End-to-end Argument Mining with Cross-corpora Multi-task Learning

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, and Kohsuke Yanai

Research and Development Group
Hitachi, Ltd.
Kokubunji, Tokyo, Japan
{gaku.morio.vn,hiroaki.ozaki.yu,
terufumi.morishita.wp,kohsuke.yanai.cs}@hitachi.com

Abstract

Mining an argument structure from text is an important step for tasks such as argument search and summarization. While studies on argument(ation) mining have proposed promising neural network models, they usually suffer from a shortage of training data. To address this issue, we expand the training data with various auxiliary argument mining corpora and propose an end-to-end cross-corpus training method called **Multi-Task Argument Mining** (**MT-AM**). To evaluate our approach, we conducted experiments for the main argument mining tasks on several well-established argument mining corpora. The results demonstrate that MT-AM generally outperformed the models trained on a single corpus. Also, the smaller the target corpus was, the better the MT-AM performed. Our extensive analyses suggest that the improvement of MT-AM depends on several factors of transferability among auxiliary and target corpora.

1 Introduction

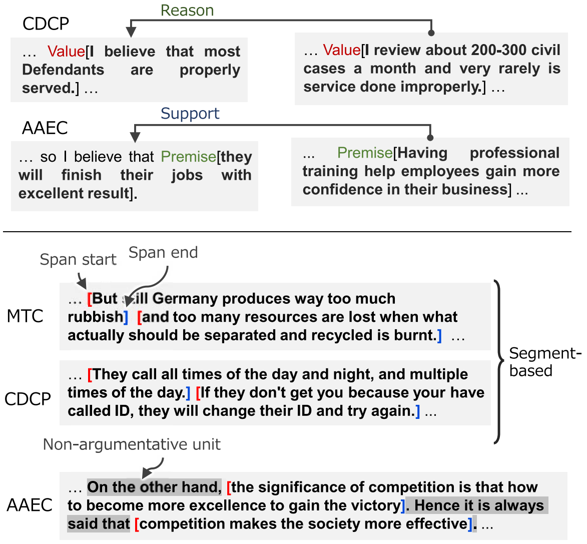
Argument(ation) mining (AM), the task of identifying an argument structure from text, has been gaining attention in recent years (Stede and Schneider, 2018; Lawrence and Reed, 2019). Also known as argument(ation) structure parsing (Kuribayashi et al., 2019), AM typically identifies argumentative component spans, classifies the type of the components, and classifies the relations between the components. In span identification, we discriminate argumentative text units (i.e., spans) from non-argumentative ones in a given text. In component classification, we classify the spans into argumentative labels such as Claim and Premise. Relation classification detects argumentative links between the components and classifies each link into a relation label such as Support and Attack.

Researchers have been utilizing various datasets (AM corpora) to develop AM models. These corpora are designed on the basis of different theoretical frameworks and conceptualizations (Daxenberger et al., 2017). For example, Peldszus and Stede (2016) developed a corpus called the Microtext Corpus (MTC) on the basis of Freeman's theory of the macro-structure of arguments (Freeman, 2011) to capture hypothetical dialectical exchanges. Stab and Gurevych (2017) created the Argument-annotated Essays Corpus (AAEC), in which a connected tree is used to represent the structure of each argument in a paragraph. A notable feature of these AM corpora is that, due to the lack of a clear underlying framework on which to base the annotation, new annotated corpora are likely to diverge in terms of the span, type of components, and type of relations.

In addition, developing an AM corpus is expensive because of the annotation cost (Schulz et al., 2018). For example, Lauscher et al. (2018) conducted multiple calibration phases when training annotators to improve the low inter-annotator agreement, which drove up costs. This suggests that it would be infeasible to create a sufficiently large AM corpora for a new domain, and the resultant lack of data inevitably degrades the performance of argument structure parsing.

Given the lack of a large annotated AM corpus, multi-task learning on AM corpora labeled with various annotation schemes may be a promising approach for improving parsing performance. Few studies have investigated this except Schulz et al. (2018) and Putra et al. (2021a,b). Schulz et al. (2018) showed the effectiveness of multi-task learning for component classifications under low-resource settings, while Putra et al. (2021a) showed how argumentative link prediction can be improved by multi-corpora training.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 639–658, 2022. https://doi.org/10.1162/tacl_a_00481
Action Editor: Vincent Ng. Submission batch: 9/2021; Revision batch: 1/2022; Published 5/2022.



… Value[I believe that most Defendants are properly served.] …

… Value[I review about 200-300 civil cases a month and very rarely is service done improperly.] …

… so I believe that Premise[they will finish their jobs with excellent result].

… Premise[Having professional training help employees gain more confidence in their business] …

… [But still Germany produces way too much rubbish] [and too many resources are lost when what actually should be separated and recycled is burnt.] …

… [They call all times of the day and night, and multiple times of the day.] [If they don't get you because your have called ID, they will change their ID and try again.] …

… On the other hand, [the significance of competition is that how to become more excellence to gain the victory]. Hence it is always said that [competition makes the society more effective]. …

Figure 1: Potentially transferable properties between AM corpora. **Top**: Two corpora have a similar relation label. **Bottom**: MTC and CDCP both use consecutive segment-based spans.

However, to the best of our knowledge, no prior studies have conducted extensive analysis for all AM tasks (i.e., span, component, and relation tasks) with an end-to-end model because it is challenging to take various differently annotated corpora into account for the modeling.

Through our observations, we hypothesize that parsing performance by multi-task learning can be improved by the potential transferability between AM corpora. Figure 1 (top) shows two different AM corpora: Cornell eRulemaking Corpus (CDCP; Park and Cardie [2018]) and AAEC. The relations of CDCP (i.e., Reason) and AAEC (i.e., Support) have different label names but share the concept of support connectivity, where both argumentative component pairs have a similar implicit relation. In Figure 1 (bottom), CDCP and MTC use a similar type of span. Although it is not a common definition, we refer to the span type as segment-based span, where the text is segmented into consecutive elementary discourse units or elementary units in which most of the units are clauses or sentences (Peldszus and Stede, 2016), and the text does not contain or rarely contains non-argumentative component parts. The potential ability of these transferable properties is referred to as *transferability* in this study.

In this paper, we propose **Multi-Task Argument Mining (MT-AM)**, an end-to-end cross-corpus training model for AM, to address the shortage of training corpora. We first describe single-corpus learning, namely, a single-task (ST) model, and then detail the extension of the model (MT-AM) for multi-task learning on various corpora. For the ST model, we provide a simple but flexible architecture called the span-biaffine architecture that can handle different concepts of spans, types of component labels, and graph structures. MT-AM involves two training stages: *multi-task pre-training* to capture transferable properties between auxiliary corpora and a target corpus, and *target corpus fine-tuning* to adjust the parameters for each target corpus.

Experiments using five AM corpora showed that MT-AM performed generally better than the ST model. More importantly, the smaller the target corpus was, the better MT-AM performed, suggesting the effectiveness of MT-AM for remedying the lack of training data, which is likely in line with the previous work (Schulz et al., 2018). To investigate transferability among the corpora, we also examined all the corpus pairs of an auxiliary corpus and target corpus and found that the choice of auxiliary corpus affects the parsing performance (Figure 5 and Table 6).

Finally, we further discuss transferability under low-resource settings and advocate three types of hypothetical transferability in AM on the basis of the following observations. (i) *Data/training sufficiency*: With a small number of data samples or training steps for the target corpus, MT-AM detected larger number of components or relations than an ST model (Figure 7). (ii) *Annotation compatibility*: Partial compatibility of the annotation design between the auxiliary corpora and target corpus helped MT-AM improve the parsing performance (Figures 8 and 9, and Table 7). (iii) *Semantic compatibility*: Argumentative knowledge of an auxiliary corpus that is semantically compatible with the target corpus could be transferred using MT-AM (Figure 10 and Table 8). These insights will be useful for designing an effective cross-corpus AM method. We released our code at https://github.com/hitachi-nlp/graph_parser. All corpora used in this paper are publicly available from each distributor.

2 Background

2.1 Evolution of AM

Argument(ation) is an activity that involves reasoning, in which an arguer attempts to rationally justify his or her points on a certain topic (Eemeren et al., 1996). In natural language processing, studies on identifying or classifying arguments from text computationally have been underway for the past several years (Eckle-Kohler et al., 2015; Rinott et al., 2015; Park et al., 2015b; Habernal and Gurevych, 2017; Trautmann et al., 2020; Trautmann, 2020; Boltužić and Šnajder, 2020).

The main objective of AM is to predict the argument structure from an unstructured text (Peldszus, 2014; Peldszus and Stede, 2015; Lawrence and Reed, 2019; Kuribayashi et al., 2019). Stab and Gurevych (2017) defined AM as a pipeline consisting of three tasks: component identification to predict an argument component span, component classification to predict an argument component type such as Claim or Premise, and structure identification to predict argumentative relations between the components. Discourse parsing is sometimes compared with AM. For example, Rhetorical Structure Theory (RST; Carlson et al., 2001) parsing postulates a hierarchical discourse structure (Ji and Eisenstein, 2014), and the model is strictly constrained by the tree-based rules. In contrast, AM handles a variety of corpora with different conceptualizations.

AM as Relation Extraction: Many methods used in AM are derived from dependency parsing or relation-extraction methods. While syntactic dependency parsing converts a sentence into word token-to-token relations, AM converts a text into component-to-component relations. A widely known method of parsing component-to-component relations is to use pairwise classification. One of the first instances of this was a feature-based relation extraction developed by Stab and Gurevych (2014, 2017), who parsed AAEC by pairwise classification. Persing and Ng (2016) also applied a relation classifier for each component pair. Our study follows the pairwise classification for the relation classification as well, and we jointly optimize the span identification and component classification.

Due to the rapid advancement in neural architectures, AM has been using representations contextualized by neural networks. For example, Eger et al. (2017) used LSTM-ER (Miwa and Bansal, 2016), which was originally utilized to parse relations between entities, and a bidirectional LSTM (BiLSTM; Graves and Schmidhuber, 2005)-based model (Søggard and Goldberg, 2016) to parse the argument structure on AAEC. Kuribayashi et al. (2019) investigated a span representation with BiLSTM that takes into account argumentative markers. Potash et al. (2017) used an encoder-decoder approach with a pointer network (Vinyals et al., 2015) to predict relations between argument components. Niculae et al. (2017) proposed a neural model in which a factor graph formulation is provided. Galassi et al. (2018) proposed a neural architecture for parsing argument graphs. The model architecture that we propose in this study is based on biaffine operation (Dozat and Manning, 2017), which was used in the work by Morio et al. (2020).

The rapid advances in pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2019 have affected AM studies (Kuribayashi et al., 2019; Mayer et al., 2020). Mayer et al. (2020) presented a BERT model with a component pair classifier for predicting relations. Wang et al. (2020) also used BERT in encoders. In contrast to the above studies, since a given input argument can have many sentences, we use Longformer (Beltagy et al., 2020) to encode text, as it can handle a longer input sequence than BERT.

Low-resource Issues: Because AM corpora tend to be small, the training strategy needs to be refined to improve parsing under low-resource settings. Schulz et al. (2018) reported that multi-task learning for component identification is effective, particularly in data-sparsity settings. Accuosto and Saggion (2019) proposed a transfer-learning method that uses contextualized representations learned from discourse parsing tasks. Chakrabarty et al. (2019) used Universal Language Model Fine-tuning (ULMFiT; Howard and Ruder, 2018) on 5.5 million opinionated claims in a Reddit corpus for claim detection. In contrast, we use multiple AM corpora to predict argument structure. In discourse parsing, Guz et al. (2020) used a pre-trained language model and training with silver-standard data for parsing, and Huber and Carenini (2019) provided a distant supervision method on an auxiliary sentiment-classification task. Recently, Putra et al. (2021a) reported that using multi-corpora learning with selective sampling improves the link prediction performance. We also study the multi-corpora learning method, further analyzing the method using an end-to-end model.

Corpus	Component	Relation (Blue: support, *Red*: attack)	Segment-based spans
AAEC	MajorClaim (751), Claim:For (1228), Claim:Against (278), Premise(3832)	Support (3613), *Attack* (219)	
MTC	Opp (125), Pro (451)	Sup (281), *Und* (64), *Reb* (110), Exa (9)	✓
CDCP	Value (2160), Fact (746), Policy (815), Testimony (1026), Reference (32)	Reason (1307), Evidence (46)	✓
AbstRCT	MajorClaim (93), Evidence (2193), Claim (993)	Support (1762), *Partial-Attack* (238), *Attack* (60)	
AASD	Proposal (110), Means (63), Result (74), Observation (11), Assertion (88), Description (7)	Detail (129), Support (126), Sequence (11), Additional (27), *Attack* (0)	✓

Table 1: Label distribution for each corpus. We show the number of component labels, relation labels, and the type of span.

Models for Various AM Corpora: A few studies, such as those by Cocarascu et al. (2020) and Galassi et al. (2021), investigated models that can be used for various AM corpora. Wachsmuth et al. (2017) provided a unified view for three AM corpora to analyze the patterns in their overall argumentation. In parallel with our study, Bao et al. (2021) proposed a transition parser for both tree and non-tree arguments. Compared with that study, we include end-to-end learning (i.e., from span identification to relation classification) and do not require any transition designs.

3 Overview of AM Corpora

We focus on five differently annotated corpora: AAEC, a medical abstract corpus named AbstRCT (Mayer et al., 2020),[1] CDCP, MTC, and the argument-annotated SciDTB (AASD; Accuosto and Saggion, 2020). These corpora are useful for discussing differences and similarities in argument structure. Table 1 shows the label distribution for these five corpora.[2] As can be seen, MTC and AASD are relatively low-resource corpora in terms of both components and relation labels. The details of each corpus are as follows.

AAEC (Stab and Gurevych, 2017) contains student essays. There are two types of data: essay-level and paragraph-level (Eger et al., 2017). A stance for a controversial theme is expressed by a MajorClaim component as well as Claim components, and Premise components justify or refute the Claims. Attack and Support labels are defined as relations. The span covers a statement, *which can stand in isolation as a complete sentence*, according to the AAEC annotation guidelines (Stab and Gurevych, 2017). All components are annotated with minimum boundaries of a clause or sentence excluding so-called ‘‘shell’’ language such as *On the other hand* and *Hence it is always said that*, as seen in Figure 1 (bottom). Thus, this corpus discriminates argumentative text units from non-argumentative ones, producing many non-argumentative component parts. This is different from the segment-based spans that do not contain or barely contain non-argumentative component parts.

MTC (Peldszus and Stede, 2016) is based on Freeman’s theory of the macro-structure of arguments (Freeman, 2011) and incorporates the ideas of Toulmin (Toulmin, 2003) into diagramming techniques. MTC introduces dialectical exchange between Pro (proponent) and Opp (opponent) components. Relations include Add, Exa (example), Reb (rebut), Sup (support), and Und (undercut). According to precedent studies, we pre-processed the Add relations similarly to Kuribayashi et al. (2019). MTC introduces a segment-based span where a segment usually contains a clause or sentence.

CDCP (Park and Cardie, 2018) consists of comments in which five types of components (Fact, Testimony, Reference, Value, and Policy) and two types of supporting relations (Reason and Evidence) are annotated on the basis of the study by Park et al. (2015a). The spans are segmented into elementary units with a proposition consisting of a sentence or a clause. CDCP also discriminates argumentative text units from non-argumentative ones, but we classify the span type of CDCP as segment-based because very few non-argumentative units exist in the corpus. We pre-processed this corpus using a similar approach

[1] AbstRCT: Abstracts from Randomized Controlled Trials.

[2] The CDCP statistics differ from those in Galassi et al. (2018), probably due to the preprocessing difference of a link transitive.

	AAEC	MTC	CDCP	AbstRCT	AASD
type	tree	tree	graph	graph	tree
# texts	**1833**	112	731	500	60
# components	**6089**	576	4779	3279	353
# relations	**3832**	464	1353	2060	293
% trees	73.27	**100**	7.8	31.40	**100**
% reentrant	0	0	**4.29**	1.06	0
relation density	0.812	**1**	0.391	0.767	**1**
% noncrossing	**99.78**	94.64	95.08	69.80	95.00

Table 2: Statistics of AM corpora (maximum values are shown in **bold**). For AAEC, paragraph-level statistics are shown.

to Morio et al. (2020), where continuous spans are merged and transitive closures are processed.

AbstRCT (Mayer et al., 2020) includes abstracts of randomized controlled trials (RCTs) from the MEDLINE database.[3] MajorClaim, Claim, and Evidence components, along with Support, Attack, and Partial-attack relations, are annotated for various diseases (e.g., neoplasm, glaucoma, hepatitis, diabetes, and hypertension). Similar to AAEC, this corpus discriminates argumentative text units from non-argumentative ones, producing non-argumentative component parts. We used 350 training, 50 development, and 100 test neoplasm texts.

AASD (Accuosto and Saggion, 2020) was created to address the lack of a scientific AM corpus. The authors enriched annotations for a subset of SciDTB (Yang and Li, 2018) by providing component types for Proposal, Assertion, Result, Observation, Means, and Description. Support and Attack relations are provided on the basis of the study by Kirschner et al. (2015). AASD also includes Detail, Additional, and Sequence relations. The spans are segmented into elementary discourse units, similar to MTC.

3.1 Discussion of the Five Corpora

Although the five corpora have different designs, they also share similarities to some extent. For example, the early datasets such as AAEC and MTC led to the development of corpora such as AbstRCT and AASD. As for structural designs, AAEC and MTC form a **tree** argument, and AASD is also tree-based. Table 2 summarizes the statistics of the AM corpora.[4] The relation density indicates an averaged value of $N_r/(N_c - 1)$, where N_r and N_c represent the number of relations in a graph and the number of components, respectively. MTC and AASD are always composed of 100% tree arguments, so the relation density is 1. Unlike these corpora, the argument structure of CDCP does not necessarily form a tree (i.e., a **graph**). For example, given two supporting relations $(a \rightarrow b)$ and $(b \rightarrow c)$, a transitive relation $(a \rightarrow c)$ is established. This is a case of **reentrancy** (Vilares and Gómez-Rodríguez, 2018). The relation density of CDCP shows that a larger number of components are isolated. We can also observe differences in **crossing** relation, which is known as projectivity in the context of dependency parsing (Covington, 2001; Oepen et al., 2019). AAEC is mostly non-crossing, while the structure of AbstRCT is more complex.

For the relation label design, AM corpora usually contain support or attack types. For example, (AAEC; Premise – Support → Claim) has a similar support type to (CDCP; Fact – Reason → Value), and (MTC; Opp – Reb → Pro) has a similar attack type to (AAEC; Premise – Attack → Premise). We indicate the connection types with different colored text in Table 1, which shows that the support type appears across all corpora.

Component spans are usually designed to capture their meaning within a minimum boundary. As shown in Table 1, MTC, CDCP, and AASD have segment-based spans in which a text is generally split into sentences or clauses. In contrast, AAEC and AbstRCT contain non-argumentative parts in text.

Certain similarities and differences can be expected in component types. While AAEC and AbstRCT share the Claim and MajorClaim components, CDCP and AASD provide corpus-specific types such as Testimony and Proposal.

4 Models

4.1 Task Formalization

Similar to the study by Stab and Gurevych (2017), we introduce three tasks: (i) **Span identification**, which predicts the component span, (ii) **Component classification**, a successive task of span identification for classifying the component label associated with a span, and (iii)

[3] https://www.nlm.nih.gov/medline/medline_overview.html.

[4] We used the mtool library (Oepen et al., 2019, 2020) to compute the statistics.

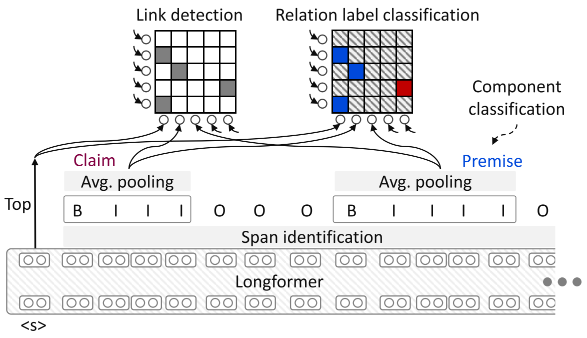


Figure 2: Overview of ST model with four output layers (span, component, link, and relation) on Longformer.

Relation classification, a successive task of the span identification for detecting and classifying argumentative relations between the spans.

Formally, we predict a set of spans (S), components (C), and their relations (R). A span can be represented as a pair $\langle s, e\rangle \in S$, where s and e are the span start and end character indices, respectively. The component can be represented as a triple $\langle s, e, c\rangle \in C$, where c denotes the type of component associated with $\langle s, e\rangle$. The relation can be represented as $\langle s_{\text{src}}, e_{\text{src}}, s_{\text{tgt}}, e_{\text{tgt}}, r\rangle \in R$, where s_{src} and e_{src} represent the source-side span, s_{tgt} and e_{tgt} represent the target-side span, and r indicates the relation label.

4.2 ST Model

Before describing MT-AM, we present an ST model (Figure 2), which is the basic architecture of MT-AM. As discussed in Section 3.1, we need an architecture that does not depend on a specific span unit or type of graph structure. To this end, we provide a simple end-to-end architecture, the concept of which is similar to that in previous studies (Eberts and Ulges, 2020; Jiang et al., 2020).

Longformer: We use Longformer-base (Beltagy et al., 2020), a pre-trained language model that can handle a long input sequence. To reduce the computational complexity ($O(n^2)$ in a self-attention), Longformer takes both local and global contexts into account. The global attention focuses on inductive bias on a specific task (Beltagy et al., 2020). The local attention is computed using a windowed self-attention that reduces the computation time to $O(nw)$, where w is the window size.

hyperparameter	AAEC	CDCP	AbstRCT
batch size		4	
MLP dropout, dim		0.1, 768	
λ_{c}	0.18	0.057	0.035
λ_{ℓ}	1.05	0.82	0.58
λ_{r}	0.21	0.15	0.17
LR	9.1e-5	5.6e-5	8.1e-5
LR (multi-task pre-training)	1.7e-5	2.5e-5	1.9e-5
epochs (single-task training)		20	
epochs (multi-task pre-training)	2	4	4
epochs (target corpus fine-tuning)	18	16	16
auxiliary weight	0.24	0.66	0.76
Adam beta1, beta2		0.9, 0.998	

Table 3: Rounded hyperparameter values.

We tokenize the input text using Longformer's tokenizer, inserting special tokens such as the beginning and ending tokens (i.e., <s> and </s>). We apply the global attention for the first token <s> to account for the entire argument.

Span-biaffine Architecture: As shown in Figure 2, we developed this architecture by providing task-specific classifiers on the Longformer.

First, a span classifier produces BIO tags by means of a multi-layer perceptron (MLP) on top of Longformer outputs. We obtain span representation with average pooling. Gold spans are used in training, and only predicted spans are used in inference.

The span representation is then used to classify the component type. For a span representation of $\langle s, e\rangle$, we apply an MLP to obtain a probability distribution for the component label c.

The span representations are also used to compute relations. Given the different costs required to detect and classify relations (as shown in Table 3), we use two biaffine operations (Dozat and Manning, 2017, 2018), similar to the study by Morio et al. (2020). A link detection biaffine classifier predicts if a relation between a source span $\langle s_{\text{src}}, e_{\text{src}}\rangle$ and target span $\langle s_{\text{tgt}}, e_{\text{tgt}}\rangle$ exists. A relation label biaffine classifier predicts the label r associated with the relation. We use MLPs with the biaffine classifier as follows:

$$\mathbf{h}_i^{(\text{src})} = \textsc{Mlp}^{(\text{src})}\left(\mathbf{e}_i\right), \; \mathbf{h}_j^{(\text{tgt})} = \textsc{Mlp}^{(\text{tgt})}\left(\mathbf{e}_j\right),$$

$$\mathrm{P}\left(y_{i\to j}\right) = f\left(\textsc{Biaffine}\left(\mathbf{h}_j^{(\text{tgt})}, \mathbf{h}_i^{(\text{src})}\right)\right),$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ denote the span representations of the i-th and j-th components, and BIAFFINE represents the biaffine (or possibly bilinear) operation. For link detection, f is a sigmoid function, so $\mathrm{P}\left(y_{i \to j}\right)$ is a probability value. For relation label classification, f is a softmax function, so $\mathrm{P}\left(y_{i \to j}\right)$ is a probability distribution of labels. By combining both outputs, we obtain the relation $\langle s_{\mathrm{src}}, e_{\mathrm{src}}, s_{\mathrm{tgt}}, e_{\mathrm{tgt}}, r \rangle$. Note that relation labels are only backpropagated on the gold links (Dozat and Manning, 2018).

We consider an imaginary top for the first token `<s>`. The top is linked to all components that have no outgoing relations. This enables us to use an optimization algorithm to make the graph into a tree.

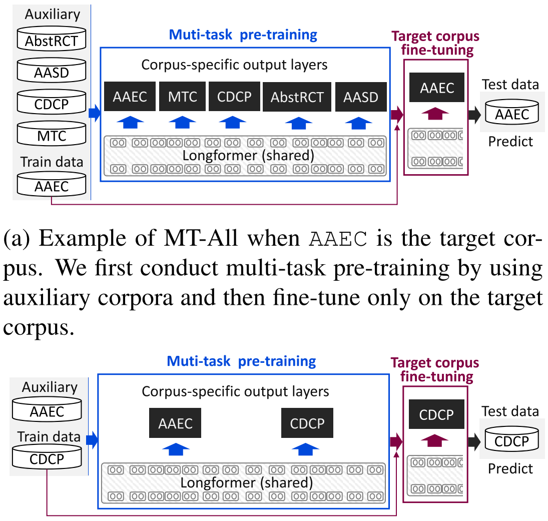
Let the cross-entropy loss of the span classifier be $\mathcal{L}_{\mathrm{s}}$, cross-entropy loss for the component classification be $\mathcal{L}_{\mathrm{c}}$, binary cross-entropy loss for the link detection be $\mathcal{L}_{\ell}$, and cross-entropy loss for the relation label classifier be $\mathcal{L}_{\mathrm{r}}$. The objective to be optimized is $\mathcal{L} = \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}} + \lambda_{\mathrm{c}}\mathcal{L}_{\mathrm{c}} + \lambda_{\ell}\mathcal{L}_{\ell} + \lambda_{\mathrm{r}}\mathcal{L}_{\mathrm{r}}$, where λ are hyperparameters.

4.3 MT-AM

We propose MT-AM, an end-to-end model using a two-staged method (multi-task pre-training and target corpus fine-tuning), which is similar to the work of Liu et al. (2019a). We modify the model architecture of ST for the multi-task pre-training because the output labels are different for each corpus, and corpus-specific output layers are implemented while still sharing Longformer parameters. In the target corpus fine-tuning, we further train the model on only the target corpus.

Generally speaking, the objective can be the sum of losses for each corpus in the multi-task pre-training, and MT-AM should capture transferable features across different corpora. However, each auxiliary corpus would still have distant information, which degrades the parsing performance for a target corpus. We therefore provide a smaller loss weight (van der Goot et al., 2021) (i.e., auxiliary weight) for the auxiliary corpora. We also provide a different learning rate (LR) and epochs for the multi-task pre-training and target corpus fine-tuning. These hyperparameters are described later.

We provide the following variants of MT-AM: **MT-All**, which conducts multi-task pre-training with all five corpora. An example of MT-All for `AAEC` is shown in Figure 3a, where the five corpora provide corpus-specific output layers in the multi-task pre-training. In this case, the auxiliary corpora are `MTC`, `CDCP`, `AbstRCT`, and `AASD`. After a couple of epochs in this stage, we further fine-tune on the target corpus (i.e., `AAEC`). Finally, we evaluate the test data of the target corpus. **Corpus$_x$ → Corpus$_y$**, which uses a single auxiliary corpus (i.e., Corpus$_x$) only for analysis (see Sections 5.3 and 5.4). We use Corpus$_x$ and the training data of Corpus$_y$ in the multi-task pre-training. After a couple of epochs in this stage, we further fine-tune on the target corpus (i.e., Corpus$_y$). An example of `AAEC` → `CDCP` is shown in Figure 3b.



(a) Example of MT-All when `AAEC` is the target corpus. We first conduct multi-task pre-training by using auxiliary corpora and then fine-tune only on the target corpus.

(b) Example of Corpus$_x$ → Corpus$_y$. `AAEC` is the auxiliary corpus (Corpus$_x$) and `CDCP` is the target corpus (Corpus$_y$).

Figure 3: Overview of MT-AM.

5 Experiments

We compared the MT-AM and ST models (Section 5.2), investigated the transferability (Section 5.3), and conducted an in-depth analyses under low-resource settings (Section 5.4).

5.1 Experimental Setup

Hyperparameters: We apply dropout (Srivastava et al., 2014) to the MLP layers. We use the Adam (Kingma and Ba, 2015) optimizer with a linear warmup scheduler (Howard and Ruder, 2018). We tune the LR and λ values as well as the number of epochs in the multi-task pre-training and the auxiliary weight by Optuna (Akiba et al., 2019), a hyperparameter optimization framework.

Five-fold cross-validation (CV) is applied to the training data for the hyperparameter optimization. MTC and AASD use CV in the experiments, so instead of tuning the hyperparameters we use the same hyperparameters as AAEC. To stabilize the tuning, we fix the total number of training epochs to 20, set $\lambda_s = 1$, and tune the other λ values by sampling from a uniform distribution within the range of $[0.01, 2.0]$. We first tune the hyperparameters related to the ST model and then those related to the multi-task pre-training (e.g., LR, auxiliary weight, and number of epochs). The number of epochs in the multi-task pre-training is tuned by sampling from 2, 4, 6, and 8 while keeping the total number of epochs to 20.

The tuned and fixed hyperparameters are shown in Table 3. We found that the LR in the multi-task pre-training was generally lower than that of the target corpus fine-tuning, suggesting that multi-task pre-training degrades the target corpus representation if we use a higher LR.

Implementation Details: After applying the relation classifiers, we use Chu–Liu/Edmonds' algorithm (Chu and Liu, 1965) for AAEC, MTC, AbstRCT,[5] and AASD to make the predicted links (i.e., the score matrix produced by the link detection) into a tree. We use given training and test data for AAEC, CDCP, and AbstRCT, examine 30 different seeds, report the average score, and use ten sets of five-fold CV (Kuribayashi et al., 2019) for MTC and AASD. Unless otherwise specified, we use the essay-level data for AAEC. The development data are randomly sampled from the training data except for AbstRCT. We use given development data for AbstRCT. We select the optimal model on the basis of the development score by evaluating once every two epochs.

Evaluation Metrics: Let $\mathcal{G}_{\text{task}}$ and $\mathcal{S}_{\text{task}}$ be a set of gold and system outputs for a task, respectively. Precision $P = |\mathcal{G}_{\text{task}} \cap \mathcal{S}_{\text{task}}|/|\mathcal{S}_{\text{task}}|$, recall $R = |\mathcal{G}_{\text{task}} \cap \mathcal{S}_{\text{task}}|/|\mathcal{G}_{\text{task}}|$, and $F = 2PR/(P + R)$ are then defined. For example, $\langle id, s, e \rangle \in \mathcal{G}_{\text{span}}$, where id indicates an ID inherent to a text. Similarly, $\langle id, s, e, c \rangle \in \mathcal{G}_{\text{component}}$, and $\langle id, s_{\text{src}}, e_{\text{src}}, s_{\text{tgt}}, e_{\text{tgt}}, r \rangle \in \mathcal{G}_{\text{relation}}$. We also introduce the *link* score (Kuribayashi et al., 2019), which is used to measure F-scores for determining the existence of relations regardless of their labels.

Corpus	Model	Span	Component		Link	Relation	
			F	Macro		F	Macro
AAEC	ST	85.21	75.54	66.59	55.66	55.17	42.30
	MT-All	85.20	**75.66**	**67.03**	**55.72**	**55.17**	41.92
	ST	OS	87.44	79.55	67.16	66.29	55.95
	MT-All	OS	**87.68**	**80.37***	**67.88***	**66.91***	55.84
MTC	ST	87.68	78.83	73.77	53.43	45.92	33.07
	MT-All	**88.11**	**80.98***	**77.59***	**57.73***	**51.25***	**39.65***
	ST	OS	89.85	83.70	65.06	55.23	39.27
	MT-All	OS	**91.58***	**87.22***	**68.55***	**60.45***	**47.11***
CDCP	ST	82.88	68.90	65.78	31.94	31.94	16.26
	MT-All	**83.02**	68.81	64.24	**33.76***	**33.74***	**17.18***
	ST	OS	81.03	82.34	40.15	40.11	20.39
	MT-All	OS	80.84	80.88	**42.15***	**41.72***	**21.21***
AbstRCT	ST	70.29	64.16	45.04	39.35	38.38	31.91
	MT-All	**70.93**	**64.78**	44.56	**39.74**	**38.71**	**33.94***
	ST	OS	89.37	67.57	59.65	57.10	47.12
	MT-All	OS	89.23	67.14	**60.66***	**58.05***	**50.75***
AASD	ST	87.10	69.06	58.06	54.82	49.83	42.10
	MT-All	86.70	**72.88***	**63.57***	**58.51**	**53.89***	**47.64***
	ST	OS	77.78	65.90	63.96	58.30	49.22
	MT-All	OS	**81.93***	**71.45***	**68.77***	**63.40***	**54.95***

Table 4: F-scores [%]. MT-All (shown in **bold**) generally outperformed the ST models. ''OS'' denotes oracle span setting. ''Macro'' represents macro-averaged F-score. * shows statistical significance $p < 0.05$.

Note that the scores of component and relation classifications are affected by the performance of the span identification task.

5.2 Comparison of MT-All with ST Model

5.2.1 Effectiveness of MT-AM

To determine the effectiveness of MT-All, we compared it against the ST model. Table 4 shows the overall results on the five AM corpora. We also show oracle span (OS) results, where the components and relations are predicted on gold spans, to determine the improvement for each component and relation classification. As we can see in the table, in most corpora, MT-All outperformed the ST model on average. The largest improvement by MT-All was observed for MTC and AASD in the component and relation classification. This would be because these two corpora are smaller than the others and MT-All benefited more from the auxiliary corpora. However, the ST model performed better than MT-All for some tasks in AbstRCT and CDCP. In CDCP, labels that do not have corresponding labels in other corpora, such as Policy and Testimony (see Table 1), would be more challenging to improve by multi-task learning.

[5] This is done since most of the AbstRCT components have only one outgoing relation (Mayer et al., 2020).

5.2.2 MT-All under a Low-resource Setting

The above results indicate that the parsing performance of MT-AM improves for low-resource corpora. Because AM corpora in real settings are generally low-resource, we further investigated the effectiveness of MT-AM under the following three extremely low-resource settings inspired by Schulz et al. (2018):

1. Sample n [%] texts of the training and development data in a target corpus.
2. For MT-AM, conduct multi-task pre-training with auxiliary corpora and the sampled data of the target corpus. For the target-corpus fine-tuning, use the sampled data. For ST, conduct fine-tuning on the sampled data.
3. Evaluate test data in the target corpus.

Similar to the study by Liu et al. (2019a), we changed the sample amount n to 1%, 10%, or 100%. When $n = 1\%$, the amount of sampled training data of `AbstRCT` was 3 (texts) and that of `MTC` and `AASD` was 1.

To investigate the improvement of MT-AM for an ST model, we used the following metric (larger is better), similar to the study by Morishita et al. (2020):

$$\text{Error reduction } [\%] = \epsilon(\text{ST}) - \epsilon(\text{MT-AM}),$$

where ϵ is a function that computes the error rate, that is, $100 - $ F-score [%]. However, it is more robust to use a degree of error reduction for errors produced in an ST model as a metric. For example, error reduction by MT-AM (95 F-score) for an ST model (90 F-score) is the same as that by MT-AM (55 F-score) for an ST model (50 F-score). Intuitively, improving the F-score from 90 to 95 is more challenging than improving it from 50 to 55. Thus, we define the error reduction rate (ERR; larger is better) as

$$\text{ERR } [\%] = 100 \times \frac{\text{Error reduction}}{\epsilon(\text{ST})}.$$

Following the above setting, Figure 4 represents the ERR by MT-All for an ST model with three different training data amounts.[6] Overall, we found that when there was less training data in the target corpus, the ERR was larger. This suggests that MT-AM is especially effective in the low-resource setting, which is likely in line with the previous work (Schulz et al., 2018). On the other hand, when $n = 100\%$, ERR for relation and component tasks was larger than that for the span identification task. However, when $n = 1\%$ or 10%, the ERR for the span identification task was larger than that for the other tasks. Given that the span identification task is less semantic[7] than component and relation classifications, we conclude that the ERR of spans can be improved with less training data. In other words, it is difficult for the span identification task to benefit from multi-task learning when the amount of training data is sufficient. In contrast, relation and component tasks are more challenging to train; therefore, the two tasks require more training data.

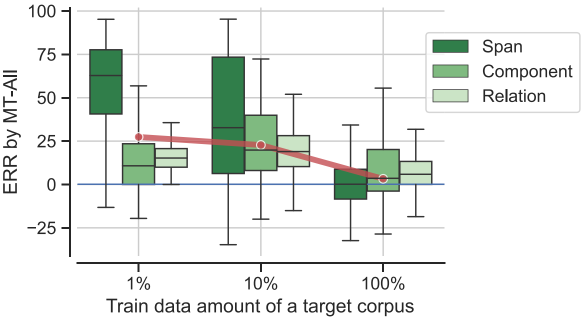


Figure 4: ERR (with OS for components and relations) by MT-All for ST with different training data amounts (n%). We averaged the results of five corpora, removing outliers for visibility. Red line shows average values over three tasks.

5.2.3 Evaluation of the ST Model

To investigate whether our ST model (and hence the MT-AM model) performs reasonably, we compared it with the following state-of-the-art end-to-end `AAEC` parsers:

- **BLCC** (Eger et al., 2017), which parses the argument structure as a sequence tagging task.
- **LSTM-ER** (Eger et al., 2017), which uses LSTM-ER (Miwa and Bansal, 2016) (originally used to parse relations of entities).
- **BiPAM-syn** (Ye and Teufel, 2021), a state-of-the-art parser based on a biaffine operation and syntactic information.

In addition to the essay-level results, we provide paragraph-level results. We also provide

[6] We compute ERRs for each CV or seed.

[7] For example, some corpora annotate spans based on (semi-)automatically split segments for ease of annotation.

Corpus	Model	Span	Component F	Component Macro	Link	Relation F	Relation Macro
AAEC	BLCC	–	63.23			34.82	
	LSTM-ER	–	66.21			29.56	
	ours*ST	–	**76.55**			**54.66**	
AAEC (para-graph-lev.)	BLCC	–	66.69			39.83	
	LSTM-ER	–	70.83			45.52	
	BiPAM-syn	–	73.5			46.4	
	ours*ST	–	**76.48**			**59.55**	
	ILP Joint	OS		82.6	58.5		
	Ptr. Net.	OS		84.9	60.8		
	Span Repr.	OS		85.7	67.8		
	BERT Trans.	OS		**88.4**	**70.6**		
	ours†ST	OS	88.40	86.82	69.33	68.14	57.11
MTC	ILP Joint	OS		85.7	48.6		
	Ptr. Net.	OS		81.3	57.7		
	Span Repr.	OS		83.5	57.5		
	ours†ST	OS	95.65	**93.08**	**65.06**	57.89	57.34
CDCP	TSP-PLBA	OS		78.91	34.04		
	BERT Trans.	OS		**82.5**	37.3		
	ours ST	OS	81.03	82.34	**40.15**	40.11	20.39
Abst-RCT	Rel.RoBERTa	OS				48.72	17.53
	Rel.SciBERT	OS				**58.21**	36.76
	ours ST	OS	89.37	67.57	59.65	57.10	**47.12**

Table 5: Comparison of our ST model and other models. * is computed by C-F1 (100%) and R-F1 (100%) introduced by Eger et al. (2017). † calculates scores on the basis of label sets introduced by Kuribayashi et al. (2019) for comparison, where we map AAEC's Claim:Against and Claim:For labels into Claim. We also map an MTC's component labels to Claim or Premise and relation labels to Support or Attack.

the following state-of-the-art OS baselines for reference:

- **ILP Joint** (Stab and Gurevych, 2017), which uses integer linear programming (ILP) for parsing.
- **Ptr. Net.** (Potash et al., 2017), which uses a pointer network (Ptr. Net.) to predict relations.
- **Span Repr.** (Kuribayashi et al., 2019), which uses distinct encoders to represent argumentative markers.
- **BERT Trans.** (Bao et al., 2021), a transition-based parser with BERT.
- **TSP-PLBA** (Morio et al., 2020), a biaffine-based parser.
- **Rel.RoBERTa/SciBERT** (Mayer et al., 2020),[8] a Transformer (Vaswani et al., 2017)-based AM parser that uses RoBERTa (Liu et al., 2019b) or SciBERT (Beltagy et al., 2019).[9]

All the scores are drawn from the original papers of the above baselines except for Rel.RoBERTa and Rel.SciBERT.

Table 5 shows the F-scores, where we can see that in both essay- and paragraph-level AAEC, our ST model outperformed other end-to-end models (i.e., non-OS models). Compared with existing methods like BiPAM-syn, our model significantly improved the relation classification score. Although our model is not tailored to OS architectures or hyperparameters, when combined with OS it showed comparative or better parsing scores than the other models. Interestingly, BERT Trans. had better scores than our model in AAEC, while our model was better in link detection on CDCP. Since the architectures of our model and BERT Trans. are different (cf. graph-based vs. transition-based; Falenska et al., 2020), we conclude that incorporating the advantages of both methods may improve the parsing performance.

5.3 Discussion: Transferability among Corpora

To investigate transferability in MT-AM, we first examined the corpus-wise transferability by all pairs of Corpus$_x$ $\rightarrow$ Corpus$_y$ to determine the capabilities of each corpus. We then analyzed the label-wise transferability for relations.

Corpus Preference: Figure 5 shows a matrix plot created by computing the error reduction ($n =$ 100%) for all Corpus$_x$ $\rightarrow$ Corpus$_y$ pairs. The vertical axis represents the auxiliary corpus and the horizontal axis represents the target corpus. As shown in the figure, each corpus pair has a different corpus preference. The error reduction in each column varies in accordance with the Corpus$_x$ $\rightarrow$ Corpus$_y$ pairs. For example, when AAEC was an auxiliary corpus (i.e., AAEC row), the error reduction of the relation for AbstRCT was 0.1 while that for CDCP was 0.95.

From the ''avg'' columns in Figure 5, we can evaluate how each corpus works as an auxiliary corpus on average. As shown, the lower resource corpora, MTC and AASD, are less effective as an auxiliary corpus for MT-AM because

[8] https://gitlab.com/tomaye/ecai2020-transformer_based_am.

[9] We only use the relation classification model, distinguishing Attack and Partial-attack labels.

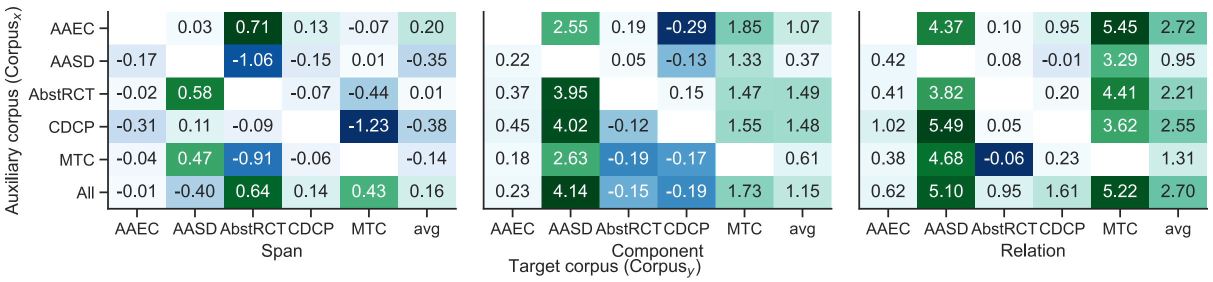


Figure 5: Error reduction (not ERR) by MT-AM (Corpus$_x$ → Corpus$_y$) for ST models (with OS for components and relations), showing each auxiliary corpus (vertical; Corpus$_x$) and target corpus (horizontal; Corpus$_y$) combination. For example, error reduction in the AAEC →AASD relation is 4.37. ‘‘All’’ represents MT-All. Darker green shows positive effects of MT-AM and darker blue shows negative effects. Note that the darkness is normalized in each task (i.e., span, component, or relation).

they produce lower error reduction (AASD-‘‘avg’’ cells and MTC-‘‘avg’’ cells). In both component and relation classifications, the average error reductions of AAEC and CDCP as an auxiliary corpus (i.e., AAEC-‘‘avg’’ and CDCP-‘‘avg’’ cells) were better among all corpora. Because CDCP forms less constrained argument structures than tree-constrained argument structures such as MTC and AASD, we presume that one factor for the high transferability of CDCP stems from its ability to represent broad argumentative relations that may appear in a target corpus. AbstRCT also showed better error reductions, namely, 1.49 in component and 2.21 in relation, as an auxiliary corpus (i.e., AbstRCT-‘‘avg’’ cells). However, some auxiliary corpora barely improved error reductions on the AbstRCT components and relations. This asymmetry result can be due to a domain difference.

The above results indicate that the average error reduction varies depending on the auxiliary corpus (Corpus$_x$). On the other hand, the average performance of MT-All (i.e., the ‘‘All’’-‘‘avg’’ cells) shows stable error reductions. For example, when CDCP was a target corpus, the average error reduction of MT-All (i.e., ‘‘All’’-CDCP cells) was better, namely, 0.14 in span and 1.61 in relation, when compared with all Corpus$_x$ → CDCP pairs. We presume that MT-All stabilized the fluctuation of performance produced by each auxiliary corpus.

Label-wise Analysis: The label-wise best corpus combinations for the relation perspective are shown in Table 6 at $n = 100\%$. We report the best auxiliary corpus with score transition from ST to MT-AM for each target corpus and its relation labels. For example, the ST model trained on AAEC produced a 44.54 F-score for Attack relations, the MT-AM (CDCP→AAEC) that was best among all corpus combinations produced a 46.60 F-score, and the ERR between the ST and MT-AM was 3.71. The results in the table show that, similar to the corpus-wise analyses, there is a combination preference. For example, AAEC and CDCP seem to be the best combination, where injecting auxiliary CDCP into AAEC produced a 3.71 ERR in Attack relation and injecting auxiliary AAEC into CDCP produced a 1.59 ERR in Reason relation. CDCP and AAEC are also better auxiliary corpora for MTC, for example, 18.51 ERR for Exa relation by CDCP→MTC was obtained. On the other hand, CDCP, AbstRCT, and AAEC are better auxiliary corpora for specific labels of AASD. While CDCP is better for Additional and Detail relations, AAEC is

Target corpus (Corpus$_y$)	Relation	Best auxiliary corpus (Corpus$_x$)	ST → MT	ERR
AAEC	Attack	CDCP	44.54 → 46.60	3.71
	Support	CDCP	67.37 → 68.35	3.01
MTC	Exa	CDCP	3.47 → 21.33	18.51
	Reb	AAEC	53.56 → 59.58	12.96
	Sup	AAEC	59.68 → 64.22	11.26
	Und	AAEC	40.38 → 47.79	12.43
CDCP	Evidence	AAEC	0.00 → 0.67	0.67
	Reason	AAEC	40.79 → 41.73	1.59
AbstRCT	Attack	CDCP	34.26 → 37.91	5.55
	Partial-Attack	MTC	48.37 → 52.28	7.56
	Support	AASD	58.72 → 58.46	–0.63
AASD	Additional	CDCP	67.34 → 70.11	8.49
	Detail	CDCP	50.49 → 57.54	14.26
	Sequence	AbstRCT	6.21 → 17.19	11.71
	Support	AAEC	66.75 → 71.56	14.46

Table 6: Best auxiliary corpus for relations (with OS). ST → MT shows score transition from ST to Corpus$_x$ → Corpus$_y$.

better for the Support relation and AbstRCT is better for the Sequence relation. This suggests there could be corpus preferences for specific relation labels.

The results in Table 6 also indicate that the ERRs of Attack and Partial-Attack of AbstRCT significantly improved (5.55 and 7.56 ERR, respectively). In addition, we observed a significant improvement for labels such as Exa, Reb, and Und of MTC (18.51, 12.96, and 12.43 ERR, respectively). Since the numbers of these labels are smaller in each corpus (as shown in Table 1), we conjecture that the prediction performance of such lower-resource labels improved thanks to the multi-task learning. Figure 6 shows the case of an MTC sub-graph at $n = 100\%$. The ST model trained on MTC sometimes detected a false-positive Und relation from *Although Ukraine...* to *Despite the...*, while MT-AM successfully removed the false-positive relation. We explain this through the ST model trained on AAEC that was used to parse the MTC text. We found the predicted graph of the ST model (AAEC) was similar to the gold graph, since both graphs represent the second component (i.e., *the EU...*) as a root component (i.e., MajorClaim). We conclude that this transferability between the auxiliary and target corpora successfully removed the false-positive relation.

The ERRs for certain relations did not improve with MT-AM. For example, the ERR of AbstRCT Support was −0.63 (see Table 6). We presume that there are two reasons for this: (i) the number of Support labels is large enough to achieve a reasonable prediction performance and (ii) AbstRCT is a more challenging target corpus due to domain differences. For example, medical knowledge of AbstRCT cannot be obtained from other corpora. Also, a different scheme for relations, for example, a lower number of non-crossing relations (as shown in Table 2), could not be transferred from other corpora.

5.4 Discussion: Typology of Transferability in AM

We further discuss the results under low-resource settings to examine the typology of transferability. We hypothesize that there are three possible types of transferability: data/training sufficiency, annotation compatibility, and semantically induced compatibility, as follows.

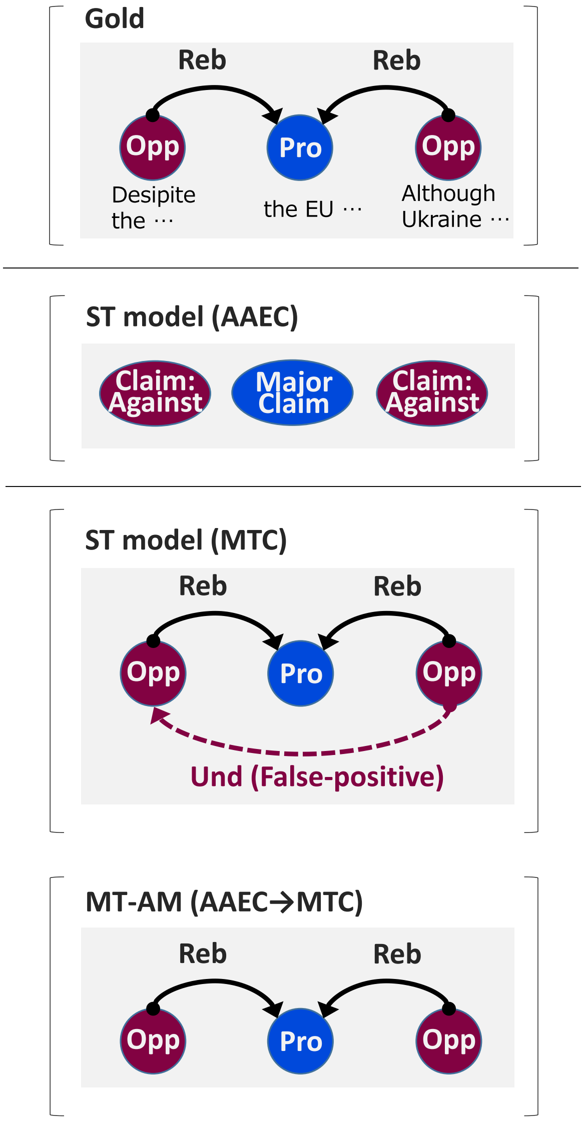
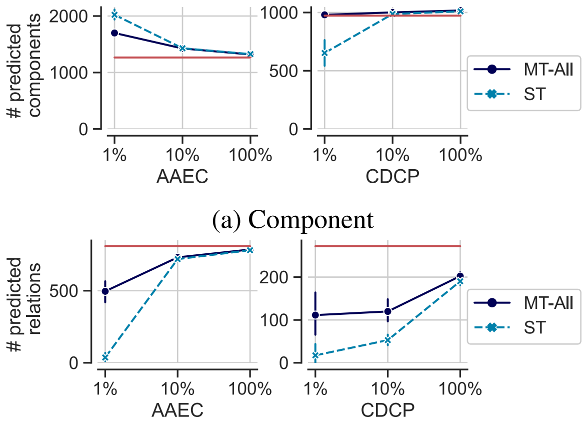


Figure 6: Case study: Predicted results of an MTC sub-graph with OS. For visibility, we combine model outputs for several CVs or seeds.

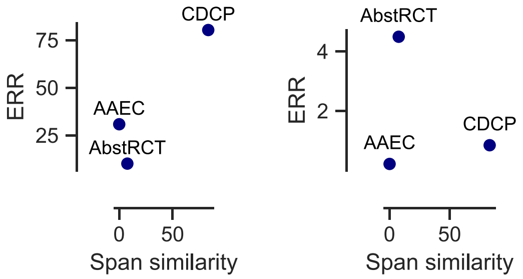
1. Data/Training Sufficiency: Models will not output minor classes such as ''B'' in the BIO tags and positive link classes when insufficient training data is available or a small number of training steps is applied. We found that MT-AM alleviates this problem. Figure 7 shows the number of predicted components and relations, regardless of whether the outputs are correct or not. For relations, we can see in Figure 7b that MT-All helps increase the link outputs for both AAEC and CDCP under low-resource settings, suggesting that the minor positive link class is more often predicted in MT-All. For components, we can see in Figure 7a that the number of predicted components increases for CDCP at $n = 1\%$, while AAEC shows a different phenomenon where the



(a) Component

(b) Relation (with OS)

Figure 7: Number of predicted components and relations for each training data amount ($n\%$). The red line shows the gold number.



(a) $n = 1\%$ (b) $n = 100\%$

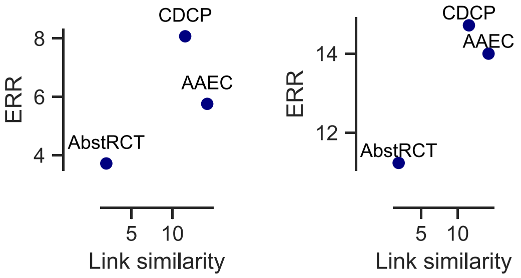
Figure 8: Relationship between *span* similarity (F-score of ST model; horizontal axis) for target corpus AASD and ERR.

number of predicted components by ST is larger than those by MT-All at $n = 1\%$. This is because there is a greater number of ''O'' tags in AAEC since the span of the corpus is not segment-based.

2. Annotation Compatibility: Under low-resource settings, because a model cannot exploit much information of a target corpus in the training, compatibility of the annotation design between the auxiliary corpus and target corpus would directly help multi-task training improve the parsing performance. To demonstrate this, we define the span or link similarity[10] that approximates the compatibility between an auxiliary corpus and target corpus. For span similarity, we compute the span F-score for a target Corpus_y by using an ST model trained on another Corpus_x. Similarly,

Gold	This paper ... reading. We observe, ... logs. The DSSM ... minimized. The effectiveness ... search. The results ... interestingness and that ... models.
ST (AASD) ($n = 1\%$)	This paper ... reading. We observe, ... models.
CDCP →AASD ($n = 1\%$)	This paper ... reading. We observe, ... logs. The DSSM ... minimized. The effectiveness ... search. The results ... models.
AbstRCT →AASD ($n = 1\%$)	This paper ... reading. We observe, ... logs. The DSSM ... minimized. The effectiveness ... search. The results ... models.
AbstRCT →AASD ($n = 100\%$)	This paper ... reading. We observe, ... logs. The DSSM ... minimized. The effectiveness ... search. The results ... models.

Table 7: Predicted AASD spans. The red box shows a span start character, and the blue box shows the end. When we add up all the CV results, the darker box is the more predicted. See also the original abstract in Gao et al. (2014).



(a) $n = 1\%$ (b) $n = 100\%$

Figure 9: Relationship between *link* similarity (F-score of ST model) for target corpus AASD and ERR.

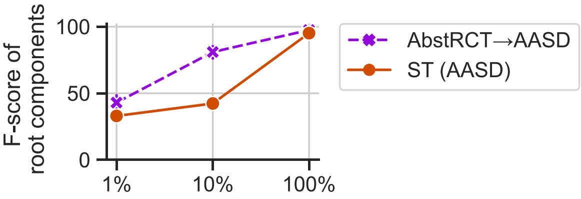


Figure 10: F-score of the root component detection (with OS) with AbstRCT→AASD for each training data amount ($n\%$).

we compute the link F-score with OS for a target Corpus_y.

Since AASD is the smallest corpus in this study, we examined the relationship between similarity and the ERR of the corpus. Figures 8 and 9 show the relationships between the computed similarities and ERRs by $\text{Corpus}_x \rightarrow$ AASD.[11] Under

[10] Because component-label types of an auxiliary corpus differ from those of a target corpus, we only investigated span and link (i.e., unlabeled) similarities.

[11] We exclude MTC as an auxiliary corpus because it is too small to compare with the other corpora.

(a)	ST model trained on AbstRCT	0. Models that..	1. Performance advantages..	2. However, such..	3. In this..	4. Our architecture..	5. We discuss..
(b)	AbstRCT →AASD ($n = 10\%$)	0. Models that..	1. Performance advantages..	2. However, such..	3. In this..	4. Our architecture..	5. We discuss..
(c)	ST model trained on AASD ($n = 10\%$)	0. Models that..	1. Performance advantages..	2. However, such..	3. In this..	4. Our architecture..	5. We discuss..
(d)	AbstRCT →AASD ($n = 1\%$)	0. Models that..	1. Performance advantages..	2. However, such..	3. In this..	4. Our architecture..	5. We discuss..
(e)	ST model trained on AASD ($n = 1\%$)	0. Models that..	1. Performance advantages..	2. However, such..	3. In this..	4. Our architecture..	5. We discuss..

Table 8: Predicted root components (with OS). We add up all CV or seed results, so the darker the color, the more predicted. Gold root component is fourth (i.e., *3. In this...*).

the extreme low-resource setting ($n = 1\%$), as shown in Figures 8a and 9a, we observe the potential for positive correlations between the similarities and ERRs. This implies that the annotation compatibility directly helps multi-task learning.

Specifically, in terms of the span, CDCP showed the best compatibility with AASD in Figure 8a ($n = 1\%$) because both span similarity and ERR are highest. AbstRCT showed the worst compatibility with AASD because both span similarity and ERR are lower. This is illustrated in Table 7, which represents the span predictions. At $n = 1\%$, the ST (AASD) often produced few segments, while the CDCP → AASD split multiple sentences into segments that were almost compatible with the gold segments. Compared with CDCP → AASD, AbstRCT → AASD produced fewer segments. This is because AbstRCT spans are partially selected from a text, and are not compatible with AASD. On the other hand, when $n = 100\%$ (as shown in Figure 8b), we found fewer positive correlations than in $n = 1\%$, suggesting that MT-AM does not benefit from the compatibility when the training data amount is sufficient. In fact, as we can see with AbstRCT → AASD ($n = 100\%$) in Table 7, the issue of fewer segments produced by AbstRCT → AASD ($n = 1\%$) has already been resolved.

In terms of links, CDCP and AAEC in Figure 9a ($n = 1\%$) showed better compatibility while AbstRCT showed the worst. In contrast with the span, the correlation can still be observed when $n = 100\%$ (Figure 9b).

3. Semantic Compatibility: Although the above-mentioned transferability focused on annotation transfer between auxiliary and target corpus, we argue that another possible transferability also exists. That is, in addition to the direct transfer based on the link and span similarity between the auxiliary corpus and target corpus in the annotation compatibility, we want to examine the transfer of potential features of the argument structure. Through our analysis, we found that MT-AM sometimes improves the root component detection significantly when $n = 10\%$. Figure 10 shows the F-score for the root component detection by AbstRCT→AASD for each training data amount. We can see that, compared with the ST model, auxiliary AbstRCT significantly improves the score when $n = 10\%$. We suppose this is derived from semantic compatibility such as implicit argumentative knowledge.

To validate this, we show the predicted root components of AbstRCT→AASD in Table 8. As mentioned above, annotation compatibility could be important for improving the parsing performance under low-resource settings. Since AASD and AbstRCT do not have much compatibility, as shown in Figure 9a, the gold root component (i.e., the fourth component) of AASD is less compatible with the ST model trained on AbstRCT (Table 8(a)). This makes the prediction of AbstRCT → AASD at $n = 1\%$ (Table 8(d)) less similar to the gold. However, as shown in Table 8 (b), we found that AbstRCT→AASD at $n = 10\%$ predicted the fourth component as the root. Surprisingly, this result was not observed in the ST model trained on AASD (Table 8 (c)). This suggests that AbstRCT is able to transfer implicit argumentative knowledge that can determine the root component of a small target corpus.

5.5 Limitation

While we have shown both the effectiveness and the transferability capability of MT-AM, the results might change depending on the model architecture, implementation, experimental design, hyperparameters, and unrecognized factors of annotations. In addition, the three types of transferability are not likely to be independent of each other and are no better than hypotheses. More experiments and discussions will be our future work.

6 Conclusion

In this paper, we focused on argument mining (AM) for handling a scarcity of resources and proposed an end-to-end multi-task AM (MT-AM) model for various corpora. Experiments showed that the proposed MT-AM generally outperformed single-task models and further improved the parsing performance under low-resource settings.

Our extensive analyses suggest that the advantage of MT-AM is due to its three types of transferability: data/training sufficiency, annotation compatibility, and semantic compatibility. In future work, we will develop more accurate MT-AM parsers on the basis of these transferability hypotheses.

Also, it has been suggested by a reviewer that using a trained parser to generate a silver corpus could be a means of extending the training data. Future work should thus address the development of a silver corpus. We are also interested in multi-lingual training using the proposed system.

Acknowledgments

Three anonymous reviewers and the action editor gave us insightful comments and suggestions for our paper. Computational resources of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) were used. Dr. Naoaki Okazaki at Tokyo Institute of Technology gave us comments for our paper.

References

Pablo Accuosto and Horacio Saggion. 2019. Transferring knowledge from discourse to arguments: A case study with scientific abstracts. In *Proceedings of the 6th Workshop on Argument Mining*, pages 41–51, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-4505

Pablo Accuosto and Horacio Saggion. 2020. Mining arguments in scientific abstracts with discourse-level embeddings. *Data Knowledge Engineering*, 129:101840. https://doi.org/10.1016/j.datak.2020.101840

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA. ACM. https://doi.org/10.1145/3292500.3330701

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6354–6364, Online. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1371

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Filip Boltužić and Jan Šnajder. 2020. Structured prediction models for argumentative claim parsing from text. *Automatika*, 61(3):361–370. https://doi.org/10.1080/00051144.2020.1761101

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. 2019. IMHO fine-tuning improves claim detection. 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 558–563, Minneapolis, Minnesota. Association for Computational Linguistics.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Oana Cocarascu, Elena Cabrio, Serena Villata, and Francesca Toni. 2020. Dataset independent baselines for relation prediction in argument mining. In *Frontiers in Artificial Intelligence and Applications*, Volume 326: Computational Models of Argument (COMMA 2020), pages 45–52. IOS Press.

Michael A. Covington. 2001. A fundamental algorithm for dependency parsing. In *In Proceedings of the 39th Annual ACM Southeast Conference*, pages 95–102.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the essence of a claim? Cross-domain claim identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2066, Copenhagen, Denmark. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1218

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Timothy Dozat and Christopher D. Manning. 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. https://doi.org/10.18653/v1/P18-2077

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *Frontiers in Artificial Intelligence and Applications*, 24th European Conference on Artificial Intelligence (ECAI 2020), pages 2006–2013. IOS Press.

Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. 2015. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2242, Lisbon, Portugal. Association for Computational Linguistics.

Frans H. Van Eemeren, Rob Grootendorst, and Francisca Snoeck Henkemans. 1996. *Fundamentals of Argumentation Theory a Handbook of Historical Backgrounds and Contemporary Developments*. https://doi.org/10.2307/358423

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. https://doi.org/10.18653/v1/P17-1002

Agnieszka Falenska, Anders Björkelund, and Jonas Kuhn. 2020. Integrating graph-based and transition-based dependency parsers in the deep contextualized era. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 25–39, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.iwpt-1.4

James B. Freeman. 2011. Argument Structure: Representation and Theory, Argumentation Library (18), Springer. https://doi.org/10.1007/978-94-007-0357-5

Andrea Galassi, Marco Lippi, and Paolo Torroni. 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. https://doi.org/10.18653/v1/W18-5201

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2021. Multi-task attentive residual networks for argument mining. *CoRR*, abs/2102.12227.

Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. 2014. Modeling interestingness with deep neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2–13, Doha, Qatar. Association for Computational Linguistics.

Alex Graves and Jügen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610. https://doi.org/10.1016/j.neunet.2005.06.042, PubMed: 16112549

Grigorii Guz, Patrick Huber, and Giuseppe Carenini. 2020. Unleashing the power of neural discourse parsers - a context and structure aware approach using large scale pretraining. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3794–3805, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179. https://doi.org/10.1162/COLI_a_00276

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1031

Patrick Huber and Giuseppe Carenini. 2019. Predicting discourse structure using distant supervision from sentiment. 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 2306–2316, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1235

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland. Association for Computational Linguistics.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2020. Generalizing natural language analysis through span-relation representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2120–2133, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.192

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015.*

Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2015. Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11, Denver, CO. Association for Computational Linguistics. https://doi.org/10.3115/v1/W15-0501

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4691–4698, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1464

Anne Lauscher, Goran Glavaš, and Simone Paolo Ponzetto. 2018. An argument-annotated corpus of scientific publications. In *Proceedings of the 5th Workshop on Argument Mining*, pages 40–46, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5206

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818. https://doi.org/10.1162/coli_a_00364

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *Frontiers in Artificial Intelligence and Applications*, 24th European Conference on Artificial Intelligence (ECAI 2020), pages 2108–2115. IOS Press.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics. `https://doi.org/10.18653/v1/P16-1105`

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 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. `https://doi.org/10.18653/v1/2020.acl-main.298`

Terufumi Morishita, Gaku Morio, Hiroaki Ozaki, and Toshinori Miyoshi. 2020. Hitachi at SemEval-2020 task 7: Stacking at scale with heterogeneous language models for humor recognition. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 791–803, Barcelona (online). International Committee for Computational Linguistics. `https://doi.org/10.18653/v1/2020.semeval-1.101`

Vlad Niculae, Joonsuk Park, and Claire Cardie. 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. `https://doi.org/10.18653/v1/P17-1091`

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.conll-shared.1`

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics. `https://doi.org/10.18653/v1/K19-2001`

Joonsuk Park, Cheryl Blake, and Claire Cardie. 2015a. Toward machine-assisted participation in erulemaking: An argumentation model of evaluability. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, ICAIL '15, pages 206–210, New York, NY, USA. Association for Computing Machinery. `https://doi.org/10.1145/2746090.2746118`

Joonsuk Park and Claire Cardie. 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).

Joonsuk Park, Arzoo Katiyar, and Bishan Yang. 2015b. Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 39–44, Denver, CO. Association for Computational Linguistics. https://doi.org/10.3115/v1/W15-0506

Andreas Peldszus. 2014. Towards segment-based recognition of argumentation structure in short texts. In *Proceedings of the First Workshop on Argumentation Mining*, pages 88–97, Baltimore, Maryland. Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-2112

Andreas Peldszus and Manfred Stede. 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. https://doi.org/10.18653/v1/D15-1110

Andreas Peldszus and Manfred Stede. 2016. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation, Lisbon 2015 / Vol. 2*, pages 801–815, London. College Publications.

Isaac Persing and Vincent Ng. 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. https://doi.org/10.18653/v1/N16-1164

Peter Potash, Alexey Romanov, and Anna Rumshisky. 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. https://doi.org/10.18653/v1/D17-1143

Jan Wira Gotama Putra, Simone Teufel, and Takenobu Tokunaga. 2021a. Multi-task and multi-corpora training strategies to enhance argumentative sentence linking performance. In *Proceedings of the 8th Workshop on Argument Mining*, pages 12–23, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jan Wira Gotama Putra, Simone Teufel, and Takenobu Tokunaga. 2021b. Parsing argumentative structure in English-as-foreign-language essays. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–109, Online. Association for Computational Linguistics.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal. Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1050

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018. Multi-task learning for argumentation mining in low-resource settings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 35–41, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-2006

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-2038

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1006

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659. https://doi.org/10.1162/COLI_a_00295

Manfred Stede and Jodi Schneider. 2018. *Argumentation Mining*, volume 11. Morgan Claypool Publishers. https://doi.org/10.2200/S00883ED1V01Y201811HLT040

Stephen E. Toulmin. 2003. *The Uses of Argument*, second ed. Cambridge University Press. https://doi.org/10.1017/CBO9780511840005

Dietrich Trautmann. 2020. Aspect-based argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 41–52, Online. Association for Computational Linguistics.

Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. Fine-grained argument unit recognition and classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9048–9056. https://doi.org/10.1609/aaai.v34i05.6438

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-demos.22

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. In *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 5998–6008.

David Vilares and Carlos Gómez-Rodríguez. 2018. A transition-based algorithm for unrestricted AMR parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 142–149, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-2023

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2692–2700. Curran Associates, Inc.

Henning Wachsmuth, Giovanni Da San Martino, Dora Kiesel, and Benno Stein. 2017. The impact of modeling overall argumentation with tree kernels. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2379–2389, Copenhagen, Denmark. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1253

Hao Wang, Zhen Huang, Yong Dou, and Yu Hong. 2020. Argumentation mining on essays at multi scales. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5480–5493, Barcelona, Spain (Online). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.478

An Yang and Sujian Li. 2018. SciDTB: Discourse dependency TreeBank for scientific abstracts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 444–449, Melbourne, Australia. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2071

Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 669–678, Online. Association for Computational Linguistics.