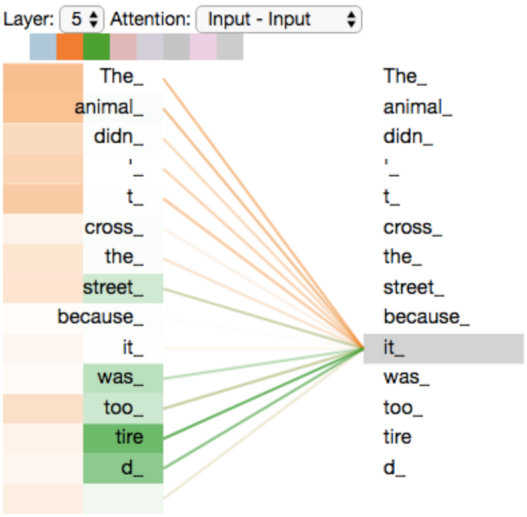

Figure 2.1: Example for multi-head attention mechanism on words within a sentence from Alammar (2018)

2.2 Transformers

The transformer architecture with its self attention mechanism was originally proposed by (Vaswani et al., 2017) as a response to the growing computational and memory requirements of recurrent neural networks (RNN) which were state of the art at the time. RNNs and transformers both aim to model global dependencies in sequential data. By using just multi-head self attention, transformers allow for significantly more parallelizable training, as opposed to the inherent sequential nature of RNNs. Using pre-trained language models based off transformers as part of a transfer learning paradigm quickly became popular for many NLP tasks and achieved state of the art results in the process in many cases. In theory these large pre-trained language models were meant to capture general linguistic knowledge which could then be leveraged and fine tuned for specific tasks. However, as these transformer based language models grew in popularity, they also grew in size as well as in number of parameters. These large and weldy models led to the development of various model compression methods in addition to methods to prevent overparameterization.

Through many works such as (Kovaleva et al., 2019), (Michel et al., 2019), (Voita et al., 2019), one thing we do know is that models based on transformer architectures like BERT (Devlin et al., 2019) are typically overparameterized. That is, many of their heads are redundant and can be pruned without much loss in performance and in some cases pruning even increases performance. This problem may be attributed to the fact that the attention heads learn the same limited set of attention patterns which result in numerous heads using identical attention patterns (Kovaleva et al., 2019). Furthermore, it has been shown that the individual heads that are important in NLP tasks such as neural machine translation (NMT) learn linguistically-interpretable information consistently and pruning the rest of the self attention heads results in similar performance (Voita et al., 2019).

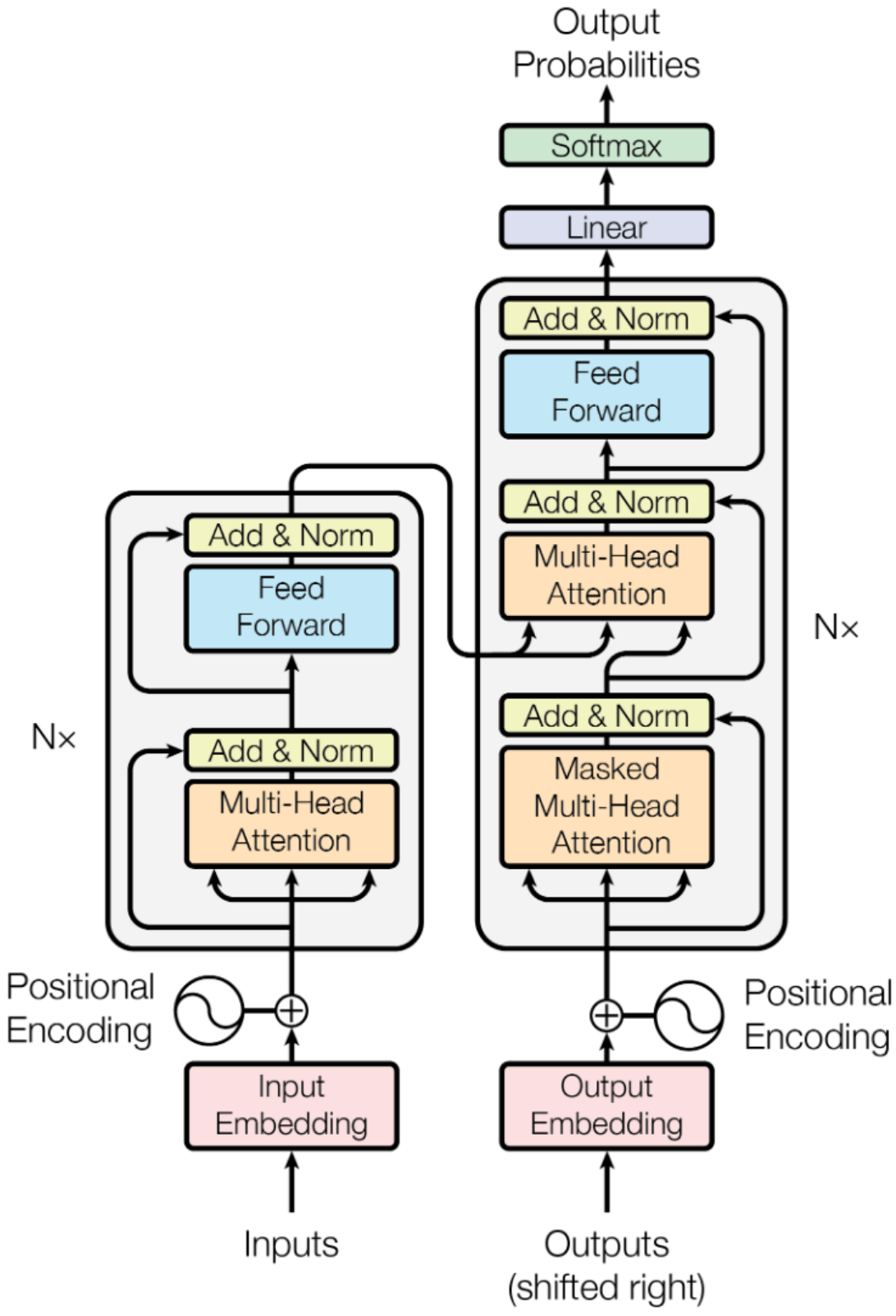

Figure 2.2: Transformer diagram from Vaswani et al. (2017)

Chapter 3

Methods

3.1 Problem Formulation

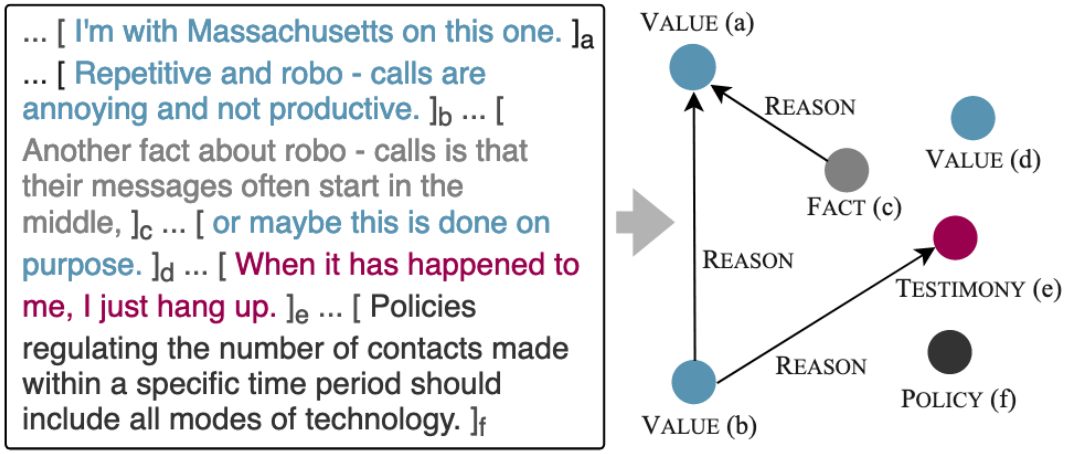

Figure 3.1: Example graph along with corresponding spans in its user comment from the CDCP corpus, from Morio et al. (2020)

In our problem our inputs will be the annotated text of a user comment, each representing an argument. Each of argument's components correspond to a specific span given by the annotated text. The outputs of our model will thus be the argument proposition type of each span as well as it's outgoing edges linking to other components of the argument. While our problem is specified towards

the domain of argument mining, our methods can also be applied to similarly structured data for other tasks.

3.1.1 Inputs

Let A_i be the ith annotation of our dataset of size D, where each annotation represents an argument with its components and relationships forming a DAG. Without loss of generality, assume the text corresponding to the annotation A_i consists of N tokens, with M spans that each correspond to a unique argument proposition (component/node). Let (s_j, e_j) be the starting and ending token indices for the jth proposition span, respectively. Therefore, $0 \leq s_j \leq e_j \leq N$.

3.1.2 Outputs

Given each annotation A_i, for each span j, we predict its proposition type as well as outgoing edges, such that the overall graph is not necessarily a tree.

3.2 Approach

3.2.1 Contextual Word Embeddings from Pre-trained Models

In our approach to argument mining, we look rely solely rely on power of the contextual word embeddings derived from pretrained models that have excelled at other tasks. Thus we replace the LSTMs, GloVe vectors (Pennington et al., 2014), and optional ELMo vectors (Peters et al., 2018) from Morio et al. (2020) with contextual word embeddings from various pre-trained transformer based models.

The first pre-trained model we test is BERT or Bidirectional Encoder Representations from Transformers (Devlin et al., 2019). In our case we tested the `bert-base-uncased` model from the Huggingface `transformers` library. BERT

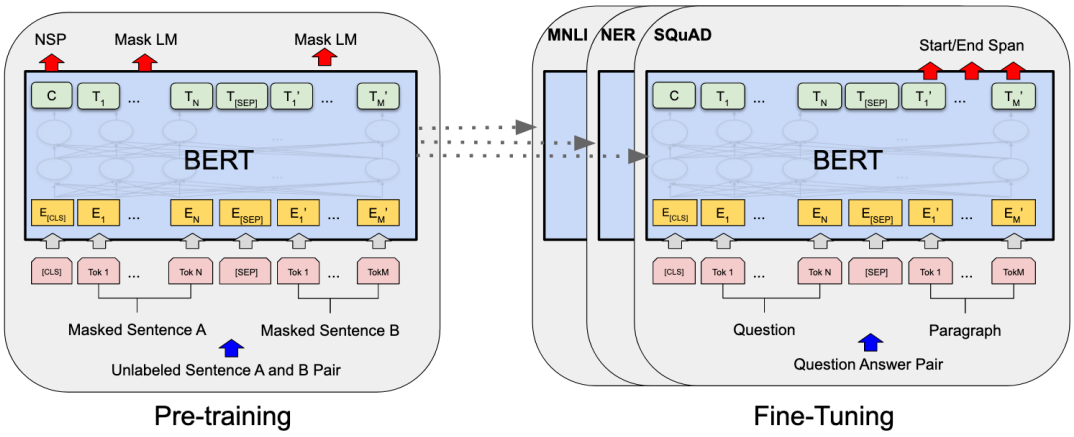

Figure 3.2: Figure from Devlin et al. (2019) detailing the predominant pre-training + fine-tuning paradigm, in this case towards the downstream task of question answering.

is pre-trained using a masked language modeling (MLM) as well as a next sentence prediction (NSP) task on a 3.3 billion word English corpus. For the MLM task, 15% of the words in a given input sentence are randomly masked. Then, the entire masked sequence is fed into the model which subsequently predicts the masked words. The main advantage of using a Transformer based model that relies on attention over a traditional recurrent neural network (RNN) is that the attention mechanism allows for the model to see all the words at once as well choose which words are most important for the given task whereas RNNs usually see one word after another in a sequential manner. The attention mechanism in the Transformer architecture also provides for significant performance boosts over RNNs since it provides for easier parallelization. For the NSP task, two randomly chosen masked sentences are concatenated as inputs for the model. The model predicts whether or not the two sentences naturally follow each other. By training with this multi-task objective, BERT theoretically learns an inner representation of the English language that can then be used to extract useful features for a variety of downstream tasks.

As per standard use, we use the pre-trained BERT model for transfer learning by fine-tuning on the CDCP corpus using a supervised learning objective.

In our implementation we fine tune two separate BERT models, one for proposition classification and another for edge detection. We score each proposition representation with task specific multi-layer hidden perceptrons (MLPs). For the proposition type classification model that is it, however for the edge detection we add a PLBA module, which we will describe in the following subsection.

3.2.2 Multi-layer Perceptrons and Proposition Level Biaffine Attention

Following the proposition type specific representations obtained from our pre-trained model, which we specify as $\boldsymbol{r}_{\text{type},j}$, we apply a one hidden layer MLP as well as a softmax operation to serve as our proposition type classifier. Notation wise, our prediction for the type of proposition j is as follows:

$$\begin{aligned}
\text{ReLU}(\boldsymbol{x}) &= \max(0, \boldsymbol{x}) \\
\text{MLP}(\boldsymbol{x}) &= \boldsymbol{W_2} \cdot \text{ReLU}(\boldsymbol{W_1} \cdot \boldsymbol{x} + \boldsymbol{b_1}) + \boldsymbol{b_2} \\
\hat{\text{type}}_j &= \text{softmax}(\text{MLP}_{\text{type}}(\boldsymbol{r}_{\text{type},j}))
\end{aligned}$$

where ReLU is the rectified linear unit activation function for our MLP and $\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{b_1}, \boldsymbol{b_2}$ are parameters. Any **bold** symbols are vector valued quantities while any unbolded symbols are scalar valued.

Following the edge specific representations obtained from our pre-trained model, which we specify as $\boldsymbol{r}_{\text{edge},j}$, we then proceed to score the representations using the non linear, on hidden layer MLP. Following Morio et al. (2020), we use biaffine attention (Dozat and Manning, 2018) to predict the presence of edges linking pairs of propositions. We compute scores of all pairs of propositions in an annotation using the following operation:

$$\text{Biaffine}_k(\boldsymbol{x}, \boldsymbol{y}) = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}^\intercal \boldsymbol{U_k y}$$

where $\boldsymbol{U_k}$ is a parameter. Notation wise, our prediction for the presence of a directed edge from proposition i to proposition j is as follows:

$$\boldsymbol{e}_i^{\text{src}} = \text{MLP}_{\text{edge}}(\boldsymbol{r}_{\text{edge},i})$$

$$\boldsymbol{e}_j^{\text{trg}} = \text{MLP}_{\text{edge}}(\boldsymbol{r}_{\text{edge},j})$$

$$\hat{\text{edge}}_{i,j} = \text{softmax}(\text{Biaffine}(\boldsymbol{e}_i^{\text{src}}, \boldsymbol{e}_j^{\text{trg}}))$$

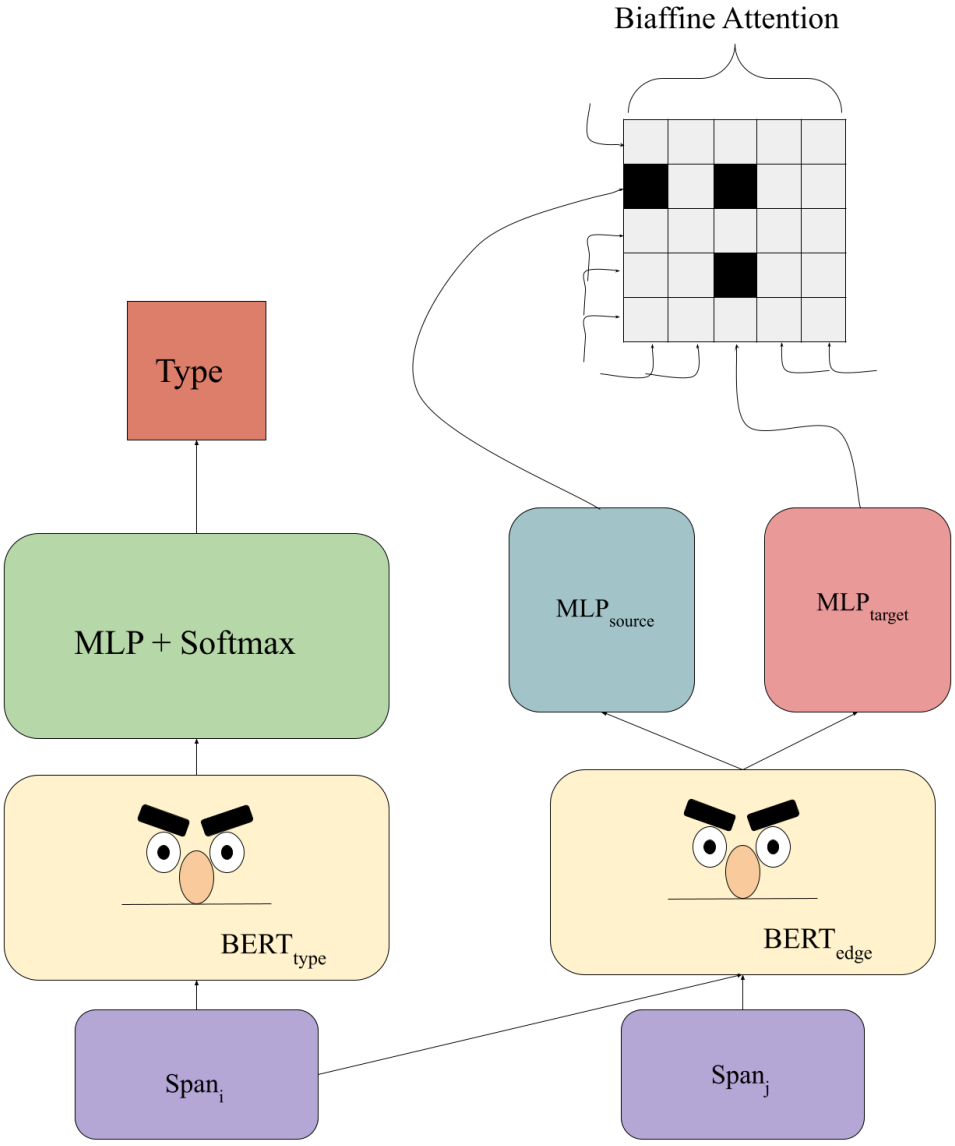

Figure 3.3: High Level Argument Mining Pipeline

3.2.3 Weighted Cross Entropy Loss

In training our two models, we use the PyTorch's (Paszke et al., 2019) implementation of cross entropy loss. Specifically for the edge detection model, this implementation allowed us to counteract the major class imbalance inherent in our dataset. Since we compare all possible pairs of propositions in each annotation and there are relatively few edges linking these propositions, there are orders of magnitude more examples for pairs without edges. The standard cross entropy loss, which we use for training our proposition type classifier model, is as follows:

$$\text{loss}(\boldsymbol{x}, \text{class}) = -\log\left(\frac{\exp(\boldsymbol{x}[\text{class}])}{\sum_{j=1}^{C}\exp(\boldsymbol{x}[j])}\right) = -\boldsymbol{x}[\text{class}] + \log\left(\sum_{j=1}^{C}\exp(\boldsymbol{x}[j])\right)$$

where $\boldsymbol{x}$ is a vector containing the predicted probabilities for C classes of a single observation. PyTorch allows us to modify the loss such that we can introduce weights for each class, like so:

$$\text{loss}(\boldsymbol{x}, \text{class}) = \textbf{weight}[\text{class}]\left(-\boldsymbol{x}[\text{class}] + \log\left(\sum_{j=1}^{C}\exp(\boldsymbol{x}[j])\right)\right)$$

where **weight** is a vector of weights scaling the weights given to each class.

Our procedure for edge detection is similar to that of Tayyar Madabushi et al. (2019) who also investigated the use of BERT on heavily imbalanced data in the form of propaganda detection in news articles. We increase the weight of the minority class, in this case the examples of pairs with an edge linking the two, which intuitively also decreases the proportional cost of the majority class, in this case the examples of pairs without an edge linking the two.

Chapter 4

Experiments

4.1 Setup

4.1.1 Dataset

For this work we utilize the most popular available non-tree argument mining corpus: the CDCP corpus (Park and Cardie, 2018; Niculae et al., 2017) which consists of 731 argument annotations, about 3800 sentences, and about 88k words. Within the corpus, there are five types of propositions: REFERENCE, FACT, TESTIMONY, VALUE and POLICY. There also two types of argumentative edges: REASON and EVIDENCE.

For each of the proposition types: FACT poses a truth value that can be verified with objective evidence, TESTIMONY refers to an objective proposition about the author's personal state or experience, VALUE refers to a proposition containing value judgements without making specific policy claims about what should be done, POLICY refers to a proposition towards a specific course of action, and REFERENCE is a reference to a source (Park and Cardie, 2018). For the edge labels, a proposition a is a REASON for a proposition b when a provides a rationale for b. Likewise, b is EVIDENCE for b when a proves whether or not b is true.

Type	**# in Training Set**	**# in Test Set**
Arguments	581	150
Propositions	3806	973
VALUE	1660	491
POLICY	662	153
REFERENCE	31	1
FACT	622	124
TESTIMONY	822	204
Edges	1081	272
REASON	1042	265
EVIDENCE	39	7

Table 4.1: CDCP Statistics

4.1.2 Baselines

We compare our models to the set of baselines from Niculae et al. (2017), which are factor based models, as well as Galassi et al. (2018), which are neural residuals networks, on test set performance of proposition type classification as well as link prediction.

4.1.3 Implementation

Following Niculae et al. (2017), we perform three fold cross validation on our models, implemented in PyTorch (Paszke et al., 2019), and chose the model with the best evaluation score to run on the test set. Using the AdamW optimizer (Loshchilov and Hutter, 2019), we trained each model for 100 epochs and tuned hyperparameters with the Weight And Biases (wandb) program (Biewald, 2020).

Hyperparameter	**Value**
MLP dimensions	700
MLP dropout	0.25
Mini-batch size	16
Epochs	100
Learning rate	$1e10^{-4}$
AdamW β_1	0.9
AdamW β_2	0.999
Class Weights	[1.0, 60.0]

Table 4.2: Hyperparameter Settings

4.2 Results

Model	**Edge Prediction**	**Type Prediction**	**Average**
Deep Basic: PG	0.22	0.63	0.43
Deep Residual: LG	0.29	0.65	0.47
RNN: Basic	0.14	0.73	0.44
SVM: Strict	0.27	0.73	0.50
TSP + PLBA	**0.34**	0.79	**0.56**
BERT + MLP/PLBA	0.15	**0.86**	0.51

Table 4.3: Test Set F1 performance versus existing models on CDCP

4.2.1 Analysis

From the results we see that the model utilizing only `bert-base-uncased` outperformed all existing argument mining models applied to the CDCP corpus for

the proposition type prediction task. Our model achieves a test set F1 score of 0.86, while the next closest models only achieve scores of 0.79 and 0.65. This is unsurprising as BERT-based approaches have consistently achieved state of the art performance on a variety of NLP tasks, particularly sequence classification tasks. However, our BERT-based model does not perform as well on the the edge prediction task as it only achieves a test set F1 score of 0.15, whereas the best model in this regard, the Morio et al. (2020) model which utilizes task specific parameterization layers as well as LSTMs, achieves an F1 score of 0.34. Overall, when macro averaging across the two tasks, our BERT based model achieves the second best performance on test set F1 score while using significantly less parameters and resources when training.

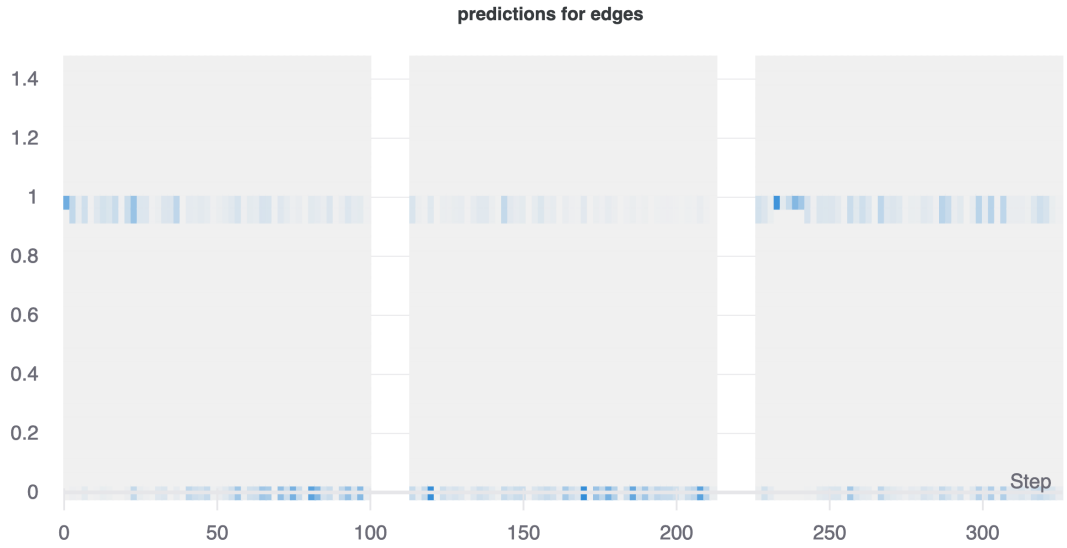

Figure 4.1: Graph of predictions for edges through each cross validation fold's training. In this case 1 is the presence of a link between propositions and 0 is the absence. As the each fold's model trains, the biases towards predicting 1s is corrected.

In 4.1, which was produced using the Weight and Biases program, we see the progression of the predictions for the presence of a link between propositions in the same argument within each cross validation fold. Due to the weighted cross entropy loss, the model of each fold begins by overpredicting the presence of links,

before being corrected as it trains. These training dynamics show the effect the weight of the "1" class has on our model and its subsequent performance.

Chapter 5

Conclusion

In this work we demonstrate the usefulness of applying popular pre-trained BERT based models as deep contextual word embeddings towards the tasks of identifying argument proposition types and identifying relations between these propositions. This work is the first to demonstrate this, in particular on a challenging non-tree argument mining corpus. While our results towards proposition type classification were very promising, there remains room to improve when it comes to edge prediction.

One prominent future direction is to incorporate recurrence along with Transformer-based models in our pipeline. Research has suggested that recurrent neural networks have slight but consistently better performance on modeling hierarchical structures when compared with models that only rely on attention (Tran et al., 2018). Combining the two approaches may yield the best results in the future.

Another possible extension of this work would be to incorporate attention between propositions using a sequence to sequence model. This could potentially improve edge prediction as the sequence to sequence model would encode the context between argument propositions.

Bibliography

Abbott, R., Ecker, B., Anand, P., and Walker, M. (2016). Internet argument corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4445–4452, Portorož, Slovenia. European Language Resources Association (ELRA).

Alammar, J. (2018). The illustrated transformer.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Cabrio, E. and Villata, S. (2018). Five years of argument mining: a data-driven analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5427–5433. International Joint Conferences on Artificial Intelligence Organization.

Chakrabarty, T., Hidey, C., Muresan, S., McKeown, K., and Hwang, A. (2019). AMPERSAND: Argument mining for PERSuAsive oNline discussions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2933–2943, Hong Kong, China. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings*

of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dozat, T. and Manning, C. D. (2018). Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Eger, S., Daxenberger, J., and Gurevych, I. (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.

Galassi, A., Lippi, M., and Torroni, P. (2018). Argumentative link prediction using residual networks and multi-objective learning. In *Proceedings of the 5th Workshop on Argument Mining*, pages 1–10, Brussels, Belgium. Association for Computational Linguistics.

Kovaleva, O., Romanov, A., Rogers, A., and Rumshisky, A. (2019). Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Lawrence, J. and Reed, C. (2019). Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32, pages 14014–14024. Curran Associates, Inc.

Morio, G., Ozaki, H., Morishita, T., Koreeda, Y., and Yanai, K. (2020). Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266, Online. Association for Computational Linguistics.

Niculae, V., Park, J., and Cardie, C. (2017). Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995, Vancouver, Canada. Association for Computational Linguistics.

Palau, R. M. and Moens, M.-F. (2009). Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, page 98–107, New York, NY, USA. Association for Computing Machinery.

Park, J. and Cardie, C. (2018). A corpus of eRulemaking user comments for measuring evaluability of arguments. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-

Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peldszus, A. and Stede, M. (2015). Joint prediction in MST-style discourse parsing for argumentation mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 938–948, Lisbon, Portugal. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Persing, I. and Ng, V. (2016). End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Potash, P., Romanov, A., and Rumshisky, A. (2017). Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1364–1373, Copenhagen, Denmark. Association for Computational Linguistics.

Reed, C., Palau, R. M., Rowe, G., and Moens, M.-F. (2008). Language resources

for studying argument. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., and Gurevych, I. (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.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Stab, C. and Gurevych, I. (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.

Stab, C. and Gurevych, I. (2017). Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Tayyar Madabushi, H., Kochkina, E., and Castelle, M. (2019). Cost-sensitive BERT for generalisable sentence classification on imbalanced data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China. Association for Computational Linguistics.

Tran, K., Bisazza, A., and Monz, C. (2018). The importance of being recurrent for modeling hierarchical structure. In *Proceedings of the 2018 Conference on*

Empirical Methods in Natural Language Processing, pages 4731–4736, Brussels, Belgium. Association for Computational Linguistics.

van Eemeren, F. H., Grootendorst, R., and Kruiger, T. (2019). *Handbook of Argumentation Theory*. De Gruyter Mouton.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.