Argument Mining with Modular BERT and Transfer Learning

Umer Mushtaq[1] and Jérémie Cabessa[2,3],
[1]Laboratoire d'économie mathématique et de microéconomie appliquée (LEMMA)
Université Paris-Panthéon-Assas, Paris, France
[2]Laboratoire DAVID, UVSQ – Université Paris-Saclay, Versailles, France
[3]Institute of Computer Science of the Czech Academy of Sciences, Prague, Czech Republic
umer.mushtaq@u-paris2.fr, jeremie.cabessa@uvsq.fr

***Abstract*—We introduce BERT–MINUS, a modular, feature-enriched and transfer learning enabled model for Argument Mining. BERT–MINUS consists of: 1) a joint module which embeds the paragraph text, and 2) a dedicated module, consisting of three customized BERT models, which contextualize the argument markers, argument components and additional features given as text, respectively. BERT–MINUS implements two kinds of transfer learning – auto-transfer (transfer from a task to itself) and cross-transfer (classical transfer from one task to another) – via a novel Selective Fine-tuning mechanism. BERT–MINUS achieves state-of-the-art results on the Link Identification task and competitive results on the Argument Type Classification task. The synergy between the Features as Text and Selective Fine-tuning mechanisms significantly improves the performance of the model. Our work reveals the importance and potential of transfer learning via selective fine-tuning for modular Language Models. Moreover, this study dovetails naturally into the Prompt Engineering paradigm in NLP.**

***Index Terms*—NLP, Argument Mining, BERT, modular BERT, Features as Text, Transfer Learning.**

I. Introduction

In Natural Language Processing (NLP), Argument Mining is concerned with identification and analysis of argumentative and discursive structures in texts [1]. This field is gaining increasing importance with the growing amount of textual data involving argumentative discourse from different sources and domains. For instance, legal texts contain law-based reasoning with a complex underlying argumentative structure [2]. Essays and articles consist of ordered presentation of claims and premises on a certain topic [2, 3]. Organized political debates involve argumentative dialogues between candidates on different issues [4, 5]. Social media platforms provide an avenue for users to debate and discuss contentious issues [6].

A complete end-to-end Argument Mining pipeline consists of the following related sub-tasks [7, 8]: 1) Argument Component Detection (ACD): given a token, classify whether it is part of an argument component or not; 2) Argument Type Classification (ATC): given an argument component, classify it as a *Major Claim*, *Claim* or *Premise*; 3) Link Identification (LI): given an argument component, classify it as either *Linked* or *Not Linked* to another argument component and 4) Link Type Classification (LTC): given a linked argument component, classify whether the link is of a *Support* or of an *Attack* type. The end output of the Argument Mining pipeline is a tree-like structure of the argumentative text [9] where the classified argument components are the nodes and links between argument components are the edges. This structure can then be utilized for downstream reasoning-based applications, like Text Summarization and Question Answering. The Argument Mining sub-tasks have been approached from both single-task and joint-task learning perspectives, using model architectures of varying complexity and with or without additional features [9–13] (see Section II Related Works for more details).

Transformer models [14], like Bidirectional Encoder Representations from Transformers (BERT) [15], have revolutionized NLP. The BERT model, composed of stacked encoder blocks of the Transformer, combines the advantages of the powerful attention mechanism [14] with a fast and parallelizable feed-forward architecture. BERT is trained in a two stage process: a self-supervised stage where the model is pre-trained on a huge textual corpus, followed by a supervised stage in which the pre-trained model is fine-tuned on a downstream task. BERT and its distilled versions have been successfully used for several NLP tasks [14, 16]. When used as sentence representations, BERT outperforms earlier embeddings like GloVe, ELMo, FastText, etc.

Despite its high efficiency, a standalone BERT fine-tuned on isolated argument components suffers from performance limitations [17]. This is due to the complicated and nuanced nature of argumentative texts, where the text of an argument component alone does not provide sufficient information for its accurate classification. In fact, the role of an argument component depends strongly, among other factors, on the presence of argument markers ('Consequently,', 'However,' etc.). Additionally, accurate classification of an argument component also requires positional and structural information about the component: its position in the paragraph and the complete essay, etc. [9]. Therefore, it is crucial for a BERT-based model for Argument Mining to have the ability to capture the contextual, structural and syntactic features which are essential

This research was supported by Labex MME-DII as well as by the Czech Science Foundation, grant AppNeCo No. GA22-02067S, institutional support RVO: 67985807.

for accurate classification. Accordingly, our approach in this paper seeks to address these dynamics exactly: we first embed the complete paragraph, allowing for connective clues and structural flow between components to be captured. Then, we contextualize the three essential feature groups (contextual, structural and syntactic) in parallel. Finally, we combine the separate contextualized feature groups to form a targeted and enriched representation of the ADU.

In Argument Mining, transfer learning between the Argument Type Classification (ATC) and the Link Identification (LI) tasks is of particular relevance [13]. For example, in the ATC task, the classifier learns that the first component of a paragraph has a higher probability of being a claim. Then, via transfer learning, the classifier can use this information in the LI task to deduce that the first component in a paragraph is most likely linked to some other component, since claims are almost always linked, either by outgoing links to major claims or by incoming links from premises in the paragraph.

This work focuses on Argument Mining in the Persuasive Essays (PE) dataset which consists of written essays on various topics. We introduce a modular BERT-based model, called *BERT–MINUS*, which consists of four BERT models, a custom *Features as Text (FeaTxt)* sentence representation, and a *Selective Fine-tuning* process for transfer learning. The architecture of this model is a generalization of the LSTM–Minus model of Kuribayashi et al. [13]. The Features as Text (FeaTxt) enhancement is inspired by the cutting-edge Prompt Engineering approach [18] and is also in line with the work of Mushtaq and Cabessa [17].

The BERT–MINUS model works as follows: the Joint Module embeds a complete input paragraph which consists of several Argumentative Discourse Units (ADU) to be classified. Taking this paragraph embedding as input, the Span Representation Function computes span-based representations for argument markers (AM), argument components (AC), and additional features – *given as text* (FeaTxt). Subsequently, the Dedicated Module, composed of three BERT models, contextualizes these span representations separately to better capture the flow between them. These contextualized representations are then concatenated to obtain a combined representation of the ADU which is finally fed to a classification layer.

To exploit transfer learning between or across LI and ATC tasks, we endow the BERT–MINUS model with both intra-task and inter-task (classical) transfer learning capabilities through the *Selective Fine-tuning* mechanism.

The BERT–MINUS model achieves state-of-the-art results on the LI task and competitive results on the ATC task. Moreover, the synergy between the Features as Text and the selective fine-tuning mechanisms significantly improve the performance of BERT–MINUS. More generally, our study reveals the importance of careful fine-tuning for modular language models. It also naturally dovetails into the Prompt Engineering paradigm in NLP. We make the code available on GitHub at the following address:
https://github.com/mohammadoumar/BERT–MINUS–FeaTxt.

The main contributions of this paper are as follows:

- We introduce a modular BERT-based model, called BERT–MINUS, which consists of four BERT models which separately, and in parallel, contextualize AMs, ACs and FeaTxt of an ADU to form a targeted and enriched embedding of the ADU.
- We introduce a two-mode Selective Fine-tuning process for transfer learning between Link Identification (LI) and Argument Type Classification (ATC).
- BERT–MINUS achieves state-of-the-art results on the LI task and competitive results on the ATC task. The Features as Text and Selective Fine-tuning mechanisms significantly improves the performance of the model.

This paper is structured as follows. Section II presents the literature related to our work. Section III introduces the BERT–MINUS model and the selective fine-tuning mechanism in detail. Section IV describes the experimental setup of our work. In Section V, we present our results and analyse them. We conclude and propose future directions in Section VI.

II. Related Works

In the literature, several distinct approaches have been proposed for Argument Type Classification (ATC) and Link Identification (LI) in structured texts. For both tasks, different architectures and feature sets have been studied and analyzed.

Stab and Gurevych [9] investigated ATC, LI and Link Type Classification (LTC) in the Persuasive Essays (PE) dataset. They used Support Vector Machines (SVM) and Conditional Random Fields (CRF) with hand-crafted feature sets consisting of lexical, structural, syntactic, contextual and discursive features. For ATC, they report that structural features produce the best results. For LI, a combination of features yields the best performance. Their work reveals the importance of well-designed feature groups for Argument Mining sub-tasks. Our work incorporates their feature groups approach into transformer-based language models.

Hadaddan et al. [11] focused on the ACD and ATC tasks. They introduced the Yes We Can! (YWC) dataset which consists of transcribed political speeches. They present both feature-based and recurrent neural network-based approaches. The former involves simple feed-forward networks with features consisting of Bag of Words (BoW), N-Grams, Part of Speech (POS) tags, Named Entity Recognition (NER) tags, etc. The latter approach involves Feed-Forward and LSTM architectures with FastText word embedding. Their work posits the importance of syntactic and grammatical features for Argument Mining.

Potash et al. [10] approached both ATC and LI as a joint learning task. They introduced a custom Joint Neural Model for the Persuasive Essays (PE) and Micro-Text Corpus (MTC) datasets. This model consists of a Bi-LSTM encoder combined with a fully connected layer for ATC and an LSTM decoder for LI. For textual representation, they use Bag of Words (BoW), GloVe embedding and structural features. This approach combines the advantages of additional features and embeddings when used in conjunction with recurrent neural networks.

Mayer et al. [12] combined the ACD and ATC tasks into one sequence tagging task. They use a dataset based on abstracts of Randomized Controlled Trials (RCT) from the MEDLINE database. They use combinations of static and dynamic embeddings as textual representations together with LSTMs, GRUs and BERT fine-tune for an end-to-end Argument Mining pipeline.

Kuribayashi et al. [13] present a model which builds upon the *LSTM–Minus* span representations of Wang and Chang [19] and Li et al. [20]. The LSTM–Minus span representation of a text span (i, j) is defined as the subtraction ('Minus') of the hidden layer outputs of the LSTM model at indices j and i. Based on this definition, Kuribayashi et al. presented two cases: (i) a *joint span model* where an argumentative discourse unit (ADU) is considered as a single span (i, j) and (ii) a *distinction model* where the ADU span (i, j) is separated into an argument marker span (i, k) and an argument component span $(k + 1, j)$. The motivation for the latter model is to better capture the flow between the argument markers ('I think', 'because', etc.) and the argument components ('we should limit immigration', 'tertiary education is more important than secondary', etc.) [13].

In the Kuribayashi et al. distinction model, the span representation of both the argument marker and the argument component is computed according to the *LSTM–Minus representation formula*. Then, these two representations are contextualized using two separate Bi-LSTMs. Finally, these contextualized representations are concatenated, optionally with BoW and structural features, to obtain the representation of the complete ADU. Kuribayashi et al. considered three tasks: ATC, LI and LTC, both separately and jointly. In the joint learning setting, they used a custom loss function consisting of weighted combination of loss functions for all three tasks. They experiment with both the PE and MTC datasets.

Finally, Mushtaq and Cabessa [17] introduced the *BERT with Features as Text (BERT–FeaTxt)* model for ATC. They present a combined features as text sentence representation which incorporates contextual, structural and syntactic features along with the argument component. The contextual features are the topic and the full sentence, while the structural features relate to the position of the component in the essay and the paragraph. As syntactic features, Part Of Speech (POS) tags of the component are used. This enriched sentence representation is then utilized to fine-tune a BERT model for the ATC task.

Mushtaq and Cabessa [17] experiment with the PE, Change My View (CMV) and Yes We Can! (YWC) datasets. They report two important results: firstly, the BERT–FeaTxt model outperforms standalone BERT, and secondly, BERT–FeaTxt outperforms the classical case where structural features are concatenated numerically to the BERT embedding.

In this paper, we combine our previous work [17] with the Kuribayashi span-representation approach [13]. We seek to leverage the features as text capabilities of our BERT–FeaTxt model and the enhanced span-representation capabilities of the Kuribayashi model.

III. Model

In this section, we first recall the *BERT with Features as Text (BERT–FeaTxt)* model [17]. We then introduce our modular *BERT–MINUS* model for Argument Type Classification (ATC) and Link Identification (LI). Finally, we explain how the BERT–MINUS model can leverage transfer learning via the *Selective Fine-tuning* mechanism.

A. BERT–FeaTxt

Contextual, structural and syntactic features are crucial for building meaningful representations of argument components [9]. Accordingly, Mushtaq and Cabessa [17] introduced the *BERT with Features as Text (BERT–FeaTxt)* model. In addition to the argument component itself, this model incorporates in its input hand-crafted features – *given in textual form* – rather than in numerical form. This approach leverages the bidirectional contextual and linguistic capabilities of BERT, enabling it to create an enriched representation of the whole input text. The BERT–FeaTxt model and the textual representations of its features are described in more detail below.

Contextual Features: The full meaning of an argument component depends inherently on the linguistic and semantic context in which it occurs. Therefore, contextual information is an important factor in the classification of an argument component. Accordingly, BERT–FeaTxt utilizes: 1) the full sentence in which the argument component occurs and 2) the topic of the essay as the contextual features for an argument component. Formally, the textual representation of contextual features is given as follows:

`contextual_fts` = 'Topic: t. Sentence: s.'

Structural Features: Written essays naturally follow a structured argumentative pattern. The essay usually begins with a statement of the writer's stance on the topic. Thereafter, claims in support of the stance and premises to support these claims are presented in successive paragraphs. Therefore, the position of the argument component in the essay and paragraph contains vital information for its classification. As structural features, BERT–FeaTxt utilizes: 1) the paragraph number in which the argument component appears, 2) whether it is in the introductory or 3) concluding paragraph, and 4) if it is the first or 5) last component in the paragraph. Formally, the textual representation of structural features is given as follows:

`structural_fts` = 'Paragraph Number: n. Is in introduction: i. Is in conclusion: c. Is first in paragraph: f. Is last in paragraph: l.'

Syntactic Features: The linguistic and grammatical characteristics of an argument component are also a factor in determining its argumentative role. Accordingly, BERT–FeaTxt incorporates Part of Speech (POS) tags of the argument component as its syntactic features. POS tags determine whether each token is a noun, a verb, an adjective, and so on. Formally, the textual representation of syntactic features is given as follows:

`syntactic_fts` = 'Part Of Speech tags: t_1, t_2, ..., t_n'

where t_i represents the POS tag of the i-th token in the argument component.

Combined Features as Text: As its input, the BERT–FeaTxt model combines the contextual, structural and syntactic features as follows:

```
combined_fts = contextual_fts +
               structural_fts +
               syntactic_fts
```

where '+' denotes the string concatenation operation. Note that the argument component itself is, by definition, included in the contextual features.

B. BERT–MINUS

We now introduce our modular *BERT–MINUS* model in detail. BERT–MINUS contextualizes AM, AC and FeaTxt of an ADU in parallel to form a targeted and enriched embedding of the ADU. In the main, BERT–MINUS consists of three parts: 1) a joint BERT module, 2) a span representation function and 3) a dedicated module consisting of three BERT models with customized input embeddings. In addition to these parts, the BERT–MINUS model also has intermediate layers and an output layer (see Figure 1).

Input: The input to the BERT–MINUS model consists of a paragraph from an essay and the spans tensor of the paragraph (see Figure 1, Paragraph and Spans). Each paragraph contains a number of Argumentative Discourse Units (ADU). Each ADU consists of an Argument Marker (AM) (blue text in Figure 1) and an Argument Component (AC) (red text in Figure 1). For a text sequence, its span is the pair of indices, (i, j), of its first and last token in the tokenized paragraph.

The BERT–MINUS model can be utilized both without or with features as text (FeaTxt). In the former case, the spans tensor consists of the AM spans (i_{am}, j_{am}) and the AC spans (i_{ac}, j_{ac}) of all ADUs in the paragraph. In the latter case, the spans tensor also includes the spans (i_{fts}, j_{fts}) of the features as text (cf. Section III-A) of all ADUs. The spans tensor for the paragraph, then, consists of the list of spans

$$\big[(i_{am_1}, j_{am_1}), (i_{ac_1}, j_{ac_1}), (i_{fts_1}, j_{fts_1}), \\ (i_{am_2}, j_{am_2}), (i_{ac_2}, j_{ac_2}), (i_{fts_2}, j_{fts_2}), \dots\big]$$

of all the ADUs ($i = 1, 2, \dots$) in the paragraph.

Joint Module: The first module of BERT–MINUS is a standalone pre-trained BERT model (see Figure 1, $\mathrm{BERT}_{\mathrm{joint}}$). We use this model to contextualize and embed the input paragraph.

Span Representation Function: The span representation function takes two objects as input: the output of the Joint Module, which is a sequence of 768 dim vectors whose length equals the number of tokens in the paragraph, and the spans tensor of the paragraph. This function computes three span representations: one each for the AM, AC and FeaTxt of every ADU in the paragraph (see Figure 1, Span Representation Function). For a text sequence (AM, AC or FeaTxt of an ADU) with span (i, j), its BERT–MINUS span representation is computed as follows:

$$\big[\mathbf{h_j} - \mathbf{h_{i-1}} \;;\; \mathbf{h_i} - \mathbf{h_{j+1}} \;;\; \mathbf{h_{i-1}} \;;\; \mathbf{h_{j+1}}\big]$$

where $\mathbf{h_i}$ is the output of the Joint Module at the i-th index and ';' represents tensor concatenation. In this computation, the first and second term represents the embedding of the text in the forward and backward direction, respectively. The last two terms capture the preceding and succeeding context of the text sequence (span). These representations are based on the LSTM–Minus representation of Kuribayashi et al. [13].

Each span representation is of dimension $4 * 768 = 3072$. Before they are input to the next module (Dedicated Module), these span representations are reshaped using three parallel intermediate linear layers: $\mathrm{LINEAR}_{\mathrm{am}}$, $\mathrm{LINEAR}_{\mathrm{ac}}$ and $\mathrm{LINEAR}_{\mathrm{fts}}$, respectively, each of input dimension 3072 and output dimension 768.

Dedicated Module: This module consists of three dedicated BERT models, $\mathrm{BERT}_{\mathrm{am}}$, $\mathrm{BERT}_{\mathrm{ac}}$, $\mathrm{BERT}_{\mathrm{fts}}$ (see Figure 1), which process the AM, AC and FeaTxt span representations, respectively. The embedding layer of these models are customized so that they can take sequences of vectors (span representations) instead of token ids as inputs. In this way, each of the AM, AC and FeaTxt span representation is contextualized separately by a dedicated customized BERT model. The outputs of these dedicated models are then used to obtain a combined representation of the whole ADU as follows:

$$\begin{aligned}\mathrm{REP}_{\mathrm{ADU}} \;=\; \big[&\mathrm{BERT}_{\mathrm{am}}(\mathrm{am\_span\_representation}); \\ &\mathrm{BERT}_{\mathrm{ac}}(\mathrm{ac\_span\_representation}); \\ &\mathrm{BERT}_{\mathrm{fts}}(\mathrm{fts\_span\_representation})\big]\end{aligned}$$

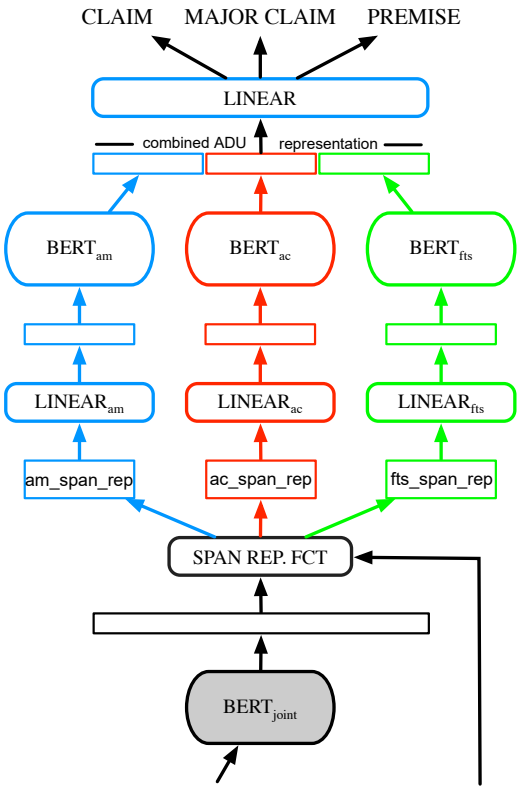
This BERT–MINUS ADU representation is finally fed to a fully connected layer for classification into the respective classes for the two tasks.

C. Selective Fine-Tuning

To enable transfer learning between Link Identification (LI) and Argument Type Classification (ATC) tasks, we adjoin a three-step, two-mode *Selective Fine-tuning* mechanism to our BERT–MINUS model:

1) A pre-trained BERT model is fine-tuned on one task, ATC or LI.
2) This fine-tuned model is instantiated as the Joint BERT module of the BERT–MINUS model.
3) BERT–MINUS is fine-tuned, either on the same task as Step 1 (*auto-transfer* mode) or on the other task (*cross-transfer* mode).

Instead of a generic pre-trained BERT model, the selective fine-tuning mechanism uses a BERT model already fine-tuned on one of the two tasks. By means of transfer learning, the paragraph embedding computed by the joint BERT module is more targeted towards the particular task. In addition, the selective fine-tuning mechanism is also motivated by Wieting



Paragraph:
The issue of whether using of machine are bring many advantages to society is of great concern to many people. In my opinion, although using machines have many benefits, [SEP] 1, Yes, No, Yes, No [SEP] we cannot ignore its negative effects. [SEP] 1, No, Yes, Yes, No [SEP]

Spans:
[[21, 23], [25, 25]]
[[25, 30], [43, 48]]
[[33, 41], [51, 59]]

Fig. 1: Architecture of the BERT–MINUS model. The paragraph and the spans tensor are input to the model. The paragraph contains AMs (blue text), ACs (red text), and additional features as text (green text, only abbreviated form shown for brevity's sake), separated by the [SEP] tokens. The spans tensors consists of the span indices of AMs, ACs and features as text of the ADUs in the paragraph. The paragraph is fed to a joint BERT model. The output of this model, together with the spans tensor, are fed to the spans representation function. The AM, AC and FeaTxt BERT–MINUS representations obtained from this function are reshaped via three linear layers. These reshaped representations are fed to three dedicated BERT models. The outputs of these models are then concatenated to construct an enriched representation of the whole ADU. This ADU representation is then fed to a final fully connected layer for classification. The selective fine-tuning of the joint BERT module is represented by a gray coloring.

and Kiela [21] who emphasize the importance of the embedding layer over the complexity of the subsequent encoder block.

IV. Experiments

A. Dataset

We use the Persuasive Essays (PE) dataset introduced by Stab and Gurevych [9]. The PE dataset consists of 402 structured essays on various controversial topics such as *'Businesses should be only concerned about making profits'* and *'Spending money on supporting art or protecting environment'*. Of the 402 essays, 322 are set aside for the train set and 80 for the test set. The statistics of the the PE dataset are given in Table I.

For our BERT–MINUS model, we separated each Argumentative Discourse Unit (ADU) of the dataset into an argument marker (AM) and an argument component (AC). To that end, we used the four types of argument markers of Stab and Gurevych: forward, backward, thesis and rebuttal [9].

Corpus Statistics		Component Statistics	
Tokens	147,271	Major Claims	751
Sentence	7,116	Claims	1,506
Paragraphs	1,833	Premises	3,832
Essays	402	Total	6,089

Table I: Corpus and component statistics for the PE dataset.

B. Tasks

We focus on the two following Argument Mining sub-tasks:

1) *Link Identification (LI):* Given an argument component (AC), classify it as either *Linked* or *Not Linked*. Here, we approach LI as the task of classifying single argument components, as opposed to pairs of components as in [9, 13]. Since linked claims and linked premises are, by and large, linked to the major claims and the claims at the beginning of the paragraph, respectively, the essay tree structure can be properly reconstructed from the classification of separate components [9].
2) *Argument Type Classification (ATC):* Given an argument component (AC), classify it as either a *Major Claim*, a *Claim* or a *Premise*.

C. Models

In our work, we consider the following models:

- **BERT:** a standalone BERT model fine-tuned on argument components alone, without features as text (FeaTxt).
- **BERT–FeaTxt:** a BERT model fine-tuned on the combined features as text representation, as described in Section III-A. The standalone BERT and BERT–FeaTxt models represent our baselines.
- **BERT–MINUS:** a BERT–MINUS model fine-tuned on paragraph texts as described in Section III-B. This model takes no additional features as text (FeaTxt) as inputs (green features and modules in Figure 1) and has no selective fine-tuning.
- **BERT–MINUS–Auto:** a BERT–MINUS model where the joint BERT module is selectively fine-tuned on the same task (LI or ATC) as the one on which the BERT–MINUS model is being trained, as described in Section III-C. We call this mode *auto-transfer learning*, i.e., transfer from one task to itself.

- **BERT–MINUS–Cross:** a BERT–MINUS model where the joint BERT module is selectively fine-tuned on the opposite task (LI → ATC, and vice-versa) as the BERT–MINUS model. We call this mode (classical) *cross-transfer learning*, i.e., transfer from one task to another.
- **BERT–MINUS–FeaTxt:** a BERT–MINUS model augmented with features given as text (FeaTxt), as described in Section III-A and illustrated in Figure 1, and with no selective fine-tuning.
- **BERT–MINUS–FeaTxt–Auto:** a BERT–MINUS–FeaTxt model with selective fine-tuning in the *auto-transfer* mode.
- **BERT–MINUS–FeaTxt–Cross:** a BERT–MINUS–FeaTxt model with selective fine-tuning in the *cross-transfer* mode.

V. Results and Analysis

We present and analyze the results of the various BERT–MINUS models on the Link Identification (LI) task and the Argument Type Classification (ATC) tasks. We also present the result of the Link Type Classification (LTC) task with the BERT–FeaTxt model [17].

A. Link Identification Task

The results for the LI task are given in Table II. The analysis of these results reveals several important insights and patterns.

Models	L	NL	F1
BERT	0.216	0.833	0.524
BERT–FeaTxt	0.585	0.877	0.731
BERT–MINUS	0.721	0.826	0.773
BERT–MINUS–Auto	0.760	0.830	0.795
BERT–MINUS–Cross	0.750	0.835	0.793
BERT–MINUS–FeaTxt	0.709	0.800	0.755
BERT–MINUS–FeaTxt–Auto	0.763	0.841	0.802
BERT–MINUS–FeaTxt–Cross	0.778	0.850	**0.814**
Stab and Gurevych [9]	0.585	0.918	0.751
Niculae et al. [22]			0.601
Kuribayashi et al. [13]			0.783

Table II: Results for the LI task. The performance of the different BERT and BERT–MINUS models described in Section IV-C are reported. L and NL represents the F1 scores for *Linked*, and *NotLinked*, respectively. F1 stands for the macro F1 score. The empty cells come from the fact that in the literature, only the macro F1 score was given. The 2 first rows concern the BERT model, the 3 next ones the BERT–MINUS model, and the 3 following ones the BERT–MINUS–FeaTxt model.

First, we see that the BERT–FeaTxt model drastically improves on the standalone BERT performance (Table II, rows 1 and 2). This improvement is due to the addition of contextual, structural and syntactic features which allows the model to build richer embeddings of the argument components. This observation comports exactly with the results of Mushtaq and Cabessa [17]. Indeed, they further show that the additional features are better exploited when given in a textual rather than numerical form.

Secondly, we see that the BERT–MINUS model significantly improves over both standalone BERT and BERT–FeaTxt (Table II, rows 1, 2 and 3). Recall that BERT–MINUS takes complete paragraphs as input whereas both BERT and BERT–FeaTxt take single components only. Consequently, BERT–MINUS is better able to capture the contextual and argumentative flow between successive components. As a result, the BERT–MINUS component representations are more contextually enriched, leading to improved accuracy. This suggests that, for some tasks, it is more efficient for a model to build contextualized representations from raw texts (BERT–MINUS) than from descriptive features (BERT–FeaTxt). We will see, however, that this does not apply to the ATC task.

Thirdly, for both BERT–MINUS and BERT–MINUS–FeaTxt models, the selective fine-tuning mechanism improves the results (Table II, rows 3–5 and 6–8). We believe that this phenomenon is due to two important reasons: firstly, when selectively fine-tuned, BERT–MINUS is placed in a more 'informative' initial configuration from which it can reach a lower local minimum during training. Secondly, selective fine-tuning improves the quality of the paragraph embedding which, in turn, positively impacts the whole training process. These results are in line with those of Wieting and Kiela [21], who show the importance of the embedding over the complexity of the subsequent encoder.

Furthermore, note that for BERT–MINUS, both *auto-transfer* and *cross-transfer* modes achieve comparable results (Table II, rows 4–5). By contrast, for BERT–MINUS–FeaTxt, *cross-transfer* significantly outperforms *auto-transfer* (Table II, rows 7–8). In fact, the results achieved by BERT–MINUS with FeaTxt and *cross-transfer* are state-of-the-art.

Finally and surprisingly, BERT–MINUS outperforms BERT–MINUS–FeaTxt (Table II, rows 3 and 6). This shows that, when no transfer-learning is involved, it is actually more efficient for BERT–MINUS to build contextualized representations from raw texts than from descriptive features. By contrast, when transfer-learning come into play, BERT–MINUS–FeaTxt outperforms its BERT–MINUS counterpart (Table II, rows 4–5 and 7–8). In fact, performing both the first and third steps of selective fine-tuning with same features as text leverages and enables transfer learning between the tasks.

B. Argument Type Classification Task

The results of the ATC task are presented in Table III.

As for the LI task, we see that BERT–FeaTxt significantly outperforms standalone BERT (Table III, rows 1 and 2). This shows that contextual, structural and syntactic features – *given as text (FeaTxt)* – capture important information necessary for determining the argumentative role of a component [17].

Secondly, BERT–MINUS also improves upon standalone BERT (Table III, rows 1 and 3). As already explained for the LI task, the contextualized representations built by BERT–MINUS capture argumentative flow from complete paragraphs as opposed to individual components. However, in contrast with the LI task, BERT–FeaTxt outperforms BERT–MINUS

Models	MC	C	P	F1
BERT	0.703	0.507	0.841	0.686
BERT–FeaTxt	0.855	0.678	0.909	0.814
BERT–MINUS	0.784	0.602	0.865	0.750
BERT–MINUS–Auto	0.847	0.617	0.888	0.784
BERT–MINUS–Cross	0.813	0.633	0.888	0.778
BERT–MINUS–FeaTxt	0.746	0.537	0.863	0.715
BERT–MINUS–FeaTxt–Auto	0.900	0.687	0.903	**0.831**
BERT–MINUS–FeaTxt–Cross	0.869	0.618	0.890	0.792
Stab and Gurevych [9]	0.891	0.682	0.903	0.826
Niculae et al. [22]	0.782	0.645	0.902	0.776
Kuribayashi et al. [13]				**0.856**

Table III: Results for the ATC task. The performance of the different BERT and BERT–MINUS models described in Section IV-C are reported. MC, C and P represents the F1 scores for *Major Claim*, *Claim* and *Premise*, respectively. F1 stands for the macro F1 score.

(Tables II and III, rows 1–3). This means that, for this task, the component representations built from descriptive features are more useful than those obtained from full markers and components.

Thirdly, our selective fine-tuning mechanism improves classification accuracy for both BERT–MINUS and BERT–MINUS–FeaTxt (Table III, rows 3–5 and 6–8). As with the LI task, we conjecture that transfer learning yields an improved initial configuration of the BERT–MINUS model as well as an improved embedding of the paragraph text.

Moreover, the *cross-transfer* mode under-performs the *auto-transfer* mode for both BERT–MINUS and BERT–MINUS–FeatTxt (Table II, rows 4–5 and rows 7–8). By comparing these results for the two tasks, we conclude that transfer learning from ATC to LI is more successful than that from LI to ATC. This is explained by the fact that the argumentative role of a component is more useful for inferring its linked or not linked type, than vice versa.

Furthermore, BERT–MINUS outperforms BERT–MINUS–FeaTxt (Table III, rows 3 and 6) for the ATC task as well. However, with selective fine-tuning, BERT–MINUS–FeaTxt outperforms BERT–MINUS (Table II, rows 4–5 and 7–8). As with the LI task, this shows that transfer learning happens properly when both joint BERT module and BERT–MINUS model are fine-tuned with features as text.

Finally, we observe that the combination of the features as text and selective fine-tuning process in *cross-transfer* mode leads to the best results. The synergy of the two mechanisms generates a combined effect that surpasses the sum of its parts. For this task, we improve above Stab and Gurevych's Joint ILP Model [9], but unfortunately, remain below the Kuribayashi LSTM-Minus model [13].

C. Link Type Classification

In addition, to reinforce the results of Mushtaq and Cabessa [17], we also trained BERT–FeaTxt on the Link Type Classification (LTC) task. The results are given in in Table IV.

In the LTC task, BERT–FeaTxt improves the performance of Stab and Gurevych [9]. Once again, this shows that the features as text yield to enriched and improved representations of argument components, leading to better classification accuracy. However, we remain below Kuribayashi et al. [13] which we plan to investigate from the BERT–MINUS perspective in a future paper.

Models	Attack	Support	F1
BERT–FeaTxt	0.506	0.960	0.733
Stab and Gurevych [9]	0.413	0.947	0.680
Kuribayashi et al. [13]			**0.796**

Table IV: Results for the Link Type Classification task. The Stab and Gurevych results are for the full features set and an SVM classifier [9]. The BERT–FeaTxt results are from Mushtaq and Cabessa [17].

VI. Conclusion

In this paper, we focus on two Argument Mining sub-tasks: Link Identification (LI) and Argument Type Classification (ATC) for the Persuasive Essays (PE) dataset. More precisely, we introduce the modular BERT–MINUS model with Features as Text (FeaTxt) and Selective Fine-tuning mechanisms. The model works by constructing an enriched embedding for the whole paragraph text via a joint BERT module and then contextualizing the argument marker, component and additional features as text of the argument discourse unit (ADU) separately via a dedicated module consisting of three customized BERT models. The aggregation of these contextualized representations yields an enriched representation of the ADU. We endow our model with transfer learning capabilities via selective fine-tuning which comes in two modes: *auto-transfer* which implements intra-task transfer, and *cross-transfer* which implements inter-task/classical transfer.

Our experiments show that the BERT–MINUS model with features as text and selective fine-tuning improves over standalone BERT and BERT–FeaTxt for both LI and ATC tasks. The combination of features as text and selective fine-tuning mechanisms significantly augment the capabilities of the BERT–MINUS model. With this enhanced combination, we achieve state-of-the-art results on the LI task and competitive results for the ATC task.

We believe that our work opens up several interesting research directions. For example, an end-to-end Argument Mining pipeline based on our BERT–MINUS-FeaTxt model is the natural next step. Furthermore, we think that selective fine-tuning, both in the *auto-transfer* and the *cross-transfer* modes, can be used to investigate transfer learning between Argument Mining sub-tasks in various architectures and models like Potash et al. [10] and Kuribayashi et al. [13]. Moreover, our BERT–MINUS model is a generalization of LSTM–Minus span representation-based model of Kuribayashi et al. [13]. We think that span representation computations can be enhanced using BERT's particular attention-based contextualization capabilities instead of the LSTM–Minus construction. In addition, following Kuribayashi et al. [13] who report

improvements in the joint-task learning setting, we plan to investigate joint-task learning for the BERT–MINUS model.

More generally, we believe that our selective fine-tuning mechanism opens possibilities for exploration and implementation in other modular Language Models. Finally, our work also dovetails naturally into the cutting-edge Prompt Engineering paradigm in NLP.

REFERENCES

[1] I. Habernal, H. Wachsmuth, I. Gurevych, and B. Stein, "The argument reasoning comprehension task: Identification and reconstruction of implicit warrants," in *Proceedings of NAACL-HLT 2018*, pp. 1930–1940, ACL, 2018.

[2] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, "Automatic detection of arguments in legal texts," in *Proceedings of ICAIL 2007*, pp. 225–230, ACM, 2007.

[3] Y. Song, M. Heilman, B. Beigman Klebanov, and P. Deane, "Applying argumentation schemes for essay scoring," in *Proceedings of ArgMining@ACL 2014*, pp. 69–78, ACL, 2014.

[4] S. Menini, E. Cabrio, S. Tonelli, and S. Villata, "Never retreat, never retract: Argumentation analysis for political speeches," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018* (S. A. McIlraith and K. Q. Weinberger, eds.), pp. 4889–4896, AAAI Press, 2018.

[5] M. Lippi and P. Torroni, "Argument mining from speech: Detecting claims in political debates," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 2979–2985, AAAI Press, 2016.

[6] S. Somasundaran and J. Wiebe, "Recognizing stances in online debates," in *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore* (K. Su, J. Su, and J. Wiebe, eds.), pp. 226–234, ACL, 2009.

[7] E. Cabrio and S. Villata, "Five years of argument mining: A data-driven analysis," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pp. 5427–5433, AAAI Press, 2018.

[8] A. Peldszus and M. Stede, "From argument diagrams to argumentation mining in texts: A survey," *Int. J. Cogn. Informatics Nat. Intell.*, vol. 7, pp. 1–31, 2013.

[9] C. Stab and I. Gurevych, "Parsing argumentation structures in persuasive essays," *Computational Linguistics*, vol. 43, no. 3, pp. 619–659, 2017.

[10] P. Potash, A. Romanov, and A. Rumshisky, "Here's my point: Joint pointer architecture for argument mining," in *Proceedings of EMNLP 2017*, pp. 1364–1373, ACL, 2017.

[11] S. Haddadan, E. Cabrio, and S. Villata, "Yes, we can! mining arguments in 50 years of US presidential campaign debates," in *Proceedings of ACL 2019*, pp. 4684–4690, ACL, 2019.

[12] T. Mayer, E. Cabrio, and S. Villata, "Transformer-based argument mining for healthcare applications," in *Proceedings of ECAI 2020*, pp. 2108–2115, IOS Press, 2020.

[13] T. Kuribayashi, H. Ouchi, N. Inoue, P. Reisert, T. Miyoshi, J. Suzuki, and K. Inui, "An empirical study of span representation in argumentation structure parsing," in *Proceedings of ACL*, pp. 4691–4698, ACL, 2019.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NIPS 2017*, pp. 6000—6010, Curran Associates Inc., 2017.

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT 2019* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 4171–4186, ACL, 2019.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019.

[17] U. Mushtaq and J. Cabessa, "Argument classification with bert plus contextual, structural and syntactic features as text," in *Neural Information Processing - 29th International Conference, ICONIP 2022, IIT Indore, India, November 22-26, 2022, Proceedings* (A. Jatowt and A. Ekbal, eds.), CCIS, p. To appear, Springer, 2023.

[18] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, vol. 55, jan 2023.

[19] W. Wang and B. Chang, "Graph-based dependency parsing with bidirectional LSTM," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 2306–2315, ACL, 2016.

[20] Q. Li, T. Li, and B. Chang, "Discourse parsing with attention-based hierarchical neural networks," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 362–371, ACL, 2016.

[21] J. Wieting and D. Kiela, "No training required: Exploring random encoders for sentence classification," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[22] V. Niculae, J. Park, and C. Cardie, "Argument mining with structured svms and rnns," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers* (R. Barzilay and M. Kan, eds.), pp. 985–995, ACL, 2017.