ENHANCING TRANSFER LEARNING ACROSS ANNOTATION SCHEMES WITH MINERR: A NOVELMETRIC

Samuel Guilluy[1], Florian Méhats[2] and Billal Chouli[3]

[1]IRMAR, Univ Rennes, Rennes, France
[2]IRMAR, Ravel Technologies on leave from Univ Rennes, Rennes, France
[3]HeadmindPartners AI & Blockchain, Paris, France

ABSTRACT

This paper introduces MINERR (MINimal ERRor evaluation metric between consecutive tasks), a novel metric designed to enhance the efficiency of transfer learning in the context of argument structure identification. One of the principal challenges in the Argument Mining field pertains to the need for high- quality training data, which requires achieving a high level of inter-annotator agreement for argument constituents. Therefore, datasets within this domain tend to be smaller compared to those in other domains. To address this issue, we propose the consolidation of different datasets and employ the classical two-step method for argument identification, encompassing the identification of argumentative spans and the categorization of labels. An issue related to the separation of these two tasks is the errors interconnectedness between them. To tackle this problem, we introduce a new metric that distinguishes errors stemming from incorrect labelling and errors arising from span misidentification. Our approach incorporates a novel method for dissecting the prediction error of an argument component labelling task into two distinct categories: errors caused by misidentifying the component and errors resulting from assigning incorrect labels. Subsequently, we evaluate our method using a corpus including four distinct argumentation datasets. Overall, this work facilitates the development of a new transfer learning methodology for the application of diverse argument annotation schemes.

KEYWORDS

Argument Mining, Natural Language Processing, Artificial Intelligence

1. INTRODUCTION

Argumentation, as a multidisciplinary field encompassing philosophy, psychology, linguistics, theoretical mathematics, and artificial intelligence, has witnessed growing interest in recent years. In particular, the application of natural language processing (NLP) methods to identify argumentative text units, such as claims and premises, has gained significant attention [1]. The process typically involves several essential steps, including the identification of argument components and their properties, such as causality and relationships. These steps share similarities with tasks in discourse theory, constituency parsing, Named Entity Recognition, Information Extraction, Dialogue act classification, and Entity Linking.

Despite the existence of theoretical foundations and formalization efforts aiming to unify argumentation with mathematical logic ([2]; [3]), operationalizing this domain remains challenging. One of the primary difficulties lies in obtaining high-quality training data. Achievinga high level of inter-annotator agreement for argument constituents is essential but often

David C. Wyld et al. (Eds): SIPR, NCWC, BIBC, SOFEA, DSML, ARIA, NLP, CSEN -2023
pp. 307-320, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131924

demanding. As a result, comprehensive training is typically required for all three argument identification tasks, hindering the utilization of transfer learning techniques. The consequence is the creation of argument mining corpora comprising a limited number of texts, ranging from a few hundred to a few thousand. Models trained solely on these corpora-specific language patterns, such as topic words or discourse markers, may overlook crucial semantic features and logical dependencies, resulting in poor generalization to new datasets.

To address these challenges, this article focuses on the efficient application of transfer learning to the identification of argument structure. We propose a novel method to analyse prediction errors in argument component labelling tasks. This method allows us to distinguish errors caused by component misidentification from those arising from incorrect label assignment. By dissecting the prediction errors, we can gain insights into the underlying causes and develop strategies for improvement. Specifically, we introduce a metric that facilitates the segmentation of errors between span identification and label identification, enabling a more detailed analysis of the prediction process.

The main contributions of this paper are the followings. Firstly, we conduct basic experiments in order to evaluate the transfer learning method on various Argument Mining Datasets (Section 3). Secondly, we propose a new metric to analyse and understand prediction errors in argument mining tasks (Section 4). By identifying the sources of errors, we can enhance the accuracy and reliability of argument structure identification. Finally, we explore the application of transfer learning techniques to leverage knowledge from one argument mining dataset to improve performance on others (Section 5). This advancement promotes efficient knowledge transfer and enables more effective utilization of limited training data.

2. RELATED WORKS

Foundation of Argumentation Theory. The emergence of contemporary argumentation theory is intricately connected to the domain of discourse theory. Several influential works have established the groundwork for argumentation analysis, a framework still widely employed in present-day NLP research papers, as evidenced by [4] and [5]. Rhetorical Structure Theory (RST), as initially conceptualized by Mann and Thompson [6], provides a comprehensive framework for analyzing the structural arrangement of both textual and discourse components. This theory asserts that the structure of a text can be likened to a hierarchy of units, wherein a top-level discourse structure governs lower-level entities like paragraphs and sentences. RST posits that the connections between these units are founded upon rhetorical relations, which shed light on each unit's role within the broader context of discourse. These relations encompass various functions such as elaboration, evidence, and contrast. In contrast, the Grammar of Discourse, as introduced by [7], introduces the notion of "textual constituency." It posits that discoursecomprises units that are larger than individual sentences yet smaller than an entire text. The author argues that these units can be meticulously described and analysed using a set of grammatical rules and relationships. Additionally, Toulmin's model in "The Uses of Arguments" [8] challengesthe rigidity of traditional argumentation models like syllogisms, contending that they fail to account for the complexities inherent in real-world arguments. Instead, Toulmin proposes a flexible model that comprises six elements: claim, grounds, warrant, backing, qualifier, and rebuttal. This model underscores the significance of comprehending the context within which anargument is made and the underlying assumptions. Several other articles elucidate the interconnections between these theoretical components, including the definition of an Elementary Discourse Unit (EDU) [9], the link between EDUs and Compound Discourse Units (CDUs) [10], and the transition from Argument Discourse Units (ADUs) to Propositions [11].

Unit segmentation. Unit segmentation is widely recognized as the initial phase in the argument mining pipeline. This process entails partitioning a given text into its constituent Argument Discourse Units (ADUs) and their non argumentative counterparts. Subsequently, these argument units are assigned specific roles within the text's argumentative structure, with an emphasis on categorizing the relationships between them. As described in [12], an argument unit may span a clause, an entire sentence, multiple sentences, or fall somewhere in between, depending on various factors such as the argumentative text's domain (including its topic, genre, or other relevant attributes). Moreover, the sizes of these units can exhibit variations even within a single text, rendering unit segmentation a particularly intricate task. As suggested by [13], it is challenging that characterizing a clause's function solely based on grammar and without considering its context. As detailed in [11], while some studies leave the delineation of an Argument Discourse Unit (ADU) to the annotator's judgment [14], many studies rely on a set of syntactic rules as a foundational basis for this process.

Evaluation metrics in NLP. A pivotal aspect evaluation NLP systems involves the utilization of various metrics to quantify their quality and effectiveness. Several prominent metrics have emerged over the years, each with its unique strengths and limitations. BLEU (Bilingual Evaluation Understudy [15]) is renowned for its simplicity and efficiency in measuring the similarity between machine-generated text and human reference translations. Its n-gram precision-based approach has provided researchers with a valuable tool for assessing machine translation systems. The NIST (National Institute of Standards and Technology [16]) metric, which builds upon BLEU by incorporating a weighted geometric mean of n-gram precisions. NIST enhances the effectiveness of NLP evaluation by considering the importance of individual n-grams, thereby providing a more nuanced assessment of system performance. ROUGE (Recall-Oriented Understudy for Gisting Evaluation [17]) is a set of metrics designed for the evaluation of text summarization and document retrieval systems. ROUGE encompasses a range of measures, including precision, recall, and F1-score, making it a versatile tool for assessing the quality of summaries and system-generated content. METEOR (Metric for Evaluation of Translation with Explicit ORdering [18]) leverages a combination of exact word matching and stemming-based measures to account for variations in vocabulary and word choice, providing a comprehensive evaluation framework for machine translation and text generation tasks. TER (Translation Edit Rate [19]) quantifies the minimum number of edits required to transform a system-generated sentence into a human reference translation, offering insights into fluency and grammatical correctness. BERT Score [20] leverages pre-trained language models like BERT to assess the semantic similarity between machine-generated text and human references. This metric has proven invaluable for evaluating the quality of text generation models in contexts where fluency and semantic coherence are critical. Lastly, the Language Error Rate (LEPOR [21]) metric has emerged as a valuable tool for evaluating the grammatical correctness and fluency of machine-generated text. LEPOR employs deep learning techniques to detect and quantify language errors, providing a fine-grained analysis of linguistic quality.

3. EXPERIMENTATION OF A UNIFIED TRAINING APPROACH FOR SPAN IDENTIFICATION

In this section, we will assess the performance of a single model architecture when trained on a single dataset versus when trained on a combination of multiple datasets. This fundamental strategy of dataset consolidation will help us uncover the limitations inherent to this approach.

To begin, we introduce the four datasets utilized in Subsection 3.1 followed by an overview of the models employed in Subsection 3.2. Subsequently, we will present the outcomes of our investigation in Subsection 3.3.

3.1. Data Source Presentation

ARG2020 [22] is an argument mining corpus annotated with argumentative structure composed of "claims" and "premises". It is composed of 145 English argumentative essays selected through the Writing Mentor Educational App. It is based on middle school students writing. A claim is characterized as a potentially debatable statement that signifies an individual's stance in favor of or against a particular proposition. Premises, on the other hand, refer to the rationale provided to either bolster or challenge those claims.

Argument Unit Recognition and Classification (AURC)

[23] is a corpus for argument mining that includes annotations for argumentative structure information, capturing the polarity of arguments on a given topic. The corpus consists of 8000 sentences, evenly distributed across 8 topics. The authors distinguished between PRO (supporting), CON (opposing) arguments, and NON (non-argumentative) words for each topic, in order to construct sentence-level labels. Their labelling rule is as follows: if only NON words occur, the sentence is labelled as NON. If both NON and only PRO (or only CON) words occur, the label PRO (or CON) is assigned. If both PRO and CON words occur, the label that appears more frequently is assigned.

The Cornell eRulemaking Corpus (CDCP)

[24] is a corpus for argument mining that includes annotations for argumentative structure information, specifically capturing the evaluability of arguments. The corpus comprises 731 user comments on the Consumer Debt Collection Practices rule issued by the Consumer Financial Protection Bureau. The resulting dataset contains a total of 4931 elementary unit annotations and 1221 support relation annotations.

Argument Annotated Essays corpus (UKP)

[14] consists of a collection of persuasive essays. The essay corpus is furnished with annotations that identify argument components at the clause level, along with the associated argumentative relationships. Specifically, it includes annotations for major claims, claims, and premises, which are interconnected through argumentative support and attack relations. The corpus was annotated by three ratters, achieving an inter-annotator agreement of alpha = 0.72 for argument components and alpha = 0.81 for argumentative relations. In total, the corpus consists of 90 essays containing 1673 sentences.

3.2. Model Presentations

This model has been introduced by [23]. It is composed of two modules. In the initial module, the sentence undergoes tokenization using the BERT tokenizer, and subsequently, the BERT model is fine-tuned for token classification. This fine-tuning process aligns the output of the last layer with the specific number of classes in the dataset. Moving on to the second module, a linear chain Conditional Random Field [25] is employed to calculate the probability associated with each label class derived from BERT. The main idea of this model is to leverage the BERT attention knowledge and then to improve the results by incorporating a linear chain dependency structure. This approach leverages the interdependence between neighbouring words, capitalizing on their dependency relations. The favourable outcomes achieved by this model prompted us to adopt it as a robust benchmark for evaluating our approach, which is founded on constituency trees as input representations for sentences.

3.3. Results and Limitations Presentations

In Table 1, we observe variations in model performance across different datasets. When analysing the results, the important aspect is the difference between the F1-score of the separate and mutual models rather than their absolute values. Indeed, depending on the dataset, CDCP and UKP are strongly unbalanced in favor of the presence of argument spans. The 'Mutual Model' task exhibits lower performance on the UKP dataset, performs comparably on AURC and CDCP datasets, and demonstrates notably improved performance on ARG2020.

Table 1. F1-score of the different models at token level on the different test datasets. The "Separate Models" category refers to separate training models and "Mutual Model" refers tomerging the train datasets to train a single model.

Task	**Model Mutualisation**	**Model**	**Dataset**			
			AURC	CDCP	ARG2020	UKP
Binary Classification	SeparateModels	BERT	77.7%	99.7%	75.7%	91.8%
		BERT-CRF	77.7%	99.7%	76.5%	92.9%
	Mutual Model	BERT	76.7%	99.4%	92.1%	76.2%
		BERT-CRF	76.6%	99.5%	91.9%	72.1%

By tracing the origins of these datasets, we can attribute the improvement of ARG2020 to the fact that ARG2020 was originally based on an annotation scheme similar to UKP (as explained in [22]). Despite variations in literary genres and annotators, the consolidation of these datasets brings much more data on which to train and thus has significantly enhanced results on the ARG2020 dataset.

Therefore, the objective of this paper is to employ a common model for span identification and subsequently utilize task-specific models for labelling each span. Before delving into these experiments, it is important to establish a metric that can distinguish errors stemming from the initial span identification model from those arising in the subsequent labelling model. We introduce this new metric in Section 4.

4. INTRODUCING THE MINERR METRIC

This section aims to introduce a novel approach for dissecting the prediction errors in an argument component labelling task, discerning between errors resulting from misidentification of the span and those stemming from incorrect label assignment. We begin by formalizing the problem and subsequently explore two methods, one involving constraint considerations and the other without.

If we dissociate the tasks of span identification from label identification, it is important to note that the accuracy of the labelling identification model cannot surpass the relative error introduced by the span segmentation model. We must determine the optimal match for each segment in the predicted span list (the results from the span identification model) with a corresponding segment in the reference annotated list. This matching process ensures that when two segments share the same label, we achieve the highest possible precision for our label prediction model.

4.1. Problem formalization

We first introduce some definitions required to the introduction of more complex concepts after that.

Definition 1

*Let P be a sentence composed of $(n + 1)$ tokens indexed from 0 to n. A **segmentation** of the sentence P is a strictly increasing sequence of positive integers starting with 0 and ending with n. A **segment** of the sentence P associated with a segmentation $A = \{ _0 = 0, a_1, \dots, a_j = n + 1\}$ is a pair $s_i = (a_i, a_{\{i+1\}})$ of consecutive elements of the segmentation A. We call $S^A = \{(a_0, a_1), \dots, (a_{\{}j-1\}, a_j\}$ the ordered sequence of segments of A.*

Definition 2

*Let P be a sentence of $(n + 1)$ tokens. Let A be a segmentation of P composed of $(j + 1)$ elements. We call a **labelling** of P a sequence $L = ([i])_{\in[0,n]}$ which associates a label to each token of P. We also define a labelling of P associated to the segmentation A a sequence of labels which associates a label to each segment of A: $L^A = (P^A)$. We also note f^A the set of all labelling of P associated to the segmentation A.*

i $i\in[1,j]$

Using the notation of Definition 2, a labelling of P associated to the segmentation A naturally induces a labelling of P, and without ambiguity we can use the notation $L^A[i]$ for $i \in [0, n]$. Figure1 illustrates the concept defined above in Definition 1 and 2.

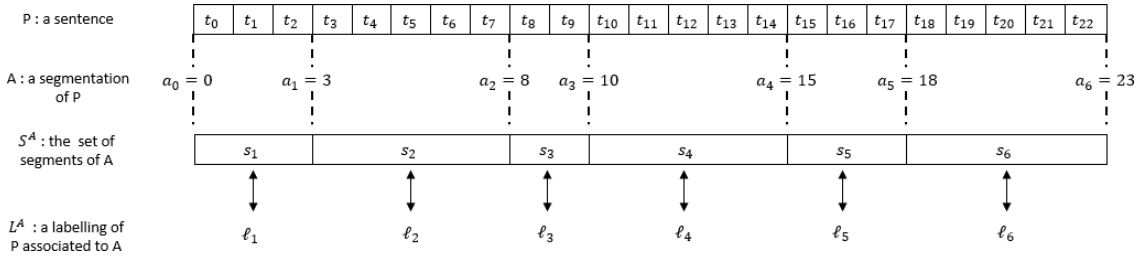

Figure 1 Illustration of concept of Sentence, Segmentation, Set of segments and Labelling. Inthis example, with the notation given in Definition 1 and 2, n = 22 and j = 6.

Introduction to the Errors

In this subsection, we introduce two distinct errors: a labelling error and a residual error.

Definition 3

*Let $\boldsymbol{P}$ be a sentence of $(\boldsymbol{n} + \boldsymbol{1})$ tokens and $\boldsymbol{A}$, $\boldsymbol{B}$ two segmentations of $\boldsymbol{P}$ with $\boldsymbol{L^A}$ and $\boldsymbol{L^B}$ two of their respective labelling. We define the **Labelling Error** (LabErr) between $\boldsymbol{L^A}$ and $\boldsymbol{L^B}$ as the percentage of tokens of $\boldsymbol{P}$ being labelled differently between $\boldsymbol{L^A}$ and $\boldsymbol{L^B}$:*

$$Lab(L^A, L^B) = Card\{\, i \in [0, n] : L^A[i] = L^B[i]\} \cdot \frac{100}{n+1}$$

This measure of the labelling error amounts to compute the accuracy when L^A is the reference value and L^B is the value predicted by a learning model.

We now define the minimal labelling error between the labelling sequences.

Definition 4
Let $\boldsymbol{P}$ *be a sentence and* $\boldsymbol{A}$, $\boldsymbol{B}$ *two segmentations of* $\boldsymbol{P}$. *Let* $\boldsymbol{L^A}$ *be a labelling associated with* $\boldsymbol{A}$. *We define the* ***Minimal Labelling Error*** *(MinLabErr) between* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *as the labelling error associated with the labelling* $\boldsymbol{L^B}$ *of* $\boldsymbol{B}$ *which minimises the labelling error between* $\boldsymbol{L^A}$ *and* $\boldsymbol{L^B}$*:*

$$MinLab(A, B, L^A) = \min_{L^B \in f^B} LabErr(L^A, L^B)$$

As this definition is not symmetric regarding to $\boldsymbol{A}$ *and* $\boldsymbol{B}$, *we will call* $(\boldsymbol{A}, \boldsymbol{L^A})$ *the* ***references*** *and* $(\boldsymbol{B}, \boldsymbol{L^B})$ *the* ***predictions***.

At first glance, MinLabErr represents the error we aim to minimize in order to achieve the best possible results. However, upon closer examination, it exhibits certain shortcomings, especially when the number of distinct labels is low. In fact, when conducting classification tasks with onlya few labels, such as "premise" and "conclusion," we still want our model to preserve the underlying structure and penalize errors if it misidentifies a segment as a conclusion but not for the correct argument.

To address this issue, we introduce a second error by treating all labels associated with a labelling as distinct.

Residual difference between two segments
We first introduce a metric to compute the residual difference between two segments of two distinct segmentations.

Definition 5

Let $\boldsymbol{P}$ *be a sentence and* $\boldsymbol{A}$, $\boldsymbol{B}$ *be two segmentations of* $\boldsymbol{P}$. *Let* $\boldsymbol{a} = (\boldsymbol{a_i}, \boldsymbol{a_{i+1}})$ *be asegment of* $\boldsymbol{A}$ *and* $\boldsymbol{b} = (\boldsymbol{b_j}, \boldsymbol{b_{j+1}})$ *be a segment of* $\boldsymbol{B}$. *We define the* ***Residual Difference*** *(ResDiff) between* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *as follows:*

$$\boldsymbol{ResDiff(a, b) = \max(a_i - b_i, 0) + \max(b_{i+1} - a_{i+1}, 0)}$$

We illustrate this definition below on a concrete example.

Illustrating example Let P be a sentence and A and B be two segmentations. We suppose that A is the reference and B the prediction. Let a_3 be the third segment of A and b_2 the second segment of B. Let us compute the residual difference between a_3 and b_2 in various situations:

- **Case 1**: $a_3 = [3,8]$ and $b_2 = [2,8]$. The segment b_2 starts before a_3. The error is equalto $3 - 2 = 1$.
- **Case 2**: $a_3 = [3,8]$ and $b_2 = [3,10]$. The segment b_2 ends after a_3. The error is equal to $10 - 8 = 2$.
- **Case 3**: $a_3 = [3,8]$ and $b_2 = [2,9]$. The segment a_3 is included in segment b_2. The erroris equal to $(3 - 2) + (9 - 8) = 2$.
- **Case 4**: $a_3 = [3,8]$ and $b_2 = [8,10]$. The two segments have no tokens in common. The

error is maximal and is equal to the size of the segment b_2 which is 2.

- **Case 5**: $a_3 = [3,8]$ and $b_2 = [4,7]$. There is no error as the segment b_2 is included in segment a_3.

An illustration of the different cases can be found in Figure 2.

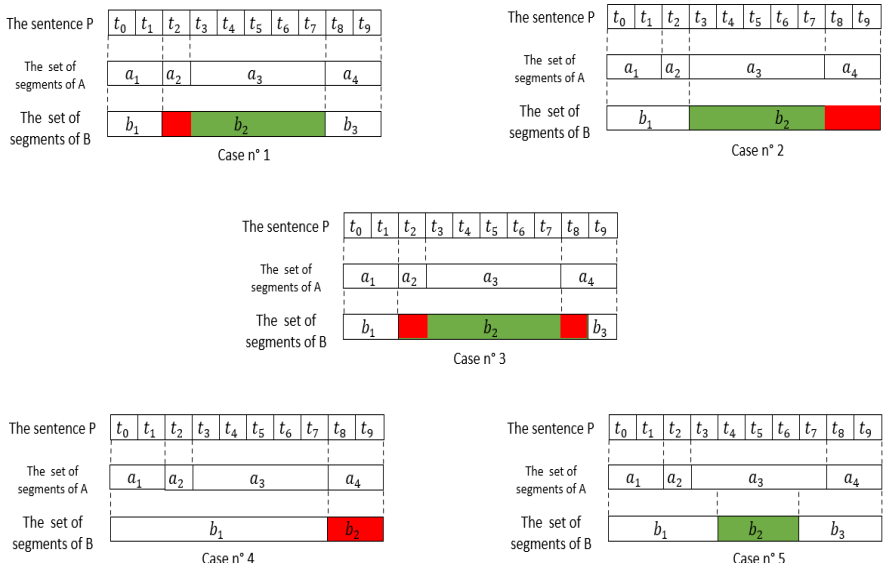

Figure 2: Visualizing Sub-Cases: Examining Residual Differences Between Segments $\boldsymbol{b_2}$ and $\boldsymbol{a_3}$. The highlighted red tokens represent the constituents of the residual difference.

We can now introduce the Minimal Error between two segmentations, which is at the core of our paper.

Definition 6

Let P be a sentence and A, B be two segmentations of P. We defined the Minimal Error between A and B as follows:

$$MIN(A,B) = \sum_{\substack{a \in S^A \\ b \in S^B}} \min ResDiff(a,b)$$

This definition remains unaffected by any labelling of the reference segmentation. Furthermore, it can be seen easily that, when we enforce the condition that the labels of L^A are two-by-two distinct, the two errors MINERR and *MinLabErr* coincide, as stated in the next Lemma.

Lemma 1

Let P be a sentence and A, B be two segmentations of P. Let $L^A = (P^A_i)_{i \in [1,j]}$ be a labelling of P associated to A. Assume moreover that $\forall i \neq k$, $P^A_i \neq P^A_k$. Then we have:

$$MINERR(A,B) = MinLabErr(A,B,L^A).$$

4.1. Computation of MINERR

4.4.1 Computation of MINERR with no constraints

Consider a sentence P with two segmentations, A and B. The MINERR computation method entails identifying, for each segment b in B, the corresponding segment a in A that minimizes the residual difference between them. A comprehensive pseudo-code description of this algorithm is provided in Algorithm 1.

Algorithm 1. Compute MINERR(A,B).

Require: The reference segmentation A and the prediction segmentation B.
  MINERR $= 0$
  for every segment $b = (b_i, b_{i+1})$ in B **do**
    $\text{Error}_b = n + 1$
    for every segment $a = (a_i, a_{i+1})$ in A_2 **do**
      $\text{Error}_b = \min(\text{Error}_b, \text{ResDiff}(a, b))$ ▷ We preserve the minimal residual difference.
    end for
    MINERR$+ = \text{Error}_b$ ▷ We add to the total sum the residual difference associated to b.
  end for
  return MINERR

While this method enables the computing of MINERR, it underscores a significant limitation inherent in MINERR itself. Specifically, it does not consider the number of segments within the reference segmentation. Failing to impose constraints on structural preservation means we miss out on valuable insights into segment relationships and overall structure. To address this challenge, we propose to add constraints in the definition of MINERR.

4.4.2 Computation of MINERR under constraints

The fundamental concept behind this novel metric is to incorporate structural constraints in the definition of MINERR we introduce the concept of 'MINimal ERRor under Constraints' to formally define this enhancement.

Definition 7
Let P be a sentence and let (A, L^A) *be the reference, assuming that labels in* L^A *are two- by-two distinct. Let B be a segmentation of P, called the prediction. We introduce the concept of "MINimal ERRor under Constraints" (MINERR*$_C$*) between A and B, which quantifies the labelling error associated with the labelling* L^B *of B. This new metric minimizes the labelling error between* L^A *and* L^B *while imposing the constraint that the labels in* L^B *are pairwise distinct.*

$$MINE(A, B, L^A) = \min_{L^B \in f^B} LabErr(L^A, L^B, A)$$

Subject to the constraint: $\forall i \neq k, P_i^B \neq P_k^B$.

Note that, the requirement for distinct labels is not mandatory, but it simplifies the definition of MINERR$_C$ as it allows us to express it in terms of labelling sequences (an alternative definition of MINERR$_C$ could be formulated without imposing the distinct label condition).

We will tackle the challenge of computing MINERR$_C$ with a recursive algorithm, whose steps are outlined below.

- **Basic association**

  - If both segmentations have an equal number of segments, the only valid labelling preserving the structure consists in associating to each segment of A the corresponding segment of B in a sequential manner (i.e. $\forall i \in [1, j]$, $a_i = b_i$).
  - Otherwise, we return a value of 0.

- **Recursive Case**

  - If A is longer than B, multiple choices exist for selecting which segment of B corresponds to each element of A. To find the most efficient way to compute these possibilities, we proceed by dichotomy. We apply the condition that the median segment of B must be chosen among the segments of A in such a way that the number of segments to its left (and right) is greater or equal to the numberof segments to its left (and right) in A. We then compute the error for the left and right segments recursively, independently. Finally, we choose the solution that minimizes the sum of the errors in the left and right parts, along with the ResidualDifference of the median value.

  - If B is longer than A, some segments in B will necessarily be associated to the same segment in A. By Definition 7, finding a solution for MINERR$_C$ in such cases is not feasible. In practice, we will relax the problem by assigning the samesegment in A to two consecutive segments in B and then continue our algorithm. Therefore, we need to determine the best pair of segments in B in terms of the Residual Difference cost, and then proceed with the remaining segmentations recursively.

The complete algorithm is detailed in pseudo-code in Algorithm 2 and 3.

Algorithm 2. Compute $\boldsymbol{MINE}(\boldsymbol{A}, \boldsymbol{B})$.

Require: two segmentations A, B
  if size(A) == size(B) **then** ▷ Basic association no. 1
    return $basic\_association(A, B)$
  end if
  if size(A) == 0 or size(B) == 0 **then** ▷ Basic association no. 2
    return 0
  end if
  $\text{MINERR}_C = n + 1$
  if $size(A) > size(B)$ **then** ▷ Recursive Case no. 1
    $differenceSize = size(A) - size(B)$
    $indiceMedian = \lfloor \frac{size(B)}{2} \rfloor$
    $B_{left} = B[: indiceMedian]$
    $B_{right} = B[indiceMedian + 1 :]$
    for i in range(differenceSize) **do**
      $\text{ResDiff\_}median = \text{ResDiff}(A[(indiceMedian + i)], B[indiceMedian])$
      $A_{left} = A[: (indiceMedian + i)]$
      $A_{right} = A[(indiceMedian + i + 1) :]$
      $cut\_\text{MINERR}_C = rec\_\text{MINERR}_C(A_{left}, B_{left}) + rec\_(A_{right}, B_{right})$
      $\text{MINERR}_C = min(cut\_\text{MINERR}_C + \text{ResDiff\_}median, \text{MINERR}_C)$
    end for
  end if
  if $size(A) < size(B)$ **then** ▷ Recursive Case no. 2
    $differenceSize = size(B) - size(A)$
    for i in range(size(B)-2) **do**
      $B_{left} = B[: i]$
      $B_{right} = B[i + 2 :]$
      for k in range(size(A)) **do**
        $\text{ResDiff\_}pair = \text{ResDiff}(B[i] + B[i + 1], A[k])$
        $A_{left} = A[: k]$
        $A_{right} = A[(k + 1) :]$
        $cut\_\text{MINERR}_C = rec\_\text{MINERR}_C(B_{left}, A_{left}) + rec\_\text{MINERR}_C(B_{right}, A_{right})$
        $\text{MINERR}_C = min(cut\_\text{MINERR}_C + \text{ResDiff\_}pair, \text{MINERR}_C)$
      end for
    end for
  end if
  return MINERR_C

Algorithm 3. Presentation of the function ***basic_association***.

Require: two segmentations A, B with the same number of segments.
  $difference = 0$
  for every segment a in A **do**
    $index\_a$ the index of a in A
    $difference+ = \text{ResDiff}(a, B[index\_a])$
  end for
  return $difference$

5. EXPERIMENTATION OF USING MINERR

In this section, we conduct a performance comparison of the metrics introduced in Section 4. We first presents the models training and then analyse their results.

5.1. Model Presentation

In this subsection, we present the training strategy employed for our metrics, denoted as MINERR and MINERRC respectively in Table 2. Our approach consists of a multi-stage process designed to optimize performance.

- **Step 1**: Span Identification Model Training: In this initial step, we trained a span identification model using four distinct datasets. Based on the results in Section 3, we have chosen to employ a BERT+CRF model, as detailed in Subsection 3.2.

- **Step 2**: Label Identification Model Training:

a Token-Level Label Estimation with BERT: Initially, we employed a BERT model to estimate the label for each individual token.

b Span-Level Label Normalization. We standardized label predictions to correspond to the span identified by the model in Step 1.

c Loss Calculation and MINERR Incorporation: We computed the Cross-Entropy loss associated with these predictions and added MINERR (or MINERR_C) from it. This training methodology underpins the development and refinement of our MINERR models, resulting in enhanced performance and accuracy in label identification tasks.

Results and Limitations

Table 2. F1-score of the different models at token level on the different test datasets. The "SeparateModels" category refers to separate training models and "Mutual Model" refers to merging the traindatasets to train a single model.

Model	**Dataset**			
	AURC	CDCP	ARG2020	UKP
Baseline	68%	80%	75%	81%
MINERR Loss	68.5%	81.1%	79.3%	76.8%
MINERRC Loss	67.2%	78.4%	76.7%	72.1%

The results are summarized in Table 2. It is noteworthy that, overall, we observe superior performance when employing the MINERR approach, while slightly diminished results are obtained when utilizing MINERRC. This discrepancy can be attributed to the introduction of constraints, which enhance the global coherence of the predictions but are not taken into account in the F1-score.

Moreover, a significant challenge encountered in the binary classification model used as first step in Subsection 4.2 pertains to situations where two argument spans appear consecutively within a sentence. In such instances, distinguishing whether these spans correspond to one or two arguments becomes a complex task, potentially leading to errors, especially in the second model. Nevertheless, it is worth highlighting that this issue might be ameliorated by replacing the binary classification model with a more intricate span identification model.

6. CONCLUSION

In this paper, we have introduced two novel metrics MINERR and MINERRC designed to enhance transfer learning across datasets with slightly different annotation schemas. During the training of our models, these metrics have enabled us to achieve improved results, especially in scenarios where the initial datasets were relatively small. Additionally, these newly proposed metrics have contributed to achieving a more consistent final result in alignment with the inherent structure of the constituent elements of the sentence.

While this approach shows promise, there are still areas for further improvement that we will discuss in detail below.

- To address potential concerns regarding the complexity of the recursive method, we can employ a "memoization" technique. Memoization involves optimizing recursive algorithms by storing the results of costly function calls and then returning the cached result when the same inputs are encountered again. This technique can lead to a substantial improvement in the efficiency of recursive algorithms, especially whendealing with overlapping subproblems. It can be particularly valuable for enhancing the computational performance of the MINERRC algorithm, making it more practical for real-world applications.

- To reduce the gap of performance between MINERR and MINERRC, we can create a family of metrics denoted as MINERRC(k) (with $k < n$ where n is the number of tokens in the sentence P), as a compromise between the MINERR and MINERRC approaches. The idea is as follows. After several recursive iterations of the MINERRC algorithm, when the difference in the sizes of A and B becomes less than k, we loosen the constraintsand compute the metric using the MINERR approach. This approach allows for a flexible trade-off between the two metrics, tailoring the level of constraint to the specific characteristics of the segmentation problem at hand.

To conclude, we believe that these techniques can also find application in various other domains of artificial intelligence, particularly in situations where data labelling and grouping are essential. For instance, they can be applied in tasks such as detecting objects in image.

REFERENCES

[1] J. Lawrence et C. Reed, « Argument Mining: A Survey », Computational Linguistics, vol. 45, no 4, p. 765-818, janv. 2020, doi: 10.1162/coli_a_00364.

[2] R. M. Palau et M.-F. Moens, « Argumentation mining: the detection, classification and structure of arguments in text », in Proceedings of the 12th International Conference on Artificial Intelligence and Law, in ICAIL '09. New York, NY, USA: Association for Computing Machinery, juin 2009, p. 98-107. doi: 10.1145/1568234.1568246.
[3] R. Baumann, G. Wiedemann, M. Heinrich, A. D. Hakimi, et G. Heyer, « The Road Map to FAME: A Framework for Mining and Formal Evaluation of Arguments », Datenbank Spektrum, vol. 20, no 2, p. 107-113, juill. 2020, doi: 10.1007/s13222-020-00343-x.
[4] M. Taboada et W. C. Mann, « Rhetorical Structure Theory: looking back and moving ahead »,
[5] Discourse Studies, vol. 8, no 3, p. 423-459, juin 2006, doi: 10.1177/1461445606061881.
[6] M. Stede, S. Afantenos, A. Peldszus, N. Asher, et J. Perret, « Parallel Discourse Annotations on a Corpus of Short Texts », in Proceedings of the Tenth International Conference on Language
[7] Resources and Evaluation (LREC'16), Portorož, Slovenia: European Language Resources Association (ELRA), mai 2016, p. 1051-1058. Consulté le: 6 avril 2022. [En ligne]. Disponible sur: https://aclanthology.org/L16-1167
[8] W. Mann et S. Thompson, Rhetorical Structure Theory: A Theory of Text Organization. 1987.
[9] R. E. Longacre, The Grammar of Discourse. Boston, MA: Springer US, 1995. doi: 10.1007/978-1-4615-8018-8.
[10] S. E. Toulmin, The Uses of Argument, 2e éd. Cambridge: Cambridge University Press, 2003. doi: 10.1017/CBO9780511840005.
[11] D. Marcu, « A Decision-Based Approach to Rhetorical Parsing », in Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, College Park, Maryland, USA: Association for Computational Linguistics, juin 1999, p. 365-372. doi: 10.3115/1034678.1034736.
[12] C. Braud, M. Coavoux, et A. Søgaard, « Cross-lingual RST Discourse Parsing », in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain: Association for Computational Linguistics, avr. 2017, p. 292-304. Consulté le: 18 janvier 2023. [En ligne]. Disponible sur: https://aclanthology.org/E17-1028
[13] Y. Jo, J. Visser, C. Reed, et E. Hovy, « A Cascade Model for Proposition Extraction in Argumentation », in Proceedings of the 6th Workshop on Argument Mining, Florence, Italy: Association for Computational Linguistics, août 2019, p. 11-24. doi: 10.18653/v1/W19-4502.
[14] Y. Ajjour, W.-F. Chen, J. Kiesel, H. Wachsmuth, et B. Stein, « Unit Segmentation of Argumentative Texts », in Proceedings of the 4th Workshop on Argument Mining, Copenhagen, Denmark: Association for Computational Linguistics, sept. 2017, p. 118-128. doi: 10.18653/v1/W17-5115.
[15] C. M. I. M. Matthiessen et S. A. Thompson, « The structure of discourse and 'subordination' », in Typological Studies in Language, vol. 18, J. Haiman et S. A. Thompson, Éd., Amsterdam: John Benjamins Publishing Company, 1988, p. 275. doi: 10.1075/tsl.18.12mat.
[16] C. Stab et I. Gurevych, « Annotating Argument Components and Relations in Persuasive Essays », in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland: Dublin City University and Association for Computational Linguistics, août 2014, p. 1501-1510. Consulté le: 6 juillet 2023. [En ligne]. Disponible sur: https://aclanthology.org/C14-1142
[17] K. Papineni, S. Roukos, T. Ward, et W.-J. Zhu, « Bleu: a Method for Automatic Evaluation of Machine Translation », in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, juill. 2002, p. 311-318. doi: 10.3115/1073083.1073135.
[18] G. Doddington, « Automatic evaluation of machine translation quality using n-gram co-occurrence statistics », in Proceedings of the second international conference on Human Language Technology Research -, San Diego, California: Association for Computational Linguistics, 2002, p. 138. doi: 10.3115/1289189.1289273.
[19] C.-Y. Lin, « ROUGE: A Package for Automatic Evaluation of Summaries », in Text Summarization Branches Out, Barcelona, Spain: Association for Computational Linguistics, juill. 2004, p. 74-81.
[20] Consulté le: 12 septembre 2023. [En ligne]. Disponible sur: https://aclanthology.org/W04-1013
[21] S. Banerjee et A. Lavie, « METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments », in Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor,
[22] Michigan: Association for Computational Linguistics, juin 2005, p. 65-72. Consulté le: 25 septembre 2023. [En ligne]. Disponible sur: https://aclanthology.org/W05-0909
[23] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, et J. Makhoul, « A Study of Translation Edit Rate with

Targeted Human Annotation », in Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, août 2006, p. 223-231. Consulté le: 12 septembre 2023. [En ligne]. Disponible sur: https://aclanthology.org/2006.amta-papers.25

[24] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, et Y. Artzi, « BERTScore: Evaluating Text Generation with BERT ». arXiv, 24 février 2020. Consulté le: 25 septembre 2023. [En ligne].Disponible sur: http://arxiv.org/abs/1904.09675

A. L. F. Han, D. F. Wong, et L. S. Chao, « LEPOR: A Robust Evaluation Metric for Machine Translation with Augmented Factors », in Proceedings of COLING 2012: Posters, Mumbai, India: The COLING 2012 Organizing Committee, déc. 2012, p. 441-450. Consulté le: 12 septembre 2023. [En ligne]. Disponible sur: https://aclanthology.org/C12-2044

[25] T. Alhindi et D. Ghosh, « “Sharks are not the threat humans are”: Argument Component Segmentation in School Student Essays », in Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, Online: Association for Computational Linguistics, avr. 2021, p. 210-222. Consulté le: 12 juin 2023. [En ligne]. Disponible sur: https://aclanthology.org/2021.bea-1.22

[26] D. Trautmann, J. Daxenberger, C. Stab, H. Schütze, et I. Gurevych, « Fine-Grained Argument Unit Recognition and Classification », Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no 05, Art. no 05, avr. 2020, doi: 10.1609/aaai.v34i05.6438.

[27] J. Park et C. Cardie, « 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), mai 2018. Consulté le: 27 octobre 2021. [En ligne]. Disponible sur: https://aclanthology.org/L18-1257

[28] J. D. Lafferty, A. McCallum, et F. C. N. Pereira, « Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data », in Proceedings of the Eighteenth International Conference on Machine Learning, in ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., juin 2001, p. 282-289.

AUTHORS

Samuel Guilluy is a final-year Ph.D. candidate at the Mathematical Laboratory of Rennes (IRMAR). His research focuses on the field of argument mining, where he harnesses the power of machine learning to analyse and extract valuable insights from low-resource datasets.

Florian Méhats is a professor of applied mathematics, on leave from the University of Rennes (France) and member of the Rennes Mathematics Laboratory (IRMAR), where his activities focus on mathematical modelling of physics, numerical analysis, PDEs and machine learning.He is currently a research associate atRavel Technology, specializing in homomorphic cryptography.

Billal Chouli is the Head of AI & Blockchain atHeadMind Partners.