Few-shot Learning for Argumentation in the Medical Domain

Author: Jon Manzanal Martín

Advisors: Rodrigo Agerri

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Final Thesis

February 2023

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Laburpena

Azkenaldian, arlo medikoan arreta handiagoa jarri da Adimen Artifizialarekin lotutako tekniketan, medikuei galderak errazago eta azkarrago ebazten laguntzeko. Hori bereziki garrantzitsua da Ebidentzian Oinarritutako Medikuntzaren arloan, medikuek egituratu gabeko informazio asko erabili behar baitute erabakiak garaiz hartu ahal izateko. Testuinguru horretan, Argumentu-Meatzaritzak lagundu egiten du argudio-osagaiak eta haien arteko harremanak identifikatzen, deliberazio-prozesuak eta azalpen medikoak dituzten testuetan.

Argumentu-Meatzaritzari buruzko lanen corpus nahiko ona dagoen arren, datu-multzo gehienak ingeleserako garatu dira, eta gaur egun bat bakarrik dago eremu medikorako. Eskura ditugun datu idatzien falta hori dela eta, tesi honetan prompting eta fine-tuning teknikak aztertuko ditugu, few-shot ingurune batean ingelesa ez den beste hizkuntza baterako eremu medikoan argumentu-meatzaritza egiteko estrategiarik onena ezartzeko. Gure emaitzek enpirikoki frogatzen dute few-shot prompting bidez sekuentziak etiketatzeko metodoak oso sentikorrak direla entrenamendu-datuak sortzeko erabilitako laginketa-metodoarekiko. Izan ere, eta argitaratutakoaren kontra, datuen laginketa alternatibo baten ondorioz, fine-tuning metodoek few-shot ebaluatzeko inguruneetako prompting teknikak gainditzen dituzte. Zehatzago esanda, arlo medikoan Argumentu-Meatzaritzarako entrenamendu-datuen %40 nahikoa da state-of-the-arten emaitzak lortzeko. Gainera, entrenamendu-datuen %10-20 soilik erabiltzeak (hau da, pertsona bakoitzak 15 orduz eskuz etiketatuta lan egiteak) oso errendimendulehiakorra lortzeko aukera ematen du.

Abstract

In recent times, in the medical field, more attention has been paid to techniques related to Artificial Intelligence to support doctors to solve questions in a simpler and faster way. This is particularly relevant in the field of Evidence-based Medicine, since doctors need to deal with a lot of unstructured information to be able to take timely decisions. In this context, Argument Mining helps to identify argumentative components and the relations between them in texts containing medical deliberation and explanatory processes. Although there is a relatively good body of work on Argument Mining, the large majority of datasets have been developed for English, and only one currently exists for the medical domain. Due to this lack of available annotated data, in this thesis we explore prompting and fine-tuning techniques to establish the best strategy to perform argument mining in the medical domain for a target language different to English in a few-shot setting. Our results empirically demonstrate that few-shot prompting approaches for sequence labelling are highly sensitive to the sampling method used to generate the training data. In fact, and contrary to published work, we show that an alternative data sampling results in fine-tuning methods outperforming prompting techniques in few-shot evaluation settings. More specifically, we establish that 40% of the training data for Argument Mining in the medical domain is enough to obtain state-of-the-art results. Furthermore, using just 10-20% of the training data (which amounts to 15 hours of manual labelling work per person) allows to obtain highly competitive performance.

Contents

List of Figures

List of Tables

1 Introduction

In a medical environment, one of the most important tasks is to be able to identify and diagnose a disease and then prescribe appropriate treatment for it, taking into account the patient's health and clinical history. However, deciding which treatment is the most appropriate can lead to several challenges. One of the challenges is correctly predicting the disease, as many diseases can produce similar symptoms, making detection difficult.

Thus, in recent times, in the medical field, more attention has been paid to techniques related to Artificial Intelligence to support doctors to solve questions in a simpler and faster way. This is particularly relevant in the field of Evidence-based Medicine, since doctors need to deal with a lot of unstructured information to be able to take timely decisions. In this context, Argument Mining helps to identify argumentative components and the relations between them in texts containing medical deliberation and explanatory processes.

In Figure 1 we can see an example of the Argument Mining task according to Stab and Gurevych (2014). This task consists of two subtasks: (i) identifying the argument components and, (ii) extracting the relations between them. In this approach there are two types of argument components, namely, *claims* and *premises*. Claims (in yellow) refer to the statements made in the text with respect to one topic whereas premises (in blue) refer to the evidence used in order to *support* or *attack* the claims. Thus, relation extraction amounts to extracting the *support* or *attack* relations between claims and premises. It should be noted that relations can link any kind of argument component.

Automatic Argument Mining techniques have recently been developed for different domains, such as education (Stab and Gurevych, 2014), news (Reed et al., 2008), law (Mochales and Ieven, 2009) or science and medicine (Mayer et al., 2021). However, most of the previously mentioned works only try to solve the problems in English, which means that there is an urgent need to generate annotated datasets for Argument Mining. This is particularly true of the medical domain, for which only one English dataset exists to learn Argument Mining models (Mayer et al., 2021).

Due to this lack of available annotated data, in this thesis we explore prompting and fine-tuning techniques to establish the best strategy to perform argument mining in the medical domain for a target language different to English in a few-shot setting. Few-shot learning is a type of machine learning where the model must perform a task with very few examples, often just one or a few. This is in contrast to traditional machine learning approaches, which require a large amount of labeled data in order to perform well. By using few-shot learning, we aim to reduce the amount of data required for training and make it possible to learn a wide range of tasks using as little data as possible.

One way to implement few-shot learning is through the use of prompts, which are short descriptions or examples that provide additional context or guidance to the model. Prompts can be used to help the model understand the task at hand and make more accurate predictions, even with very few examples.

We will explore the use of different systems, such as EntLM (Ma et al., 2022), to implement few-shot learning with prompts, and MMCV (Mayer et al., 2021) to implement

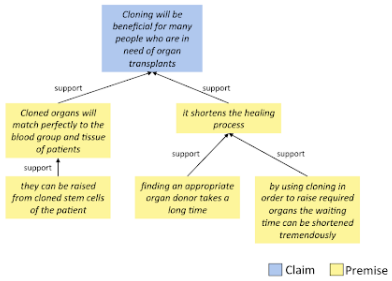

Figure 1: Argument Mining example (Stab and Gurevych, 2014).

few-shot via fine-tuning. We will also examine different methods for generating prompts, including using Natural Language Processing (NLP) techniques and Entity-Oriented Language Models.

Overall, our goal is to demonstrate the effectiveness of few-shot learning via prompting as a means of improving the performance of automatic argument mining approaches while reducing the need of manually labelled annotated data. This thesis provides the following contributions:

1. To the best of our knowledge, this thesis provides the first comprehensive study of few-shot prompting for argument mining (both argument component detection and relation extraction).

2. Our results empirically demonstrate that few-shot prompting approaches for sequence labelling are highly sensitive to the sampling method used to generate the training data.

3. In fact, and contrary to published work (Ma et al., 2022), we show that an alternative data sampling results in fine-tuning methods outperforming prompting techniques in few-shot evaluation settings.

4. We establish that 40% of the training data for Argument Mining in the medical domain is enough to obtain state-of-the-art results. Furthermore, using just 10% of the training data (which amounts to around 15 hours of manual labelling work per person) allows to obtain highly competitive performance.

5. Code and datasets are publicly available[1].

The structure of this thesis is the following: In the following section, we describe the most important previous work done in the field of argument mining. We also review the most important approaches to address the lack of manually labelled training data, including few-shot prompting, and cross-lingual approaches.

[1] `https://github.com/jonmanzanal/Sampling-FewShot-ArgumentMining`

Second, in Section 3 we provide a detailed description of the datasets and systems to perform the experiments. Then in Section 4 the data sampling methods used for few-shot experimentations are explained, focusing on the resulting dataset structure for each type of data sampling. We also discuss the way the label words work for prompting approaches. Finally, the hyperparameters used for experimentation are specified. In Section 5 we report the experimental results and quantitatively and qualitatively analyze them. An error analysis is provided in Section 6 and final conclusions and future work are discussed in Section 7.

2 Related Work

In this section, we will provide an overview of previous research and studies that are related to both argument mining and few-shot learning.

2.1 Argument mining

The field of argument mining, also known as argumentation analysis or argumentation extraction, has seen significant growth and development in recent years. A number of research efforts have focused on the identification and classification of arguments in text, as well as the development of techniques for automatically extracting argumentative structure from the text.

One early example of work in this area is Toulmin's (2003) model of argumentation, which provides a framework for identifying the components of an argument and analyzing their relations. This model has been widely adopted in the field of argumentation studies and has influenced the development of various argument mining techniques.

More recent efforts in argument mining have focused on the use of natural language processing (NLP) techniques, based on machine learning and deep learning, to automatically identify and classify argumentative components in text. For this objective, there are several tasks that try to accomplish it, such as argument component identification (Stab and Gurevych, 2016; Palau and Moens, 2009), Argument unit segmentation (Ajjour et al., 2017), and detecting the relations between arguments (Nguyen and Litman, 2016; Mayer et al., 2021).

In addition to these technical approaches, there has also been a growing interest in the application of argument mining to various domains. Here we list the most popular ones:

- Education: Stab and Gurevych (2014) developed an argument mining dataset composed of 90 persuasive essays in English.
- News: a corpus composed of English news from different newspapers from around the world (Reed et al., 2008).
- Law: Mochales and Ieven (2009) created a corpus composed of 45 judgments and decisions collected from CD recordings from August to December 2006. They releseased it both in English and French.
- Politics: Vivesdebate (Ruiz-Dolz et al., 2021) is a dataset composed of Catalan, Spanish, and English texts, where Catalan is the source language and English and Spanish have been automatically translated.
- Scientific/medical: the AbstRCT dataset includes claims and premises of clinical cases and their relations in English (Mayer et al., 2021). This is the only dataset annotated with argument components and their relations for the medical domain.

One of the main sources of training data for argument mining has been annotated datasets such as the PDTB (Pragmatic Discourse Treebank) by Prasad et al. (2017) and the Argument Reasoning Corpus (ARC) (Habernal et al., 2018). However, these datasets are limited in size and coverage and are primarily in English. This has led to a reliance on methods such as transfer learning (Pathak et al., 2022; Parthipan and Wischik, 2022), and data augmentation (Perez and Wang, 2017; Park et al., 2019), to improve the performance of argument mining systems in other languages and domains.

It can be seen that most of the corpora described above are for English, with a notable lack of corpora in other languages, such as Spanish for which only an automatically translated version of Vivesdebate exists. While some efforts have been made to create multilingual argumentation datasets, such as the Multilingual Argumentation Mining Corpus (MAMC) (Toledo-Ronen et al., 2020) and the Cross-Lingual Argumentation Corpus (CLAC) by Eger et al. (2018), they are still limited in size and language coverage. This presents a significant challenge for the development of argument mining systems that can operate in a wide range of languages and domains.

2.2 Cross-lingual transfer

To overcome the problem of missing data for a given target language, approaches such as cross-lingual transfer are used. Cross-lingual transfer refers to the transfer of knowledge learned from one language to another (García-Ferrero et al., 2022). In this work they study both model- and data-transfer crosslingual strategies to perform a sequence labelling task whenever no training data is available for the target language. The crosslingual model-transfer approach consists of leveraging multilingual language models such as multilingual BERT (Devlin et al., 2019) or XLM-RoBERTa (Conneau et al., 2020) to learn models in a source language and predict into a different one. The data-transfer approach from a cross-lingual perspective consists of translating gold-labelled text from the source into the target language and then, using automatic word alignments, project the labels from the source into the target language. This results in an automatically generated dataset in the target language that can be used for training a sequence labelling model. Furthermore, Yeginbergenova and Agerri (2023) show that the data-transfer approach is more effective than the model-transfer to automatically obtain labeled data for argument mining.

Apart from the model- and data-transfer approaches, there are more techniques that are being tested in recent years to overcome the lack of training data. However, instead of focusing on transferring knowledge, the focus is placed on measuring how much annotated data is actually necessary to obtain competitive results. One of the most relevant techniques nowadays on this research topic is few-shot prompting, described in the next section.

2.3 Few-shot Approaches

One of the methods to address the lack of data is to use few-shot, which is a type of technique that enables a model to learn and generalize to new tasks with only a small

amount of the training data. It is particularly useful in situations where there is a scarcity of labeled data or where collecting and labeling large amounts of data is impractical or too expensive. In few-shot learning, the model is presented with a few examples for each new task and is expected to learn from these examples in order to make predictions about unseen instances of the same task. Wang et al. (2021) reformulate a number of NLP text classification tasks as textual entailment. They then use this unified method to demonstrate that standard pre-trained language models are very effective few-shot learners. Another very interesting approach is from the paper by Brown et al. (2020), where it is shown that scaling up language models greatly improves task-agnostic, few-shot performance.

Another few-shot approach is to use prompting; few-shot prompting is a variation of few-shot learning where the model is provided with a "prompt" or a description of the new class in addition to a few examples. This additional information can help the model to better understand the characteristics of the new class and improve its ability to generalize from the few examples it has seen. The prompt can be in the form of natural language text or other forms of structured data, such as attributes of the objects in the class. The idea behind few-shot prompting is that by providing the model with additional information, it can learn to classify new instances more accurately and with fewer examples than with traditional fine-tuning methods.

A very popular approach is Pattern-Exploiting Training (PET) (Schick and Schütze, 2020a,b), a system which employs a semi-supervised training procedure based on cloze-style phrases to help language models understand a given task in low-resource settings. This is done by assigning soft labels to a large set of unlabeled examples. This approach is shown to significantly outperform regular supervised training and various semi-supervised baselines. Furthermore, it also has a version called ADAPET (Tam et al., 2021) which modifies the objective to provide denser supervision during fine-tuning in few-shot learning without any unlabeled data.

2.3.1 Sequence labeling

Sequence labeling is a task in natural language processing where the goal is to assign a label to each word or token in a given sequence. This can be done for a variety of applications, such as named entity recognition, part-of-speech tagging, or argument component detection. For example, in named entity recognition, the task is to identify and classify named entities, such as people, organizations, and locations, in a given text, and assign a label to each word or token that corresponds to the entity it belongs to. This task can be approached using a variety of machine learning algorithms, such as hidden Markov models, conditional random fields, and deep neural networks. One of the best-performing approaches for argument mining is the one obtained in Mayer et al. (2018), obtaining an F1 of 82.36 by using the mBERT model and applying both a GRU and a CRF.

NNShot and StructShot (Yang and Katiyar, 2020) propose two metrics to perform few-shot. In NNShot the aim is to increase the nearest classifier to make better predictions. With respect to StructShot, a Viterbi algorithm is applied at the time of decoding. Another interesting few-shot prompting approach is that of Template NER (Cui et al., 2021), where

templates are used to predict the labels. By constructing a template for each class, it queries each span with each class separately. The result for each request is obtained by calculating the generalized probability of the request for a pre-trained LM.

In any case, it should be considered that prompting techniques based on templates for sequence labeling is rather problematic. Given a sequence of type $X = \{x_1, ..., x_n\}$, we have to achieve a sequence of labels with the same number of elements $Y = \{y_1, ..., x_n\}$. In addition, a new slot [S] is added in order to fill the first token or a succession of spans, where it starts at x_i and ends at x_j. We could construct a simple template, like "[X] [S] is a [Z] entity", where an entity label (e.g., Location) must be predicted in [Z] by the LM. The problem of this approach is that during decoding, obtaining the labels requires enumerating the spans along the whole sentence, which is a time-consuming process that increases each time the sentence gets longer. Therefore, while template-based prompting is useful for text classification tasks, other proposals have appear trying to avoid this cumbersome repetition of templates. Current recent state of the art approaches are Huang et al. (2021) and EntLM (Ma et al., 2022), which avoid using templates obtaining state-of-the-art results at a much lower computational cost.

2.3.2 Relation extraction

Relation extraction is the task of extracting structured relations from unstructured text data. It is a type of natural language processing (NLP) task that involves identifying and extracting relations between sequences in a given text. These relations can be between sequences consisting of people, organizations, events or argument components, for example. Relation extraction typically involves two main steps:

1. Sequence labeling: This step involves identifying and labeling the main sequences that are mentioned in the text. For example, if the text mentions "John Smith" and "Apple Inc.", then "John Smith" would be identified as a person entity, and "Apple Inc." would be identified as an organization entity.

2. Extracting the relations between the entities: Once the entities have been identified, the next step is to extract the relations between them. For example, if the text mentions that "John Smith works for Apple Inc.", then the relations between "John Smith" and "Apple Inc." would be "works for".

In relation to extraction there are several types of relations that can be extracted, including:

- Nominal relations: These are relations between noun phrases, such as "John is the father of Mary" where the relation is "father of" and the entities are "John" and "Mary".

- Verbal relations: Relations between verb phrases, such as "John gave a book to Mary" where the relation is "gave to" and the entities are "John", "book", and "Mary".

- Coreference relations: Formed by entities that refer to the same real-world object, such as "John gave a book to Mary. She loved it." where the relation is "coreference" and the entities are "Mary" in the first sentence and "she" in the second sentence.
- Event-event relations: Relations between events, such as "John won the marathon. He set a new record." where the relation is "set" and the entities are "John" and "new record".
- Temporal relations: These are relations between events or entities and time expressions, such as "John was born in 1980" where the relation is "born in" and the entities are "John" and "1980".
- Modality relations: Consisting of relations between entities and modal verbs or adverbs, such as "John may be a teacher" where the relation is "may be" and the entities are "John" and "teacher".

These are some types of relations that can be extracted in relation extraction tasks, but there may be other types depending on the specific task or domain.

Relation extraction is an important task in NLP because it allows for the automatic extraction of structured information from unstructured text data. This information can then be used for various downstream tasks, such as information extraction, question answering, and document summarization. One of the proposals is PET (Schick and Schütze, 2020a,b), which consists of semi-supervised training that uses natural language patterns to adapt the inputs to predefined sentences. Other approach is PPT (Gu et al., 2022), which proposes to pre-train prompts by adding soft prompts into the pre-training stage to obtain a better initialization.

There have been several approaches in the field of relation extraction, including different types of methods based on supervised (Zelenko et al., 2002; Miwa and Bansal, 2016), semi-supervised (Chen et al., 2006; Hu et al., 2020), and distantly supervised approaches (Mintz et al., 2009; Han et al., 2018). However, the above methods have problems in detecting new types of relations if they arise in a real scenario. Therefore, there are also approaches that focus on learning relations without predefined types, including open RE (Etzioni et al., 2008; Gao et al., 2020) and continual relation learning (Obamuyide and Vlachos, 2019; Wu et al., 2021).

3 Methodology/Materials and Methods

In this section we introduce the systems and datasets used as well as the metrics applied to evaluate the systems' performance on the datasets.

3.1 Systems

All the systems work with transformers, which is a type of neural network architecture designed to process sequential data, such as natural language, more efficiently than recurrent neural networks (RNNs). The key innovation of the transformer architecture is the use of attention mechanisms to process the input data, which allows the model to parallelize the computation across the input sequence and to handle very long input sequences more efficiently than an RNN. The transformer architecture was introduced in the paper "Attention Is All You Need" (Vaswani et al., 2017) and has been successful in a variety of natural language processing tasks, including machine translation, language modeling, and text classification.

As mentioned above, the transformers are used in language models, which are trained to predict the likelihood of a sequence of words. It does this by learning the statistical patterns that are characteristic of a particular language or document collection. The basic idea is that, given a sequence of words, a language model assigns a probability to each possible word that might come next in the sequence. For example, if the language model is trained on a large corpus of text, it might predict that the word "the" is more likely to come after the word "cat" than the word "banana". In the following subsections we present the systems used in our experimentation.

3.1.1 MMCV

MMCV (Mayer et al., 2018) is a sequence labelling system that uses two bidirectional transformers in combination with other deep learning architectures such as LSTM, GRU, and CRF. This tool allows us to perform experiments both for the classification of argument components and for the detection of their relations. You can use any transformer found in the Huggingface library, so it allows you to perform experiments with different language models in a simple way. MMCV is trained by fine-tuning the model on the argument mining task.

3.1.2 BERT/mBERT

mBERT (Devlin et al., 2019) stands for multilingual BERT. BERT, or Bidirectional Encoder Representations from Transformers, is a type of language processing model developed by Google. It is a state-of-the-art method for natural language processing tasks such as language understanding, translation, and text summarization. mBERT is an extension of BERT that has been trained on a large amount of text data in 100 languages. mBERT is both used in the fine-tuning model of MMCV and in the prompting approach of EntLM, introducted next.

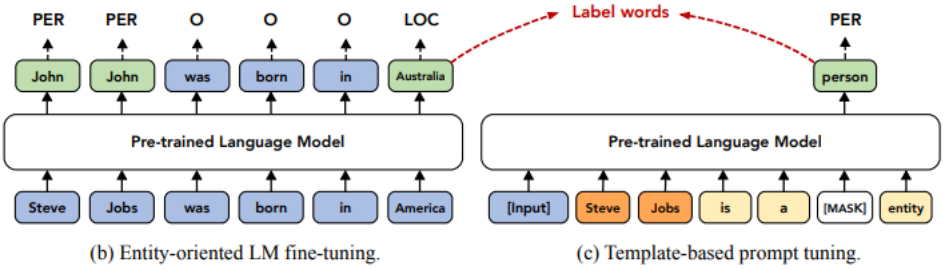

Figure 2: Difference between the operation of EntLM and a template-based model (Ma et al., 2022).

3.1.3 EntLM

EntLM (Ma et al., 2022) is a prompting based system for sequence labelling which does not uses any previously defined template. The system allows to use any transformer found in the Huggingface library, so it allows performing experiments with different transformers in a simple way. Instead of using a predefined template it works with a word label that works as a reference, generalizing the tokens for greater simplicity. For example, when fed with "[MASK] was born in America", the LM is pre-trained to predict some label word such as "John" at the position of the entity (e.g., "Obama") as an indication of the label "PER". While for the none-entity word "was", the LM remains to predict the original word. This is illustrated by Figure 2. EntLM obtains better results for sequence labeling few-shot prompting than other approaches, such as NNShot, StructShot, or Template NER.

3.1.4 PET

PET (Schick and Schütze, 2020a,b) is a prompting-based few-shot system based on providing a description of the task to be achieved with a template. It consists of semi-supervised training that uses natural language patterns to adapt the inputs to predefined sentences. PET works in three steps: First, for each pattern, a separate PLM is fine-tuned on a small training set $\mathcal{T}$. The ensemble of all models is then used to annotate a large unlabeled dataset $\mathcal{D}$ with soft labels. Finally, a standard classifier is trained on the soft-labeled dataset. There is also the option of iPET, an iterative variant of PET in which this process is repeated with increasing training set sizes.

Figure 3 illustrates the way that PET works. More specifically, it can be seen the process of creating patterns to convert training examples into cloze questions, fine-tuning a pretrained language model for each pattern, and using an ensemble of the models to annotate unlabeled data. A classifier is then trained on the annotated data.

3.2 Data

In this section, we will detail which dataset we have used to perform the experiments, as well as their main characteristics.

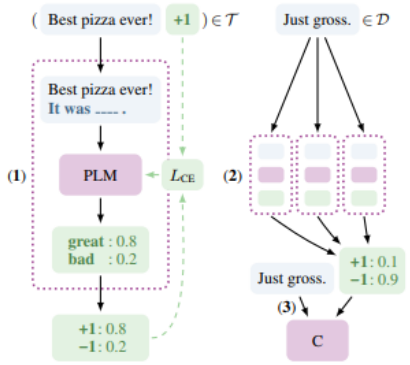

Figure 3: PET operation for sentiment classification task (Schick and Schütze, 2020a).

	Train	Dev	Test
O	67198	9124	16739
B-Claim	790	108	248
B-Premise	1644	218	438
I-Claim	15350	2087	4677
I-Premise	51839	6655	11931
Tokens	136821	18192	34033
NumberSamples	4404	678	1251

Table 1: Distribution of argument components

3.2.1 AbstRCT

The first dataset used to perform the argument mining experiments is the Randomized Clinical Trials (RCT) of medical abstracts in Mayer et al. (2021).

The dataset is divided according to 5 types of diseases: neoplasm, glaucoma, hepatitis B, and hypertension, although in our experimentation only the one corresponding to neoplasm has been used because we have splits of dev and test. The distribution of abstracts is 500 for neoplasm, 100 for glaucoma, and 100 in the mix set, which consists of a mix of 20 examples for each disease.

The dataset is annotated separately for argument and relation components, where the distribution can be seen in tables 1 and 2 respectively.

Argument components are classified with Claim and Premise labels using IOB2 encoding (Tjong Kim Sang and De Meulder, 2003). The number of premises is more than twice the number of claims, and on average, the premises are longer than the claims, as it can be seen in Table 3. Claims can be detected by the information that composes them as well as by specific phrases such as “According to the results” or “The results support”. Premises, on the other hand, describe different evidences of the study.

In Argument relations, the notation used is used to perform the classification task where two arguments are given, and you want to predict the type of relation between them in a

	Train	Dev	Test
Support	1194	185	359
Attack	200	30	60
noRel	12892	1815	3961
NumberSamples	14286	2030	4380

Table 2: Distribution of argument relations

binary classification task. The labels used for this purpose are *support*, *attack*, and *noRel* when there is no relation between the two previous ones. Each of the possible arguments has been paired with another, causing a noticeable unbalanced label distribution, as shown by Table 2. Furthermore, Example 3.1 shows an argument relation sample where each instance is composed by two argument components.

Example 3.1 Argument relation sample

__**label**__**noRel** No study arm effect was observed for function discussions −− but not for function discussion, suggesting that potentially serious problems may remain unaddressed.

__**label**__**Attack** No study arm effect was observed for function discussions. −− Training oncologists in responding to patient-reported functional concerns may increase the impact of this intervention.

__**label**__**noRel** Clinic discussions were associated with severity of patient-reported symptoms but not with patient-reported functional concerns. −− Patients in the intervention arm discussed more symptoms over time compared with patients in the attention-control (P = .008) and control (P = .04) arms.

__**label**__**noRel** Clinic discussions were associated with severity of patient-reported symptoms but not with patient-reported functional concerns. −− No study arm effect was observed for function discussions.

__**label**__**noRel** Clinic discussions were associated with severity of patient-reported symptoms but not with patient-reported functional concerns. −− A positive longitudinal impact of the intervention on symptom discussion was observed,

__**label**__**noRel** Clinic discussions were associated with severity of patient-reported symptoms but not with patient-reported functional concerns. −− but not for function discussion, suggesting that potentially serious problems may remain unaddressed.

__**label**__**Support** Clinic discussions were associated with severity of patient-reported symptoms but not with patient-reported functional concerns. −− Training oncologists in responding to patient-reported functional concerns may increase the impact of this intervention.

__**label**__**noRel** A positive longitudinal impact of the intervention on symptom discussion was observed, −− Patients in the intervention arm discussed more symptoms over time compared with patients in the attention-control (P = .008) and control (P = .04) arms.

	Train	Dev	Test
O	170524	42975	38554
B-PER	0	0	0
B-LOC	11	0	6
B-ORG	24	0	5
B-MISC	37	4	9
I-PER	11128	3149	2773
I-LOC	8286	2094	1919
I-ORG	10001	2092	2491
I-MISC	4556	1264	909
Tokens	204567	51578	46666
NumberSamples	14987	3467	3684

Table 3: Distribution of CoNLL 2003

3.2.2 CoNLL 2003

CoNLL 2003 (Conference on Natural Language Learning) (Tjong Kim Sang and De Meulder, 2003) is a shared task evaluation campaign for natural language processing systems, specifically for the task of language-independent named entity recognition. The CoNLL 2003 shared task provided a standard data set and evaluation framework for researchers to compare the performance of their named entity recognition systems. The data set consists of news articles from the Reuters Corpus, annotated with named entities and syntactic chunk tags.

For experimentation, we have been used the original CoNLL 2003 instead of the one in the EntLM paper repository. The labels use IO2 encoding following the distribution of data presented in Table 3. In the IO2 format I-PER, I-LOC, I-ORG, and I-MISC are used if a token belongs to one of the categories, B-PER, B-LOC, B-ORG and B-MISC are only used if two contiguous tokens belong to a category, in order to distinguish where one ends and the other begins; O is used when the token does not belong to any of the categories.

In general, most NER sequences are only a few tokens long, usually not exceeding three tokens. The longest-named entities are the ones related to persons, as there is a noticeable difference in the number of I-PER labels compared to the others.

3.3 Evaluation

We use F1 score to evaluate sequence labelling and relation extraction. This is calculated as the harmonic mean of the precision and recall of a model, where precision is the number of true positive predictions divided by the total number of positive predictions, and recall is the number of true positive predictions divided by the total number of actual positive examples. F1 can be calculated from a micro or macro perspective.

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}$$

F1 macro is calculated by treating each class in the dataset separately and averaging their F1 scores. This is useful for datasets where the class distribution within each group is balanced, but the overall dataset is imbalanced.

$$MacroF1 = \frac{\text{sum(F1 scores)}}{\text{number of classes}}$$

F1 micro is calculated by treating all instances in the dataset as a single group, regardless of their class labels.

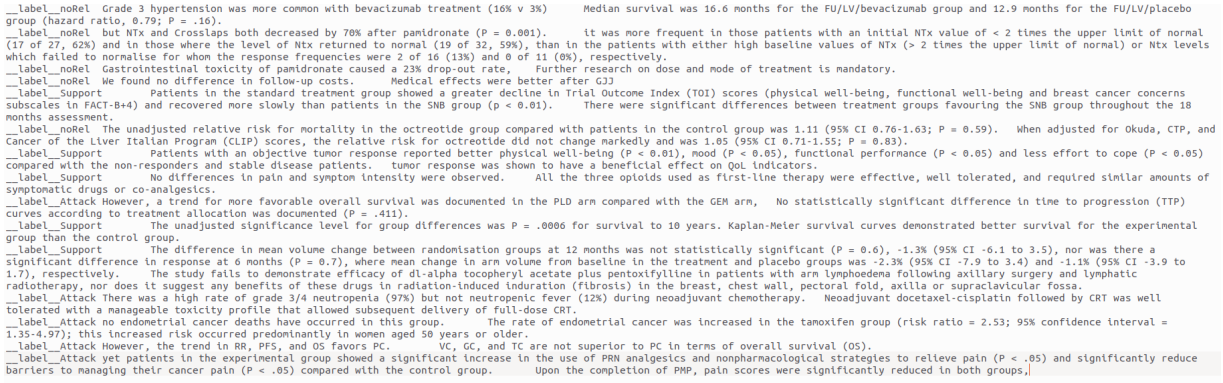

Figure 4: Examples in K5-shot for relations.

4 Experimental setup

In this section, we will first explain how the samplings for few-shot learning are created using the datasets mentioned in the previous section. Secondly, we will detail the experimental details to train the different models. Finally, we will discuss the label words used in the systems where they are needed.

4.1 Sampling

The sampling of the datasets used to perform the few-shot experiments was done in two ways, K-shot, and percentage split [2]. The first was performed using the sampling methodology used in the EntLM paper. This is based on creating 4 or 3 shots of a given number. For example, for K5-shot we use 3 documents where each document contains 5 sentences or tokens as an example for each label.

K-shot has been used both for argument components and argument relations, for which we have obtained 5, 10, 20, and 50-shots respectively. Figure 4 shows the K-5 sampling generated for relation extraction. If we apply the K-shot method to the AbstRCT (argument component detection and relation extraction) and to the CoNLL 2003 datasets then we obtain the samplings depicted in the following tables. First, Table 4 shows the distribution per argument component (claims and premises) from AbstRCT, while the sampling obtained for the relations is shown in Table 5.

Finally, the K-shot sampling generated for NER in the CoNLL 2003 dataset is described by Table 6 corresponds to the CoNLL 2003 sampling. As it can be seen, with K-5 sampling the first document is composed of 8 sentences where there are 5 instances for each NER class.

For the second sampling, we propose an alternative and simple way to generate few-shot data based on a percentage that represents the number of example sentences with respect to the whole dataset. For example, for a 5% sampling we have 5% of example sentences

[2]GitHub repo: `https://github.com/jonmanzanal/Sampling-FewShot-ArgumentMining`.

	O	B-Claim	B-Premise	I-Claim	I-Premise	Tokens	Number_Sentences
K 5							
1	109	6	6	97	118	336	15
2	136	5	5	115	191	452	15
3	117	5	5	102	82	311	13
4	126	5	8	102	144	385	14
K 10							
1	219	13	11	201	373	817	30
2	316	10	13	215	349	903	30
3	234	10	10	187	333	774	28
4	215	10	14	225	344	808	26
K 20							
1	515	24	25	471	841	1876	59
2	683	21	24	496	632	1856	60
3	508	23	20	447	627	1625	56
4	575	23	24	485	657	1764	56
K 50							
1	1284	60	63	1144	2031	4582	147
2	1552	56	58	1230	1609	4505	147
3	1320	57	59	1101	1697	4234	145
4	1325	61	58	1308	1825	4577	142

Table 4: Distribution for the K-shot sampling in AbstRCT in the component detection task

	noRel	Attack	Support	Num_Sentences
K 5				
1	5	5	5	15
2	5	5	5	15
3	5	5	5	15
K 10				
1	10	10	10	30
2	10	10	10	30
3	10	10	10	30
K 20				
1	20	20	20	60
2	20	20	20	60
3	20	20	20	60
K 50				
1	50	50	50	150
2	50	50	50	150
3	50	50	50	150

Table 5: Distribution for the K-shot sampling in AbstRCT in the relation detection task

	O	B-PER	B-LOC	B-ORG	B-MISC	I-PER	I-LOC	I-ORG	I-MISC	Tokens	Num_Sentences
K 5											
1	168	0	5	0	5	2	0	3	4	187	8
2	148	0	5	0	5	2	9	0	10	179	8
3	150	0	5	0	5	0	14	1	8	183	8
K 10											
1	350	0	10	0	10	9	23	1	21	424	18
2	342	0	10	0	10	8	25	0	20	415	18
3	309	0	10	0	10	6	23	8	18	384	17
K 20											
1	489	0	11	0	20	14	37	6	28	605	23
2	474	0	11	0	20	11	30	9	31	586	24
3	576	0	11	0	20	15	39	10	41	712	26
K 50											
1	811	0	11	0	39	23	49	18	66	1017	37
2	811	0	11	0	39	23	49	18	66	1017	37
3	811	0	11	0	39	23	49	18	66	1017	37

Table 6: Distribution for the K-shot sampling in CoNLL 2003

based on the whole dataset. This is illustrated by Table 7, where the complete dataset of AbstRCT is composed of 4404 sentences, 5% of 4404 is 220, the same examples as in the 5% sampling. Following this, we have created 5 splits increasing up to 40% of the dataset, the splits being 5%, 10%, 20%, 30%, and 40% respectively. These splits have been made by multiplying the total number of sentences of the dataset by the respective percentage in order to divide it and save the part of the examples respective to the percentage. It should be noted that our method mantains the original distribution of argument components and relations across classes.

	O	B-Claim	B-Premise	I-Claim	I-Premise	Tokens	Number_Sentences
100%	67198	790	1644	15350	51839	136821	4404
5%	3358	36	81	671	2885	7031	220
10%	6578	77	164	1507	5800	14126	440
20%	13455	161	318	3052	10534	27520	880
30%	20123	236	501	4667	15938	41465	1321
40%	26756	312	664	6153	21162	55047	1761
5% balance	3358	36	26	671	756	4847	174
O less	3010	36	26	671	756	4499	162
10% balance	6577	77	58	1507	2288	10507	352
O less	5887	77	58	1507	2288	9817	330

Table 7: Distribution for the percentage sampling in AbstRCT in the component detection task

As our percentage-based method maintains the original distribution among argument component classes between models, we thought it could also be interesting to generate samplings to obtain more balanced data. This is because if we look at, for example, CoNLL 2003 or AbstRCT, there is one label whose representation is notably inferior to the others. In CoNLL 2003 it is I-MISC, as it can be seen in the Table 9 that for the whole dataset, there are 4556 tokens classified with I-MISC in comparison with the more than 11128 classified as I-PER. With respect to the AbstRCT dataset, Table 7 shows that B-Claim label is underrepresented.

	noRel	Attack	Support	Num_Sentences
100%	12892	1194	200	14286
5%	656	7	51	714
10%	1297	16	115	1428
20%	2582	39	236	2857
30%	3901	56	328	4285
40%	5189	74	451	5714
10% less	780	16	115	911

Table 8: Distribution for the percentage sampling in AbstRCT in the relation detection task

	O	B-PER	B-LOC	B-ORG	B-MISC	I-PER	I-LOC	I-ORG	I-MISC	Tokens	Num_Sentences
100%	170524	0	11	24	37	11128	8286	10001	4556	204567	14987
5%	8290	0	1	0	5	811	584	424	294	10409	749
10%	18474	0	1	0	8	1487	1132	839	586	22527	1498
20%	37845	0	1	0	14	2991	2260	1616	1217	45944	2997
30%	54880	0	1	0	14	3851	2990	2942	1606	66284	4496
40%	69443	0	1	0	21	5065	3722	4056	2063	84371	5994
5% balance	6141	0	1	0	5	585	434	265	294	7725	507
O less	5436	0	1	0	5	554	402	238	265	6901	477
10% balance	13311	0	1	0	8	1164	799	500	586	16369	982
O less	11964	0	0	0	8	1111	745	445	533	14806	903

Table 9: Distribution for the percentage sampling in CoNLL 2003

With this idea of balancing, we try to balance the dataset across labels for the 5% and 10% samplings. If we look at Table 9, it can be seen that we have been able to achieve in CoNLL 2003 to lower all the labels closer to I-MISC. In the case of AbstRCT, the difference is more remarkable, as can be seen in the Table 7, reducing the B-Premise up to 26 in 5% and up to 58 in 10%, closer to the frequencies reported to B-Claim.

We also performed an alternative sampling by reducing the number of sentences composed of tokens that are not classified with any particular label, namely, classified as O. In both CoNLL 2003 and AbstRCT, the most frequent label is Os, as it can be seen in Table 9 and Table 7, so we reduced by 10% and 5% the number of sentences composed of only tokens with the label O for the already balanced of 5% and 10% splits created previously. Thus, for CoNLL2003 Table 9 shows that we have reduced the Os from 8290 to 5436 for 5% and 10% from 18474 to 11964. With respect to AbstRCT, Table 7 reports that at 5% they have been reduced to 3010 and at 10% to 5887.

Finally, we also rebalanced the relations in the AbstRCT dataset. As the task is formulated as a pair-wise classification task between two arguments, this means that we need to consider, for a given document, all possible relations (support, attack, norel) with each other. This means that the resulting dataset is heavily biased towards *noRel*. The result of this, as seen in Table 8, is that for the 10% split there are 1297 examples classified as noRel while for *attack* there are just 16, and for the *support* relation 115. With this in mind, we raised the idea that this could negatively affect the performance of language models for the relation extraction task, so it would be interesting to mitigate this issue by removing 40% of examples with the noRel relations. We test this with the 10% split. The

	MMCV	BERT	EntLM
Sequence length	128	128	128
Train batch size	32	4	4
Eval batch size	8	8	32
Lr	5e-5	1e-4	1e-4
Epochs	3	20	5
Model	BERT	BERT	BERT

Table 10: Hyperparameters for the CoNLL 2003 dataset.

result can be seen in the Table 8, where the examples of noRel are reduced from 1297 to 780.

4.2 Training details

In this section we will explain how the experiments have been set up, detailing which hyperparameters have been used, and which dataset with the system has been used. To perform these experiments we have used two GPUs, a Titan Xp and a Titan V, both of which have a VRAM of 12 GB.

4.2.1 CoNLL 2003

We used three systems to experiment with CoNLL 2003: MMCV, BERT, and EntLM. We have done experiments with each of the splits with both the data in its original form and by lowercasing it. We have made 4 runs with a random seed in each one of them and the final reported result is based on the average of the 4 experiments. The results are reported on the development split.

For the MMCV system, the same hyperparameters have been used for both the CoNLL 2003 and the AbstRCT datasets (Mayer et al., 2021): sequence length of 128, a batch size of 32, and an eval batch size of 8. We have fine-tune the model and evaluated it in 3 epochs in CoNLL 2003 dataset with a learning rate of 5e-5.

For fine-tuning the BERT model we followed Ma et al. (2022): 128 sequence length, a train batch size of 4 and an eval batch size of 8, and a learning rate of 1e-4. We have trained the model and evaluated it in 20 epochs.

The EntLM prompting system is trained following the method of its original paper (Ma et al., 2022). Thus, we have used the default hyperparameters, which are 128 sequence length, a training batch size of 4 and an eval batch size of 32, and a learning rate of 1e-4. The model is trained and evaluated after 5 epochs.

Sequences are represented using the IO encoding because the EntLM authors claimed that results were better in this way (Ma et al., 2022).

	MMCV-Train	MMCV-Test	mBERT	EntLM
Sequence length	128	128	128	128
Train batch size	32	32	16	16
Eval batch size	8	8	32	20
Lr	5e-5	2e-5	5e-5	5e-5
Epochs	3	1	5	20
Model	mBERT	mBERT	mBERT	mBERT

Table 11: Hyperparameters for the AbstRCT dataset in the component detection task

	MMCV	PET	iPET
Sequence length	128	256	256
Train batch size	32	16	16
Eval batch size	8	24	24
Lr	5e-5	1e-5	1e-5
Epochs	3	3	3
Model	SciBERT	SciBERT	SciBERT

Table 12: Hyperparameters for the AbstRCT dataset in the relation detection task

4.2.2 AbstRCT

We use the three same systems for argument component detection: MMCV, mBERT, and EntLM. We have made a run for each sampling with a predefined seed. We have only made one run for each system, and we evaluated it on the test split.

As it can be seen in Table 11, for fine-tuning MMCV we defined a sequence length of 128, a train batch size of 16 and an eval batch size of 20, and a learning rate of 5e-5. We fine-tune the model for 20 epochs.

With respect to mBERT system, used a sequence length of 128, a train batch size of 16 and an eval batch size of 32, and a learning rate of 5e-5. The model was fine-tuned for 5 epochs.

Finally, the EntLM system we used the default hyperparameters: 128 sequence length, a training batch size of 16 and an eval batch size of 20, and a learning rate of 5e-5. The models is trained over 20 epochs.

To extract argument relations we used two systems: MMCV, PET and its iPET variation. We make a run for each sampling with a predefined seed. We have only made one run for each system, and we evaluated it on the test split.

For MMCV we used the default parameters where we defined a sequence length of 128, a batch size of 32, and a learning rate of 5e-5. We have trained the model for 3 epochs (Mayer et al., 2021).

In PET, as we do for the rest of the systems, we also use the default script where a train batch size of 16, an eval batch size of 24, and a gradient accumulation of 4, are defined. We train the model for 3 epochs. Experiments have also been run with iPET, a system

based on iterating several times the PET system to achieve better results; iPET is used using the same hyperparameters as for PET.

4.3 Label words

Both EntLM and PET need a label word that allows them to relate a token with a possible label in order to generalize to the other tokens and correctly predict the corresponding label.

In the case of EntLM, different label words were made using the code released by EntLM. To generate the label words, we define the percentage of token selection, how many tokens we want and the filtering method according to three methods: timesup, data, and LM.

- Data: in this method, we select the most frequent word of the given label in the corpus.
- LM: in this method, we leverage the pre-trained language model for label word searching.
- Timesup: in this method, we select the label words while considering the data distribution and LM output distribution.

After trying the three methods to obtain the label words, with a 0.95 and 0.6 filter ratio and 10 elements and 6 elements respectively, it was found that depending on the split, one label word method performed better than the others. For CoNLL 2003 the best method is the timesup, 0.6 ratio, and getting 6 tokens per label for all splits, using both the percentage and K-shot samplings. For AbstRCT this is a little bit more complicated: for the 100% and 40% splits the most best method is based on the same parameters as for CoNLL 2003, but for 5%, 10%, and 20%, the most useful is the one using LM as a method, 0.95 ratio, and 10 tokens per label. For K-shot, the best choice is LM for all, but for K-5 and K-50 it is a 0.6 ratio and 6 tokens, while for K-10 and K-20 it corresponds to 0.95 and 10 tokens. In addition to using the EntLM method, we created our own label words by detecting the 20 most frequent tokens for each class and obtaining the unique tokens for each of them.

In the case of PET, we use two methods to obtain label words: the one that is used for NLI, provided in the PET system, and our own method which generates label works by using the 20 most frequent tokens per label and obtaining the unique labels for each of them.

5 Results

In this section we will show the results obtained in the experiments. First we will report the sequence labelling results for NER with CoNLL 2003 and argument mining using the AbstRCT dataset. Second, in Section 5.3 we present the results obtain for few-shot learning for the argument relation extraction task.

5.1 CoNLL 2003

Table 13 reports the results with the whole dataset, and with both the percentage and K-sampling for the three systems used for sequence labelling, namely, MMCV, BERT and EntLM.

First, we note that fine-tuning MMCV achieves very poor results when train on a few-shots using the K-shot sampling. However, this changes substantially when trained on the percentage-based samples, were this model obtains quite good results, obtaining 0.8340 of F1 micro using only 5% of the data. We can also observe that with splits as small as 10% and 20%, the results are very good, so doing few-shot with the MMCV system allows us to achieve good results, without the need of having a large dataset. When we balance the dataset there is a notorious loss of performance, achieving with 5% of the data a result of 0.7130 and with 10% a result of 0.8600. This constitutes a loss of 12 and 5 points respectively. Reducing the number of Os does not help either.

In the case of BERT, we can observe that for the K-shots the results are quite good, especially for K-50, where we obtained 0.6280 of F1 score. With the percentage splits, it maintains a fairly stable learning curve with results around 0.86, so the increase of sentences considered as examples does not affect the performance of this system. With the balanced dataset, there is no noticeable difference between the 5% balanced and unbalanced, losing only 3 points.

The EntLM system achieves similar results to those obtained with BERT for K-shot sampling. In fact, they are best across all the systems, achieving with K-50 a score of 0.5781. When using percentage sampling obtains good results but not as good as those achieved by MMCV. As with previous methods, balancing the samplings results also in loss of performance.

When analyzing the results obtained lowercasing the dataset, the overall trend is similar to the one observed with the dataset in its original form except for a couple of differences: (i) performances across systems and type of sampling worsen considerably and, (ii) EntLM degrades most when trained on lowercased data with the K-sampling.

Summarizing, and contrary to published results (Ma et al., 2022), the few-shot experiments show that fine-tuning MMCV on the percentage-based sampling outperforms any other option, including EntLM. This is further illustrated in Figures 5 and 6 (ECAI 2020 refers to the MMCV system), where MMCV obtains the best results overall. Furthermore, while EntLM is best in K-shot settings, those results remain well below the ones obtained using only 5% of the data.

Dataset	Original			Lower			Avg
	MMCV	Bert	EntLM	MMCV	Bert	EntLM	
100%	0.9646	0.8697	0.9099	0.9512	0.8771	0.8649	0.9151
K5	0	0.2005	0.2453	0.0001	0.2056	0.0043	0.1486
K10	0.0016	0.4689	0.4904	0.0001	0.2797	0.1704	0.3198
K20	0.0005	0.5642	0.5781	0.0009	0.3602	0.1857	0.3808
K50	0	0.6280	0.6326	0	0.4127	0.2384	0.4202
5%	0.8340	0.8069	0.8282	0.6863	0.7114	0.7201	0.8142
10%	0.9182	0.8562	0.8745	0.8391	0.7582	0.7791	0.8815
20%	0.9410	0.8736	0.8837	0.8917	0.8093	0.8159	0.8996
30%	0.9509	**0.8824**	0.8886	0.9167	0.8308	0.8265	0.9064
40%	**0.9541**	0.8769	**0.8942**	**0.9204**	**0.8356**	**0.8275**	0.9082
5%‡	0.7130	0.7718	0.7784	0.5960	0.6542	0.6543	0.7521
O less‡	0.6896	0.7594	0.7816	0.5845	0.6410	0.6195	0.7357
10%‡	0.8600	0.8059	0.8137	0.7600	0.7078	0.7200	0.8237
O less‡	0.8517	0.7993	0.8085	0.7585	0.7057	0.7099	0.8172
Avg %	0.9271	0.8610	0.8799	X	X	X	X
Avg K-shot	0.0005	0.4654	0.4866	X	X	X	X
Avg balanced	0.7786	0.7841	0.7956	X	X	X	X

Table 13: CoNLL 2003 results with dev data.‡Results with balanced and balanced data with 10% fewer tokens classified as O. Avg only of the results with the original dataset.

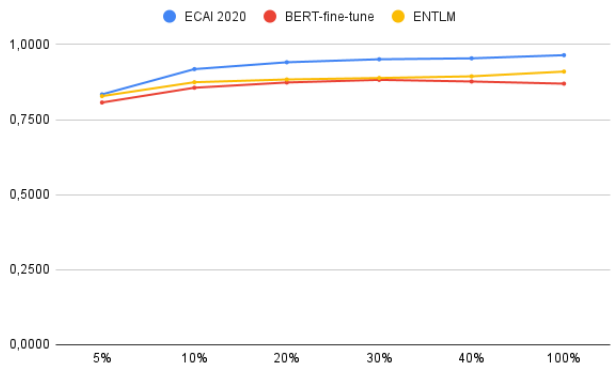

Figure 5: Learning curve for CoNLL 2003 dataset.

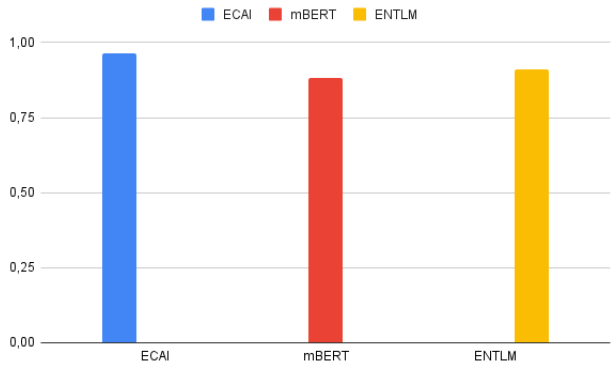

Figure 6: Max values comparison for CoNLL 2003 dataset.

5.2 AbstRCT

The argument component detection evaluation is performed in the in the same way done for the CoNLL 2003 data. Table 14 shows that best results are again obtained by MMCV, although for competitive performance at least 10% of the data is needed. Furthermore, MMCV outperforms EntLM also using K-sampling, although K-sampling results are in general quite poor.

As an example, with K-10 we obtain a very bad result of 0.0002 F1 score, but with K-50 the results are already decent, achieving 0.5804 F1 score. Balancing the dataset also affects negatively. For example, fine-tuning on the balanced 5% split, we obtain a result of 0.4897 for F1 score, a difference of 18 points compared to the original 5% split.

If we check the results obtained using mBERT, with both K-shot and percentage-based sampling the results are quite bad but more stable than those of MMCV. As it is the case for any other evaluation setting, any kind of balancing results in worse performance results.

The EntLM results are quite bad for every setting, including those based on K-shot sampling. While its performance is a bit better using the percentage sampling it still remains far from MMCV.

When testing our method to obtain the label words, we can see that in general it does not improve over the original EntLM's technique. In any case, the trend previously mentioned with respect to other system is maintained.

Finally, the performance when lowercasing the dataset is similar to the one we observed for CoNLL 2003. Thus, results are in general lower and the trends already discussed in previous paragraphs remain the same. It is true that the results obtained using the percentage splits are quite competitive and that in some cases they even improve over those obtained using the original data. Thus, using 20% MMCV obtains a F1 score of 0.8462 while for 10% we have achieved 0.8395 as a result, which is a little higher than that achieved with the original dataset. Another difference is that balancing slightly improves results.

Since the EntLM paper mentions that better results are achieved with IO encoding, we

Dataset	Original				Lower				Avg
	MMCV	mBert	EntLM	EntLMm	MMCV	mBert	EntLM	EntLMm	
100%	0.8916	0.5661	0.5859	0.5381	0.8837	0.5652	0.4366	0.4319	0.6454
K5	0	0.0768	0.1210	0	0	0.0653	0.1417	0	0.0495
K10	0.0002	0.1501	0.1585	0.0039	0.0001	0.1359	0.1782	0.0116	0.0782
K20	0.0841	0.2254	0.2203	0	0.0959	0.2146	0.2042	0.1853	0.1324
K50	0.5804	0.4180	0.3749	0.1502	0.5835	0.4063	0.3210	0.2903	0.3809
5%	0.6701	0.4548	0.3980	0.1341	0.6774	0.4642	0.2538	0.2087	0.4142
10%	0.8378	0.5094	0.5327	0.3593	0.8395	0.5003	0.3056	0.2673	0.5598
20%	0.8575	**0.5436**	0.5328	0.4212	0.8462	**0.5291**	0.3577	0.3442	0.5888
30%	0.8639	0.5102	**0.5761**	**0.5065**	0.8628	0.5013	0.3846	0.3821	0.6142
40%	**0.8746**	0.5247	0.5527	0.4972	**0.8736**	0.5289	**0.4058**	**0.3840**	0.6123
5%‡	0.4897	0.3966	0.1959	0	0.4903	0.3699	0.1112	0.0945	0.2706
O less‡	0.4976	0.3499	0.2272	0	0.4907	0.3317	0.1209	0.0948	0.2687
10%‡	0.7603	0.3754	0.2965	0.2148	0.7640	0.3698	0.1885	0.1359	0.4118
O less‡	0.6884	0.3497	0.3142	0.1389	0.7169	0.4564	0.2456	0.2298	0.3728
Avg %	0.8326	0.5181	0.5297	0.4094	X	X	X	X	X
Avg K-shot	0.1662	0.2176	0.2187	0.0385	X	X	X	X	X
Avg balanced	0.6090	0.3679	0.2585	0.0884	X	X	X	X	X

Table 14: AbstRCT results using IOB encoding with test data.‡Results with balanced and balanced data with 10% fewer tokens classified as O. Avg only of the results with the original dataset.

have done the experiment with the AbstRCT dataset. The results obtained are presented in Table 15, consisting first of the result with the whole dataset, then the sampling used in EntLM, based on K-shots, followed by our sampling based on percentages.

First, modifying the encoding does not affect the results obtained with mBERT. On the other hand, with EntLM there is an improvement in the results. Thus, if we look at the K-shot splits, there is a 7 point improvement with K-5 shot and a more noticeable 10 point improvement for the K-50 split, obtaining a value of 0.4781. If we look at the percentage splits, the improvement is also quite good. For example, for the 5% split, there is a difference of 14 points while when training on the 40% split there is an improvement but not as noticeable, achieving a result of 0.5937. Finally, and contrary to what happened when using the IOB encoding, in this case balancing the splits actually benefits the performance of the models.

EntLM's results are also substantially improved when using our own method to obtain the label words together with the IO encoding. For example, for K-50 we obtain an improvement of 21 points, achieving a result of 0.3698. If we look at the percentage splits, the most noticeable improvement is in the 10% split with a difference of 17 points with respect to those obtained using the IOB encoding.

In any case, the main result is that MMCV is the best system to perform few-shot learning for argument component detection when used on our percentage-based sampling. This is further illustrated by the learning curve in Figures 7 and 8, where we represent the maximum scores obtained for each system.

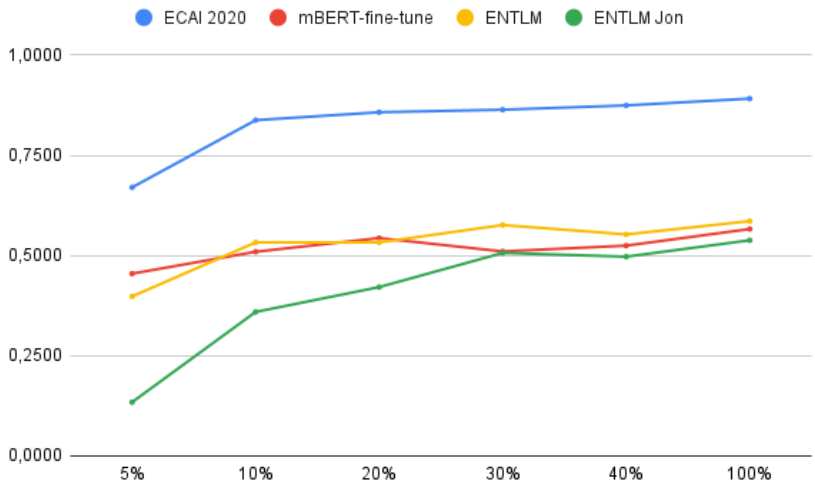

Figure 7: Learning curve for AbstRCT dataset (ECAI 2020 refers to the MMCV system.)

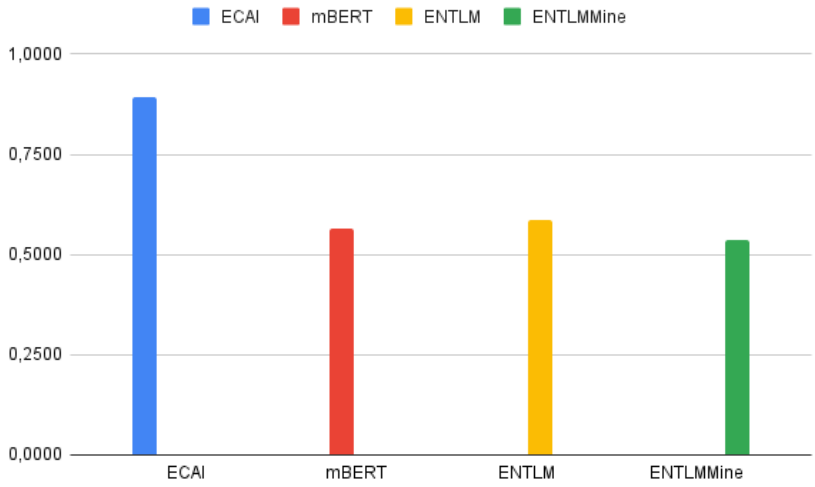

Figure 8: Max values comparison for AbstRCT dataset (ECAI 2020 refers to the MMCV system.)

	mBERT	mBERT†	EntLM	EntLM†	EntLMmine	EntLMmine†
100%	0.5661	0.5661	0.5859	0.6401	0.5381	0.5997
K5	0.0768	0.0612	0.1210	0.1945	0	0.0066
K10	0.1501	0.1209	0.1585	0.2168	0.0039	0.0922
K20	0.2254	0.2158	0.2203	0.3487	0	0.0341
K50	0.4180	0.4088	0.3749	0.4781	0.1502	0.3698
5%	0.4548	0.4548	0.3980	0.5417	0.1341	0.2062
10%	0.5094	0.5094	0.5327	0.5944	0.3593	0.5226
20%	**0.5436**	**0.5436**	0.5328	0.5869	0.4212	0.5043
30%	0.5102	0.5102	**0.5761**	**0.6095**	**0.5065**	0.5278
40%	0.5247	0.5247	0.5527	0.5937	0.4972	**0.5453**
5%‡	0.3966	0.3836	0.1959	0.2522	0	0.0769
O less‡	0.3499	0.3328	0.2272	0.2933	0	0.0394
10%‡	0.3754	0.4007	0.2965	0.4040	0.2148	0.2814
O less‡	0.3497	0.4069	0.3142	0.4296	0.1389	0.2483

Table 15: Comparison of results using IOB encoding and †IO encoding for AbstRCT.‡Results with balanced and balanced data with 10% fewer tokens classified as O.

5.3 Argument relation extraction

We report the results obtained for relation extraction into two different tables. First, Table 16 presents the results with K-shot sampling using both F1-micro and F1-macro for each system used, namely, MMCV, PET, and iPET. Second, Table 17 includes the results obtained using percentage-based sampling.

First, we can see that by doing percentag-based sampling we obtain much better results across systems. Second, while performance in K-shot settings improves as we augment the size of the training data, this is not the case for the splits generated by the percentage method, as 5% is enough to obtain results close to those obtain using the full size dataset. Third, while the difference between the two variants of PET and MMCV are not significant when trained generating the splits by percentage of sentences, this is not the case in the K-shot setting, in which MMCV clearly outperforms the other systems.

For relation extraction we also used our own method to generate the label words. Thus, if we observe the results obtained using PET but with the NLI label words, the results are slightly worse than those obtained with our label words.

If we look at iPET for K-shot sampling, the system achieves worse results than PET and MMCV. This is also the case when training on the splits by percentage, although the difference is not very large. With the balanced split, the behavior is similar to what was achieved in both MMCV and PET, obtaining a result of 0.9010, slightly lower than what was achieved with the corresponding unbalanced split.

After commenting on the results, we can conclude that PET and iPET with their own label words achieve better results than MMCV, as it has been commented or as it can be observed in Figure 9, in which we can see the learning curve, and in Figure 10, where we

Dataset	MMCV		PET		iPET	
	F1-micro	F1-macro	F1-micro	F1-macro	F1-micro	F1-macro
K5						
1	0.4064	0.2553	0.0537	0.0483	0.0163	0.0178
2	0.5956	0.3445	0.2828	0.1739	0.0163	0.0162
3	0.3419	0.2402	0.0749	0.0656	0.0468	0.0369
K10						
1	0.4621	0.3002	0.0921	0.0566	0.0581	0.0589
2	0.6099	0.3570	0.0911	0.0557	0.0931	0.0723
3	0.4453	0.2939	0.0911	0.0557	0.7409	0.3144
K20						
1	0.4567	0.3076	0.3394	0.2011	0.0207	0.0287
2	0.4941	0.3281	0.2552	0.1574	0.0433	0.0687
3	0.3808	0.2775	0.5650	0.2638	0.0315	0.0409
K50						
1	0.7123	0.4610	0.5842	0.3860	0.6946	0.4152
2	0.7522	0.4823	0.4266	0.2906	0.2557	0.2027
3	0.7818	0.4910	0.6399	0.4040	0.4690	0.3207
Avg K5	0.4479	0.2800	0.1371	0.0960	0.0264	0.0237
Avg K10	0.5057	0.3170	0.0915	0.0560	0.2974	0.1485
Avg K20	0.4438	0.3044	0.3865	0.2074	0.0319	0.0461
Avg K50	0.7488	0.4781	0.5502	0.3602	0.4731	0.3128

Table 16: Argument relations with K-shot sampling on test data.

	MMCV	PET	PET†	iPET	iPET†	Avg
100%	0.9113	0.9143	0.9108	0.9177	0.9177	0.9144
5%	**0.9143**	0.9084	0.9054	0.9000	0.9049	0.9066
10%	0.9099	0.9089	0.9089	0.9059	0.9113	0.9090
20%	0.9049	**0.9172**	0.9099	0.9113	0.9138	0.9114
30%	0.9118	0.9133	0.9138	0.9113	0.9108	0.9122
40%	0.9118	0.9167	**0.9163**	**0.9158**	**0.9148**	0.9151
10% less‡	0.9044	0.9030	0.9049	0.9010	0.9069	0.9040
Avg	0.9107	0.9131	0.9109	0.9103	0.9122	X

Table 17: Argument relations results with percentage-based sampling on the test data.†Results obtained using the NLI label word.‡Results with 10% with 40% fewer examples classified as noRel.

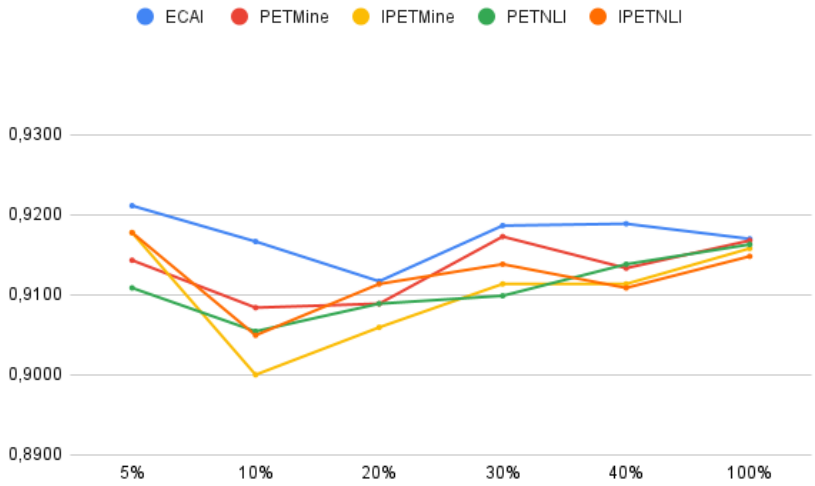

Figure 9: Learning Curve for relations (ECAI refers to the MMCV system).

see the maximum value for each system. On the other hand, NLI's label words achieve quite good results, so using NLI's own label words for argument relations would be a good idea, saving the time of making one of our own. It should be mentioned that this system has a very long execution time as several of them have to be executed iteratively, so it may be that other systems are more recommendable in some situations.

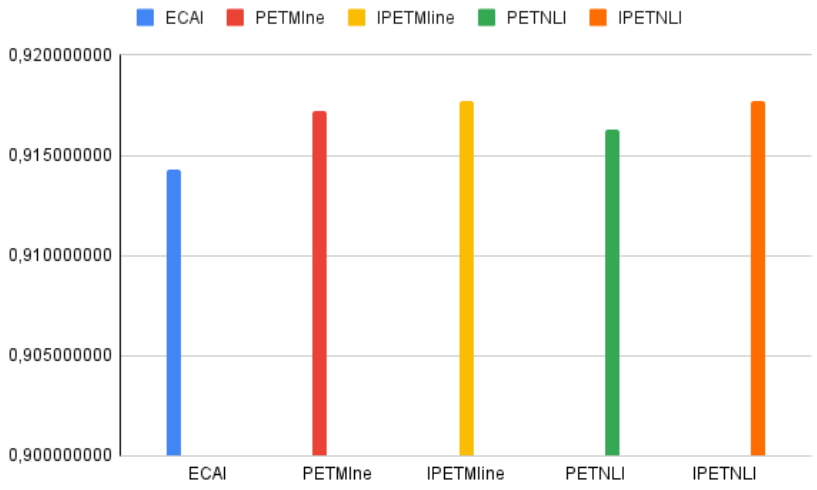

Figure 10: Max values comparison for relations (ECAI refers to the MMCV system).

6 Error Analysis

From the results obtained in previous sections, it has been clear that the model MMCV consisting of SciBERT with GRU layer is the best-performing model for argument mining. In this section, we will provide a qualitative analysis of the predictions produced by the models, with the aim of identifying the most important errors.

6.1 Argument Component Detection

Before diving into a detailed analysis of outputs generated under each individual setting, there are some errors in the predictions throughout all the experiments that can be more or less generalized. Overall, the identification of Premise was more accurate compared to Claim. In fact, the majority of the misclassifications happened in determining Claim arguments. It should be noted that by using IO encoding with EntLM using mBERT, these errors are reduced since all labels start with I, thus eliminating the errors of predicting whether they are B-Claim or B-Premise.

After commenting and analyzing the errors in a generalized way, we will proceed to an analysis of the errors for each of the systems. For this purpose, we have collected the predictions for each system of the percentage split corresponding to 10%. The results have been represented in Table 18.

For the MMCV system we can observe that among the most frequent errors are those corresponding to classify as Premise or Claim tokens that are classified as O. This could be due to the fact that in many sentences O is interspersed between Premise or Claims, and the system classifies the whole sentence with a label without differentiating these Os mentioned. The second most frequent error is not knowing how to differentiate between Claim and Premise (2487 cases).

With respect to mBERT, we can verify that a total of 5551 errors have occurred, the most frequent error being the one that corresponds to the detection of Premise or Claims labels that are actually O, with a number of 2188. The reason must be the same as the one previously described, eliminating the O's that are interspersed throughout the sentence and not differentiating these. The second most frequent error, with a number of 2091, is the one corresponding to the erroneous classification between Claim and Premise. This being a system that is not capable of clearly differentiating between these two labels.

Looking at the EntLM system we have detected a total of 5020 errors. The most frequent error is the same as previously described in both the MMCV and mBERT systems, i.e., the prediction of Claim or Premise instead of O, with a number of errors of 2505. The second most frequent error is not being able to differentiate between Claim and Premise, with an error rate of 1360.

EntLM using IO encoding performed better so the number of errors is lower. The most frequent error is the same as for the other systems, namely, detecting as Premise or Claim tokens that should be classified as O (2097). The second most frequent error, with a number of 1546, is to missclasify O labels as Premise or Claim. Lastly, the least frequent is mislabel Claim as Premise. This is due to the fact that having only IO encoding means

System	Total	O-Premise/Claim	Premise/Claim-O	Premise-Claim
MMCV 10%	7717	3294	1900	2487
mBERT 10%	5551	2188	1272	2091
EntLM 10%	5020	2505	1155	1360
EntLMIO 10%	4941	2097	1546	1298
EntLMMine 10%	5790	2214	2031	1545
EntLMMineIO 10%	5299	2319	1457	1523

Table 18: Prediction errors made by the systems used for AbstRCT.

System	Total	noRel-Sup/Att	Sup/Att-noRel	Sup-Att
MMCV 10%	180	50	128	2
PET 10%	185	59	123	3
iPET 10%	191	64	124	3
PETNLI 10%	193	68	122	3
iPETNLI 10%	190	47	141	2

Table 19: Prediction errors made by the systems used for AbstRCT in the relation detection task.

that there are fewer errors since it only has to predict whether it is Claim or Premise. EntLM using our method to obtain the label words, for both IO or IOB encodings, incurs into more errors than when using the EntLM technique.

6.2 Argument Relation Extraction

As we have done previously we have identified that most of the classification errors are related to misclassifying *support* or *attack* relations as *noRel*, due to the manner in which the dataset has been built.

If we look at the specific errors of each system by looking at the predictions on the test corresponding to the 10% split, we can observe the following issues, summarized in Table 19. Thus, for the MMCV system, we can verify that only 180 errors have occurred, a very small number, the most frequent error being the prediction of noRel instead of Support or Attack. The second most frequent error is the opposite of the previous one, namely, detecting Support or Attack instead of noRel (50 times). The pattern described for MMCV above is repeated for PET and iPET systems, be that using our method for generating the label words or the PET NLI-based technique.

6.3 Other Experiments

In this section we carry out a small experiment to determine how much actual data is required to annotate in order to obtain competitive results using few-shot learning for argument mining in Spanish. Since for this language we do not have any manually annotated

	Neoplasm	Glaucoma	Mixed
10%	0.5686	0.5772	0.5679
20%	0.5841	0.5810	0.5801
30%	0.6974	0.6896	0.6988
40%	0.7622	0.7499	0.7610
50-shot	0.1737	0.1806	0.1768

Table 20: Results of F1 macro obtained for argument component detection on the AbstRCT dataset in Spanish.

data, we use the automatically translated and projected version of AbstRCT published by Yeginbergenova and Agerri (2023).

In order to do so, we first sample the 10%-40% range and 50-shot of the training data and then evaluating on the test data of neoplasm, glaucoma, and mixed datasets. This experiments has been undertaken using MMCV, since it has been the best system for the English version of this dataset. We used the same script and the same hyperparameters that have been used to perform the experiments previously described. The results are presented in Table 20 below.

We can see that few-shot using the 40% of the data is enough to achieve very good results, saving the need to label the full dataset to achieve competitive performance. Taking this result into account, we perform an exercise consisting of manually annotating during 1 hour (after some previous training to get familiarize with the task) a random sample of 1000 sentences from the neoplasm train set. After one hour, we had annotated 58 sentences[3]. Thus, by extrapolating this result we could conclude that in around 10 hours we could obtain around 600 sentences (between 10% and 20% using our percentage sampling method) which is the amount of data required to obtain competitive performance for the argument component detection task. This is considerably less than the effort required to annotate the whole dataset, which would amount to more than 80 hours.

[3]We did check with their gold standard version to check our annotations were in sync.

7 Conclusion

This is thesis is the first comprehensive study, as far as we know, of few-shot learning for argument mining. Our work shows that when performing few-shot learning for sequence labelling and relation extraction tasks the data sampling method has a big influence in the results obtained. For example, it can be observed that fine-tuning MMCV if the best method if trained when percentage-based sampling. This contradicts previous results on few-shot learning for sequence labelling (Ma et al., 2022), suggesting that more work is required to understsand exactly the behaviour of sequence labelling systems in few-shot settings.

Thus, our experimental results show that for argument component detection annotating training data for around 15 hours is enough to obtain state-of-the-art results. Furthermore, we have also shown that 40% of the data allows us to obtain 0.8746 in F1 score, only 2 points less than when fine-tuning MMCV with the 100% of the data.

For relation extraction the results are even more encouraging, as the few-shot methods we have tested were approximating the state-of-the-art using as little as 5% of the training data.

Thus, we must conclude that we need to think more carefully the way we sample our data to perform few-shot learning, as it significantly impacts the results. Thus, future work should explore the effectiveness of different sampling approaches in different few-shot learning scenarios, as well as the development of new sampling methods that can better capture the underlying structure of the data. Another task to be performed would be the utilization and generation of different label words that are more optimal when carrying out this type of task, such as the detection of components in arguments. It would also be interesting to carry out few-shot experiments in cross-lingual settings.

References

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*, pages 118–128, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5115. URL https://aclanthology.org/W17-5115.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, page 129–136, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220192. URL https://doi.org/10.3115/1220175.1220192.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *ACL*, 2020.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.161. URL https://aclanthology.org/2021.findings-acl.161.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Steffen Eger, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 831–844, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1071.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, dec 2008. ISSN 0001-0782. doi: 10.1145/1409360.1409378. URL https://doi.org/10.1145/1409360.1409378.

Tianyu Gao, Xu Han, Ruobing Xie, Zhiyuan Liu, Fen Lin, Leyu Lin, and Maosong Sun. Neural snowball for few-shot relation learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7772–7779, Apr. 2020. doi: 10.1609/aaai.v34i05.6281. URL https://ojs.aaai.org/index.php/AAAI/article/view/6281.

Iker García-Ferrero, Rodrigo Agerri, and German Rigau. Model and data transfer for cross-lingual sequence labelling in zero-resource settings, 2022. URL https://arxiv.org/abs/2210.12623.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.576. URL https://aclanthology.org/2022.acl-long.576.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. URL http://aclweb.org/anthology/N18-1175.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1247. URL https://aclanthology.org/D18-1247.

Xuming Hu, Chenwei Zhang, Fukun Ma, Chenyao Liu, Lijie Wen, and Philip S. Yu. Semi-supervised relation extraction via incremental meta self-training, 2020. URL https://arxiv.org/abs/2010.16410.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.813. URL https://aclanthology.org/2021.emnlp-main.813.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. Template-free prompt tuning for few-shot NER. In *Proceedings of the 2022 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5721–5732, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.420. URL https://aclanthology.org/2022.naacl-main.420.

Tobias Mayer, Elena Cabrio, Marco Lippi, Paolo Torroni, and Serena Villata. Argument Mining on Clinical Trials. In *COMMA 2018 - 7th International Conference on Computational Models of Argument Proceedings*, volume 305 of *Frontiers in Artificial Intelligence and Applications*, pages 137 – 148, Warsaw, Poland, September 2018. URL https://hal.archives-ouvertes.fr/hal-01876462.

Tobias Mayer, Santiago Marro, Elena Cabrio, and Serena Villata. Enhancing evidence-based medicine with natural language argumentative analysis of clinical trials. *Artificial Intelligence in Medicine*, page 102098, 2021.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1113.

Makoto Miwa and Mohit Bansal. 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, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1105. URL https://aclanthology.org/P16-1105.

Raquel Mochales and Aagje Ieven. Creating an argumentation corpus: Do theories apply to real arguments? a case study on the legal argumentation of the echr. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, page 21–30, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585970. doi: 10.1145/1568234.1568238. URL https://doi.org/10.1145/1568234.1568238.

Huy Nguyen and Diane Litman. Context-aware argumentative relation mining. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1137, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1107. URL https://aclanthology.org/P16-1107.

Abiola Obamuyide and Andreas Vlachos. Meta-learning improves lifelong relation extraction. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 224–229, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4326. URL https://aclanthology.org/W19-4326.

Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, page 98–107, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585970. doi: 10.1145/1568234.1568246. URL https://doi.org/10.1145/1568234.1568246.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*. ISCA, sep 2019. doi: 10.21437/interspeech.2019-2680. URL https://doi.org/10.21437\%2Finterspeech.2019-2680.

Raghul Parthipan and Damon J. Wischik. Don't waste data: Transfer learning to leverage all data for machine-learnt climate model emulation. 2022. doi: 10.48550/ARXIV.2210.04001. URL https://arxiv.org/abs/2210.04001.

Y. Pathak, P.K. Shukla, A. Tiwari, S. Stalin, S. Singh, and P.K. Shukla. Deep transfer learning based classification model for covid-19 disease. *IRBM*, 43(2):87–92, 2022. ISSN 1959-0318. doi: https://doi.org/10.1016/j.irbm.2020.05.003. URL https://www.sciencedirect.com/science/article/pii/S1959031820300993.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning, 2017. URL https://arxiv.org/abs/1712.04621.

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. *The Penn Discourse Treebank: An Annotated Corpus of Discourse Relations*, pages 1197–1217. Springer Netherlands, Dordrecht, 2017. ISBN 978-94-024-0881-2. doi: 10.1007/978-94-024-0881-2_45. URL https://doi.org/10.1007/978-94-024-0881-2_45.

Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. Language resources for studying argument. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/648_paper.pdf.

Ramon Ruiz-Dolz, Montserrat Nofre, Mariona Taulé, Stella Heras, and Ana García-Fornes. Vivesdebate: A new annotated multilingual corpus of argumentation in a debate tournament. *Applied Sciences*, 2021.

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few-shot text classification and natural language inference. *Computing Research Repository*, arXiv:2001.07676, 2020a. URL http://arxiv.org/abs/2001.07676.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *Computing Research Repository*, arXiv:2009.07118, 2020b. URL http://arxiv.org/abs/2009.07118.

Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays, 2016. URL https://arxiv.org/abs/1604.07370.

Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1142.

Derek Tam, Rakesh R. Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4980–4991, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.407. URL https://aclanthology.org/2021.emnlp-main.407.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://www.aclweb.org/anthology/W03-0419.

Orith Toledo-Ronen, Matan Orbach, Yonatan Bilu, Artem Spector, and Noam Slonim. Multilingual argument mining: Datasets and analysis, 2020. URL https://arxiv.org/abs/2010.06432.

Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 2 edition, 2003. doi: 10.1017/CBO9780511840005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. Entailment as few-shot learner, 2021. URL https://arxiv.org/abs/2104.14690.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Reza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. Curriculum-meta learning for order-robust continual relation extraction, 2021. URL https://arxiv.org/abs/2101.01926.

Yi Yang and Arzoo Katiyar. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.516. URL https://aclanthology.org/2020.emnlp-main.516.

Anar Yeginbergenova and Rodrigo Agerri. Cross-lingual argument mining in the medical domain, 2023. URL https://arxiv.org/abs/2301.10527.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 71–78, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118703. URL https://doi.org/10.3115/1118693.1118703.