# Cross-Lingual Approaches to Identifying Argument Components and Relations in Norwegian Reviews

Yauhen Khutarniuk


Thesis submitted for the degree of
Master in Informatics: Language Technology
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2022

# Cross-Lingual Approaches to Identifying Argument Components and Relations in Norwegian Reviews

Yauhen Khutarniuk

Cross-Lingual Approaches to Identifying Argument Components and Relations in Norwegian Reviews

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

# Abstract

Argument mining is the process of automatic extraction of certain argumentation structures from data. Argument mining consists of several stages such as argument component detection, argument component classification, and argumentative discourse analysis. The lack of training data in low resource languages is a common issue in argument mining applications. In this work we analyse the possibilities for the application of zero-shot and few-shot language transfer models trained on the language material in a resource-rich language (English) for the tasks of argument component detection, and argument component classification in a low-resource language (Norwegian) with the aim to find out if these techniques can help overcome the challenge of no available training data. In addition, we compare models based on different transformer architectures and experiment with additional hand-crafted features.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

There is no single universally applied definition of argument mining as of time of this writing. In the following we will base our discussion on two recent definitions. One - by Lawrence and Reed (2019), which is rather straightforward and ties the problem to the world of applied technology:

"Argument mining is the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language".

Another definition is more abstract and it shows that the problem of argument mining is actually an interdisciplinary problem that involves cross-domain research, argumentation theory and logic among others. Habernal and Gurevych (2017) define argument mining as the process of "applying a certain argumentation theory to model and analyse the data at hand".

Argumentation theory forms a theoretical basis for argument mining. Argumentation is an interactive and social process. It involves different parties that try to influence the opinion of an addressee so that the addressee perceives a presented standpoint as acceptable (Rigotti and Greco, 2018). A particular instance of argumentation (a text or an utterance) can be described with an argumentation model. In a general sense an argumentation model is a set of argument components along with connections that tie them together (Wambsganß et al., 2020). Stab and Gurevych (2017) in their work are citing (Bentahar et al., 2010) who distinguish three types of argumentation models, which are monological, dialogical, and rhetorical models. Monological models are tightly connected with logic, dialogical models are more focused on the cooperation between interlocutors, while rhetorical models underline how arguments are used as the means of persuasion. This again underlines that argument mining is an interdisciplinary field of study.

Based on this two definitions we can conclude that argument mining is applied to the domain of natural language, its aim is to transform unstructured textual material into structured data that complies with a chosen argumentation model and this process should be automated, i.e. performed with no human interaction.

An introduction to this topic will not be complete if we do not provide

the motivations behind the research in the field of argument mining. Alongside with purely academic interest, argument mining can help to achieve some concrete practical goals: improved information seeking, aided decision making, for example, in litigation (Moens et al., 2007), text summarization, and even more personalized recommendations for consumers (Donkers and Ziegler, 2020).

As we have mentioned above, argument mining is an automated procedure and it is solved using machine learning methods. Some previous researches made attempts to create rule-based argument mining systems (Persing and Ng, 2020) that supposedly do not need training data (at least apart from the linguistic material that had been used to create heuristics for the system). Rule-based systems however have challenges related to scalability, since they require maintenance, rules must be adjusted to different discourse types, genres, and languages. For this reason the majority of approaches focus on solutions that rely on supervised learning techniques. Supervised machine learning also comes with challenges, in particular, they require manually annotated training data.

There exist a number of annotated datasets in English suitable for argument mining, for an extended discussion see Lawrence and Reed (2020). There is only one annotated corpus for Norwegian produced by Evensen (2020). The dataset is, however, very small and consists of small it includes 40 texts sampled from the 'screen' category of the NoReC (Velldal et al., 2018) dataset. The absence of extensive training material for argument mining in Norwegian motivated us to evaluate in this thesis the possibilities of using zero-shot language transfer techniques in argument mining and the potential for improving on the results of argument mining systems using few-shot language transfer.

The first contribution of this thesis is an experimental comparison of zero-shot language transfer models, few-shot language transfer models, and the models trained on the sparse training data in Norwegian. The second contribution is the comparison of multilingual BERT (mBERT) (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) based models for this task. Thirdly, we investigate the influence of additional hand-crafted feature on the training process and the performance of the transformer based models for argument mining. We additionally evaluate how the performance of few-shot language transfer models changes as the proportion of training material in a low-resource language increases in the training dataset. Finally, we develop a domain model for manipulation and creation of annotated text data for running the experiments in argument mining with various experimental configurations.

## 1.1 Outline

This thesis is structured in the following way.

**Chapter 2** includes an overview of previous work done on the subject of argument mining. It describes the process of argument mining in more detail, covers some issues related to the practical aspects of creating

argument mining systems, such as segmentation, feature selection, and approaching argument mining in situations where few training resources are available.

**Chapter 3** includes the information about the datasets used in this thesis, including quantitative and qualitative analysis of the datasets in English and Norwegian.

**Chapter 4** gives the description of the experimental set up, including the process of preparing experimental data, the description of the models used in the experiments, and the procedures for running experiments and evaluating their results.

**Chapter 5** provides a detailed overview and analysis of the results that were achieved after running the experiments.

**Chapter 6** provides a summary of the results that we obtained in the thesis as well as draws on the possibilities for future work and improvements.

# Chapter 2

# Background

## 2.1 Argument Mining Definition

Texts published on debatable issues (whether in political, scientific or general news discourse) have been long subject to sentiment analysis and opinion mining. Although, these techniques provide us with valuable data, they lack explanatory power: while a classifier is able to predict that a discourse unit expresses an opinion, we do not get any information about why the author holds this opinion. However, for a multiplicity of practical tasks it is important not only to extract an opinion but try to find out why an author holds it.

The discourse of online consumer reviews is one of the fields, where having the answers to this why-question can help, for example, build more sophisticated recommendation systems. Conventional recommendation systems typically rely on quantitative approaches, which do lack explanatory power. The argument analysis of consumer reviews can help to extract the aspects which according to a reviewer contributed to positive or negative experiences, and thus help to tailor more personalized recommendations (Donkers and Ziegler, 2020).

In order to complete this task one needs to transform a text into structured argument data. It is necessary to identify the claims being made, the premises that are provided in support or against the claims, as well as the relationships between them. Such process is called **argument analysis**. Argument analysis can be performed manually. But manual argument analysis suffers from the following problems: it requires trained annotators, and it is time-consuming.

Research shows that even trained annotators often fail to achieve reasonable levels of agreement on the task of detecting argumentation schemes (Lindahl et al., 2019; Musi et al., 2016). Lawrence and Reed (2019) point out that it took over 7,000 hours to prepare some datasets. Thus, with the large amount of information being published it is virtually impossible to manually perform argument analysis in real time.

Argument mining addresses this issue. **Argument mining** is the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented in natural language (Lawrence and

Reed, 2019). Habernal and Gurevych (2017) describe argument mining as applying a certain argumentation theory to model and analyse the data at hand. The research of argument mining evolves in two main directions:

- argument mining on the discourse level, and
- information-seeking argument mining.

**Argument mining on the discourse level** attempts to analyse argument structure within an argumentative text. It implies that the texts being analysed belong to a specific genre, for example, argumentative essay or review, and these texts have more or less predefined structures. The downside of this approach is that it is not universally applicable, specifically it fails on texts that lack explicit argument structure. **Information-seeking argument mining** is conceptually different. Given a predefined controversial topic, the algorithm is supposed to detect premises for or against this topic in heterogenous relevant texts. Thus this approach can be applied to texts of different genres without an explicit argumentative structure. However, this approach comes with its own limitations. The input texts are supposed to be already labelled with specific topics. Another downside is that transfer learning to the unseen topics has substantially lower performance as shown by Trautmann et al. (2019).

Trautmann (2020) introduces **aspect-based argument mining** as an extension of information-seeking argument mining. Aspect-based argument mining aims to extract smaller meaningful components that belong to the argument domain. These smaller components are aspects.

In this work we are going to discuss the process of argument mining, including argument identification, argument component classification, argumentative discourse analysis, as well as issues related to argument mining for low resource languages for the current task.

## 2.2 Stages of Argument Mining Process

Argument mining process on the discourse level includes three core tasks, these are (Wambsganß et al., 2020):

- argument identification,
- argument component classification, and
- argumentative discourse analysis.

It is worth mentioning that different authors use slightly different terminology. For example, Stab and Gurevych (2017) are using terms component identification, component classification, structure identification, which correspond to the core tasks we mentioned above.

Information-seeking argument mining has less steps and it includes argument identification and argument component classification (Trautmann, 2020).

For aspect-based argument mining, Trautmann (2020) proposes two additional subtasks:

- aspect term extraction, and
- nested segmentation.

The division of the argument mining process into the subtasks implies that we are bound to take a pipeline approach to argument mining. Although, this is true about earlier research papers such as Stab and Gurevych (2014), in later research authors present end-to-end architectures. For example, in a paper by Morio and Fujita (2018) a novel parallel constrained pointer architecture is presented. This is an end-to-end architecture for relation extraction based on pointer network architecture originally presented by Potash et al. (2017). Pointer networks are networks for decoding variable length sequences, which use attention as a pointer in order to select an element of input as the output.

In this thesis we are using this division for descriptive purposes only. Also, we find it reasonable to consider the tasks of argument identification and argument classification as a single integrate task, as they are essentially overlapping in practice.

### 2.2.1 Argument Identification and Argument Component Classification

**Argument identification** is the process of identifying non-overlapping spans of text as being part of an argument structure or not. Some authors further subdivide this task into text segmentation and argument/ non-argument classification (Lawrence and Reed, 2019). **Argument classification** is a multi-class classification task. The classes in question represent the components of an argumentation model being applied in each particular case. One of the widely adopted sets of such classes are *major claim, claim, premise* as in the work by Stab and Gurevych (2014). For the definition of these classes see Subsection 2.2.3

Argumentation mining systems described in the previously mentioned works do not implement argument identification as a separate independent step. It is rather performed simultaneously with argument classification, i.e. argument component candidates are either attributed an argument component class label or not.

### 2.2.2 Text Segmentation

One of the questions that arise early in designing an argument mining system is the choice of elementary argumentative discourse units. These are those minimal units that constitute an argumentation structure and then segmenting input text into these units. In early works on argument mining, such as Moens et al. (2007), isolated sentences are used as atomic analysis units and only intra sentence features are considered for argument and non-argument classification. The drawback of such approaches is that the context where a sentence is used is disregarded. This in turn causes among others the following problems: there may be several elements of argumentation within the boundaries of a single sentence (in the corpus

of persuasive essays compiled by Stab and Gurevych (2014) only 30% of argument components span over an entire sentence, a sentence may constitute argument element in one text and when considered in another context the same sentence is not a part of an argument, an isolated sentence may simply lack any discriminative linguistic features required for the correct classification.

Despite the above named disadvantages, text segmentation into sentences is used in recent works. Although, sentences are not considered in isolation. Morio and Fujita (2018) successfully apply sentence level segmentation in argument mining for discussion threads. Habernal and Gurevych (2017) use a hybrid approach, where golden data is annotated on the token level. If a given sentence includes only one argument component, then the whole sentence gets the label of the component. If this sentence contains multiple argument components, then the sentence gets the label of the component with the largest span.

Lawrence et al. (2014) proposes to segment text into propositions. The proposed algorithm first splits a text into words, and then using a set of hand-crafted features marks proposition spans with delimiting tags. This method addresses the problem of argument elements spanning across the boundaries of multiple sentences or multiple argument elements contained within one sentence. However, this method comes with a number of disadvantages: different artifacts such as punctuation, introductory words, etc. are captured in the propositions lying on the sentence boundaries. Furthermore, in the implementation by Lawrence et al. (2014) the algorithm showed rather low precision on determining the exact boundaries of the propositions. Thus, these errors would propagate to the downstream tasks.

The corpus of persuasive essays compiled by Stab and Gurevych (2014) is marked with argument components on the clause level. I means that argument components do not necessarily span across a whole sentence and do not cross the boundaries of a sentence. It offers higher flexibility compared to sentence level segmentation. However, it still can not model complex cases, when, for example, one argument component is contained in another.

Trautmann et al. (2019) suggest to perform argument unit recognition on the token level. Argument components are annotated as spans of tokens. Trautmann et al. (2019) claim that this approach helps to create annotated text using crowd-sourcing (using non-expert annotators) and achieve high level of agreement ($\alpha_{u_{nom}} = 0.71$). However, it is worth to mention that Trautmann et al. (2019) employ simplified annotation scheme compared to, for example, Stab and Gurevych (2014). The latter report comparable level of agreement between the annotators $\alpha_{u_{nom}} = 0.72$. However, the task at hand is more complicated then the one presented by Trautmann et al. (2019). Token level segmentation is more suitable for information seeking argument mining. Since the latter requires extracting meaningful subcomponents (aspects) from argument components.

To sum it up, there are three main ways to segment texts for argument mining:

- Sentence level,
- Clause level, and
- Token level.

Sentence and clause level segmentation is more suitable for argument mining on the discourse level, while token level segmentation is required for aspect based argument mining.

### 2.2.3 Argument Component Types

An argument is not monolithic. It consists of several different components and the components are connected with certain relations. Argument components and their relations form a structure, which is commonly called an **argumentation scheme**.

Researches developed various argumentation schemes. For example, the model of argumentation by Toulmin and Dawsonera (2003) and its modifications are widely used in argument mining. The original model includes the following components (Bentahar et al., 2010):

- **Claim** - assertion or a conclusion presented to the audience and which has potentially a controversial nature.
- **Data** - statements specifying facts or previously established beliefs related to a situation about which the claim is made.
- **Warrant** - statement, which justifies the inference of the claim from the data.
- **Backing** - set of information, which assures the trustworthiness of a warrant.
- **Qualifier** - a statement that expresses the degree of certainty associated to the claim.
- **Rebuttal** - a statement presenting a situation in which the claim might be defeated.

Another conceptually similar model was proposed by Rigotti and Greco (2018) is Argumentum Model of Topics. It includes the following basis components:

- **Endoxon** - general premise that is accepted by the relevant public.
- **Datum** - a premise of factual nature.
- **Maxim** - a premise of argumentation, maxims are considered propositions that are known per se.
- **Minor premise** - first/ intermediary conclusion.
- **Final conclusion** - main conclusions at the core of the argument.

The combination thereof can form different argumentation schemes such as Intrinsic-Mereological (premise gives an example that justifies the claim), Intrinsic-Causal (premise and claim are connected by a cause-effect relation), etc. Musi et al. (2016) provide a comprehensive guide for human annotators with criteria for the identification of such schemes and their components.

Original Toulmin's argumentation scheme and Argumentum Model of Topics have a number of weak points that make it difficult to apply in argument mining (Stab and Gurevych, 2014). The components of the models lack formal unambiguous definition. It might be challenging to distinguish data, warrant, and backing components in Toulmin's argumentation scheme without extra linguistic knowledge. Similarly, endoxon, datum, and maxim are rather difficult to differentiate (for example, how one should treat a premise of factual nature that is know per se?). As the result, even trained annotators apply these schemes with a low level of agreement (Musi et al., 2016). As the result, it is difficult to produce training data for machine learning applications.

Stab and Gurevych (2014) simplified original Toulmin's model and proposed the following argument components:

- **Major claim** - the central position of an author with respect to the topic.
- **Claim** - a controversial statement that becomes valid or true in the presence of additional support, which attacks or supports a major claim.
- **Premise** - a reason given by an author for persuading readers of the claim.

Major claim is introduced in order to account for arguments with a more complicated structure as the main claim of a text. In the minimal case an argument consists of a claim (which will be the major claim) and some premises. This set of components is rather general and is not able to capture finer nuances of the argument structure, e.g. if a premise is factual or inferred from prior premises, but it proved to be suitable for argument mining. As reported by many researchers (Stab and Gurevych (2014); Habernal and Gurevych (2017); Morio and Fujita (2018)), the annotators participating in studies demonstrate high level of agreement when applying the aforementioned scheme.

Habernal and Gurevych (2017) tested a new argumentation model based on Toulmin's model (Toulmin and Dawsonera, 2003). They proposed to use the following argument components: claim, premise, backing, rebuttal, and refutation. Predictably, the highest level of annotator agreement was achieved for claim and premise components, while the agreement was unsatisfactory for backing, rebuttal, and refutation. During the study annotators had to mark text of different sizes (articles, blog posts, comments, forum posts). It is worth to mention that the agreement scores for backing and rebuttal turned out to be substantially lower (almost 0)

for larger texts, i.e. articles and blog posts. This study proves that the argumentation scheme with just two core components (claim and premise) are a more viable choice for machine learning applications.

### 2.2.4 Argument Component Classification

In this section of the article we review what methods are applied in order to mark text segments with argument component type labels. We will also review the features that are employed to carry out this task.

In general, argument component classification is a sequence classification task. The sequence in question may be one of the types described in Section 2.2.2. The classifier has to label a candidate argument component with a component type, e.g. claim, premise or none. A candidate argument component is represented with a feature vector. The following feature types can be used for the task:

- hand-crafted features;
- word embeddings;
- contextualized embeddings, and
- combination of the above mentioned.

**Hand-crafted features** - is one of the early approaches for argument candidate representation that was applied for argument mining. For instance, this approach is employed by Moens et al. (2007). As the name implies, the features that would represent an argument component candidate are manually created by a human designer. These features are supposed to represent an argument component candidate in a way that enables a learner to discriminate non-argumentative material from argument components, and argument components of different types. In their experiment Moens et al. (2007) used the following features:

- **Unigrams** - each token in the text segment.
- **Bigrams** - each pair of successive tokens.
- **Trigrams** - each three successive tokens.
- **Adverbs** - adverbs, they are identified with a part of speech tagger on the feature extraction stage.
- **Verbs** - verbs, they are identified with a part of speech tagger on the feature extraction stage.
- **Modal auxiliary** - binary feature, shows if the auxiliary is present in the text segment.
- **Word couples** - all permutations of two words in the segment.
- **Segment length** - the number of tokens in a segment.

- **Average token length** - the average length of the tokens in a segment.
- **Number of punctuation marks**.
- **Punctuation patterns** - if a punctuation mark appears more than once in a segment, it is considered to be a pattern.
- **Keywords** - used a list of 286 hand picked keywords that may indicate presence of an argumentative structure.

Additionally and among others Stab and Gurevych (2017) experiment with the following hand-crafted features:

- **Binary lemmatised unigrams**.
- **Position of the component** - shows if a component is first or last in a paragraph. Number of preceding and following components in a paragraph.
- **Indicators** - similarly to **keywords** in the feature set by Moens et al. (2007), these are words and phrases that help identify an argument component (e.g. in addition, because).
- **Context** - shared noun phrases with introduction and conclusion.
- **Conditional probability of a component** - a conditional probability that a candidate component is one of the argument component types given the tokens preceding a component, the probability is calculated using the maximum likelihood calculated from the training data.

There were attempts to extract argument components from texts in a unsupervised fashion (Persing and Ng, 2020). The argument components are labelled using predefined heuristics. These heuristics, actually, correspond with some of the hand-crafted features described above. For instance, Persing and Ng (2020) are using the number of the paragraph the argument component candidate appears in; the location of the sentence the argument component candidate appears in within its paragraph (similar to the position of the components), and the context n-grams surrounding the argument component candidate. The context n-grams are predefined and they roughly are similar to indicators and keywords from the features above.

Using hand-crafted features poses a number of problems. Some of the features mentioned above are language dependent: such as keywords or lexical indicators. One must make up a new list of such keywords, if the classifier is applied to a new language. Some features are crafted for a specific type of texts being processed, such as the position of a component or context. Stab and Gurevych (2017) parsed argumentation structures in persuasive essays, which have more or less equal length, they have several paragraphs one of which is an introduction, and one of which is a conclusion. However, argument mining is a more universal problem, and argument mining can be applied to texts of different genres which may

have varied length and structure. As the result, hand-crafted structural features can not be applied universally.

**Word embeddings** are vectors that represent words as dense vectors. These vectors are derived by various training methods from neural-network language modelling (Mikolov et al., 2011). Unlike discrete symbolic representation of words with n-grams, word embeddings can capture semantic properties of words, such as similarity, synonimity, and analogy. Word embeddings have higher generalization power. There exist pre-trained word embeddings that can be used off-the-shelf. These are created using the existing frameworks like Word2vec (Řehůřek and Sojka, 2010), fastText (Bojanowski et al., 2016), and GloVe (Pennington et al., 2014). However, the performance of an algorithm using word embeddings depends on the choice of the training corpus (genre, topic), the size of the contexts that are used during the training, as well as other hyperparameters of the algorithm used for creating word embeddings (Levy et al., 2015). Also, word embeddings do not capture the difference between different sense of a word. The word mouse, whether used in a sense of an animal or a device is represented by the same dense vector.

**Contextualized Embeddings** - these are dense vectors that represent input words and capture their semantic properties. However, there is no one to one correspondence between a word and a vector. The vectors are inferred from the context where the processed word appears. This alleviates the issue with polisemantic words. Contextualized word embeddings were shown to demonstrate high performance in a variety of natural language processing tasks (Devlin et al., 2019).

Furthermore, additional derived features can be used. These are features derived using other models and/or systems during the learning and inference stages. For example, Habernal and Gurevych (2017) are employing LDA topic labels (A. K. McCallum, 2002), scores for sentiment categories (Socher et al., 2013), semantic roles from Clear NLP Semantic Role Labeler (Choi, 2012), co-reference features from from Stanford Coreference Chain Resolver (Lee et al., 2013).

We will further give a brief overview of some model architectures employed for the classification of argument components.

Moens et al. (2007) use **maximum entropy** and **multinomial naive Bayes** models for argument component classification. The classified sequences are represented by hand-crafted features. These approaches are computationally effective. Furthermore, they allow to evaluate the influence of particular features on the results of classification. On the down side, the classes in question should be linearly separable (in case of using maximum entropy model). Also, these simple architectures are not capable to capture the influence of the context on the separate tokens, although Moens et al. (2007) used word couples feature to tackle the problem and according to their research word couples feature compared to other sequence representations (unigrams, bigrams) showed the best results.

Habernal and Gurevych (2017) are using **Structural Support Vector Machines classifier** for sequence labelling model designed by Joachims et

al. (2009). The inputs are represented by real valued vectors. Joachims et al. (2009) report that their structural SVM has time complexity linear in the number of training examples, which is substantially faster than standard implementation in Scikit Learn library [1] ranging from $\mathcal{O}(n_{features} \times n^2_{samples})$ to $\mathcal{O}(n_{features} \times n^3_{samples})$ making the algorithm more suitable for the tasks with large datasets.

Eger et al. (2017) use **BiLSTM-CRF (BLC)** (Huang et al., 2015) **with convolutional neural nets (CNNs) on the character-level** (Ma and Hovy, 2016) leading to a BiLSTM-CRF-CNN (BLCC) model. The character-level CNN may address problems of out-of-vocabulary words, that is, words not seen during training.

Trautmann (2020) utilize the base and large versions of **BERT** (Devlin et al., 2019) with an additional CRF-Layer (Sutton and A. McCallum, 2010) on top of it as the final classification layer in the architecture.

The above mentioned models are applied to different datasets and the tasks might differ slightly, e.g. different tag sets resulting in different amount of classes or different segmentation strategies. Thus, it is not reasonable to directly compare their performance. However, among the reported results the model used by Trautmann (2020), BERT large + CRF, shows the best F1 scores.

### 2.2.5 Identifying Argumentative Structure

The identification of the argumentative structure is the final step in the argument mining on the discourse level. The identification of the argumentative structure can be performed on the **macro-level**, on the **micro-level** or both. When the analysis is performed on the macro-level one considers the relations between the complete arguments, for example, Ghosh et al. (2014) analyze the argumentative structure of discussion threads, where each contribution to the thread is already considered as an argumentative unit. The relations between argument components are central for the micro-level approaches. Stab and Gurevych (2017) take the micro-level approach. Some authors attempt to perform the analysis both on the micro- and macro-levels (Morio and Fujita, 2018).

We are going to focus on the identification of the argumentative structure on the micro-level. The structure of an argument can be described with a directed acyclic graph, where the nodes are represented by argument components and the edges represent the relations between argument components. The rules for building such a graph depend on the model of argumentation applied at each particular case. For example, Stab and Gurevych (2014) use the simplified Toulmin's model of argumentation (Toulmin and Dawsonera, 2003). Stab and Gurevych (2014) apply the following principles in order to create the argumentative structure: a) there are two types of relations, which are support and attack b) the relations of both types can exist between: 1) a premise and another premise, 2) a premise and a claim, and 3) a claim and a major claim.

[1] https://scikit-learn.org/stable/modules/svm.html#complexity

The argumentative structure can be parsed using different methods. We are going to cover some of them in this thesis. These are parsed using a **context-free grammar**, using a **classifier** in order **to label the relations** between the pairs of the argument structure components, using a **classifier for sequence labelling**, and the **dependency parsing**.

Mochales and Moens (2009) manually created a context-free grammar that was able to parse the argumentative structure of legal texts. These are the examples of some terminal and non-terminal symbols that are part of the grammar: "Contrast rhetorical marker (e.g. however, although, ...).", "Support rhetorical marker (e.g. moreover, furthermore, also, ...).", "Sentence with a conclusive meaning (e.g. therefore, thus, ...)." (Mochales and Moens, 2009). The grammar is created for a certain type of discourse and language, thus it cannot be applied universally. Furthermore, it relies only on the explicit discourse markers.

Stab and Gurevych (2014) are approaching the parsing of the argumentative structure differently. Once the argument components are detected and classified they create a set of all argument component pairs between which the relations may exist as described by the applied model of argumentation. The learner is then trained to classify the relations between these component pairs. In practice this process can be combined with the argument component classification.

The features for the identification of relations between the argument structure components are generally shared with the argument component classification. For the description of features see Section 2.2.4.

Argument structure parsing problem can be formulated as a sequence tagging task. The goal of the classifier is then to label each word in an input sequence with a multi-component BIO tag. B marks the beginning of a component, I marks internal part, and O marks tokens that are not part of any component. The tag carries the information about the distance to a previous or a subsequent tag that it relates to, as well as the type of the relation. In theory any sequence labelling model can be applied for the task.

Finally, dependency parsing methods can be applied for parsing the argumentative structure.Eger et al. (2017) experiment with five different dependency parsers. These are **MST-Parser** - parser based on the search of the maximum spanning tree in a graph (McDonald et al., 2005); **Mate** - toolkit of statistical natural language processing tools that include among others a dependency parser (Bohnet and Nivre, 2012); **Kiperwasser** - parser based on bidirectional long short-term memory network (Kiperwasser and Goldberg, 2016); **LSTM-Parser** - long short-term memory parser (Dyer et al., 2015), and **LSTM-ER** - end-to-end relation extraction parser based on long short-term memory network (Miwa and Bansal, 2016). They report that **LSTM-ER** performs best for the task. LSTM-ER is a recurrent neural network based model that captures both word sequence and dependency tree substructure information.

## 2.3 Argument Mining for Low Resource Languages

As we have mentioned in Section 2.1, producing the training data for argument mining is a time-consuming and error-prone process. As the result, there are not many datasets available and most of them are in the English language. Fortunately, there are a number of approaches that can be used in order to train a model on one language (source language) and then carry out inference on another language (target language).

These approaches can be roughly divided into two types: **language projection** and **direct transfer**.

Language projection method can be described as follows. A learner is trained on a source language. Then the learner is applied on input data in the source language in order to produce labelled data, e.g. label tokens as elements of an argumentative structure. The labels obtained on the inference stage are then projected to a target language. Hence, we assume that the dataset that we apply the system to is an aligned multilingual dataset. Finally, the obtained labelled data in the target language can be used in order to train a separate learner. In a simpler setting we can have a multilingual training dataset, and project the existing labels from the source language to the target language. The exact configuration depends on the available data. Language projection has a number of inherent problems. First, it requires that one has a parallel multilingual dataset at hand. And thus we encounter the circular problem - the lack of such datasets. Second, the alignment on token-level is error prone. Although, if we design an argument mining system with the sentence as atomic unit, we can achieve lossless transfer.

Artetxe et al. (2017) apply language transfer method for the argument component identification task on the sentence level, the obtained results show agreeable level of performance.

Next, we are going to briefly describe direct language transfer approach. When using this method, a learner is trained on language-independent or shared features using the source language as the basis and then the learner is directly applied to the target language.

With that in mind it is possible to use **only language agnostic features**, such as the position of the argument component in the text (paragraph number), the position of the argument component in a separate paragraph and similar. However, the research of Stab and Gurevych (2017) shows that the models trained on lexical and syntactic features surpass the models trained on language independent features. Furthermore, using structural features would limit the application of the model to the texts of the same genre and structure, such as student essays.

It is possible to use word embeddings for the purpose language transfer. One can use either **bilingual embedding mappings** or **multilingual contextual word embeddings**. In order to produce bilingual mappings, one first learns word embeddings from monolingual corpora separately for each language. Then the transformation from one embedding space to another is learned using a bilingual dictionary. One of the methods for the generation of bilingual word embeddings is proposed by Mikolov et

al. (2013). Producing the bilingual dictionary for the task may be time-consuming. However, Artetxe et al. (2017) introduce an algorithm that can bootstrap from a small dictionary containing about 25 words and produce word embeddings with almost no bilingual data.

There exist at least two multilingual contextual word embedding models. One of them is Multilingual BERT released by Devlin et al. (2019). The other one is XLM-RoBERTa by Conneau et al. (2020).

Based on the training data it is possible to differentiate two types of language transfer. These are **zero-shot** language transfer and **few-shot** transfer. Zero-shot transfer is achieved by training the learner only on the data in the source language. Few-shot transfer is carried out by training the learner on mixed language datasets, where there are samples mostly in the source language with the addition of some samples in the target language as well.

Lauscher et al. (2020) apply multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) on the following natural language processing tasks: a) lower-level structured prediction tasks: part of speech tagging, dependency parsing, and named entity recognition, and b) higher-level language understanding tasks: natural language inference and question answering. The first perform zero-shot cross language transfer by training the respective models on the English language and then apply it on a variety of languages. Then they perform experiments with few-shot transfer. They show that zero-shot transfer is more successful for the language pairs with higher linguistic proximity. They further report that for lower-level tasks the few-shot transfer the results in the performance improvement by 14.11 and 26 percent. However, for the higher-level tasks the improvements are less pronounced: between 2.1 and 4.57 percent. Since the argument mining can be formulated as a sequence labelling task, similar to the tasks that Lauscher et al. (2020) experiment with. It means that zero- and few-shot cross lingual transfer can be applied to the argument mining task.

# Chapter 3

# Datasets Description

In the experimental part of our work we are using two datasets. These are persuasive essays dataset by Stab and Gurevych (2017) in the English language, hereinafter referred to as "persuasive essays". And film reviews dataset by Evensen (2020) in the Norwegian language, hereinafter referred to as "film reviews".

Further, we will discuss the two data sets and more detail and give a short comparison of them.

Persuasive essays dataset is based on a random sample of student essays submitted and published on a web service (essayforum.com). The dataset includes 402 texts in total. 80 texts were annotated by non-professional annotators and provided the material for Stab and Gurevych (2017) the study of inter-annotator agreement. The remaining part of the texts was annotated by a trained annotator and formed a core part of the dataset.

Stab and Gurevych (2017) are using annotation scheme comprising of the following five elements: major claim, claim-for, claim-against, premise-support, premise-attack. Major claim is the central position of an author with respect to the topic. Claim is a controversial statement that becomes valid or true in the presence of additional support, which attacks or supports a major claim. Premise is a reason given by an author for persuading readers of the claim. For a detailed description of argument component types refer to Section 2.2.3 of this thesis. Additionally, premises are marked with the relation information. They include a pointer to the respective claim that they attack or support.

Detailed persuasive essays dataset statistics are provided in table 3.1.

Argument components with stance labels ignored are distributed as shown in Table 3.2.

Chart 3.1 shows the distribution of argument components without considering a stance of a respective argument component. Premise-support and premise-attack are summed under type *Premise*, while *Claim-Against* and *Claim-For* are summed together under type *Claim*. It is important to point out that argument components of class *Premise* account for 64% of all argument components. This may pose a problem for the training and application of our models. Since the predictions are likely to get biased

| | |
|---|---|
| Total texts | 402 |
| Total tokens | 145898 |
| Total unique tokens | 8429 |
| Total argument components | 6021 |

Table 3.1: Persuasive essays dataset statistics.

| Argument component | Frequency |
|---|---|
| Premise | 3830 |
| Claim | 1499 |
| Major claim | 692 |

Table 3.2: Persuasive essays dataset. Argument components without stance.

towards the dominating class.

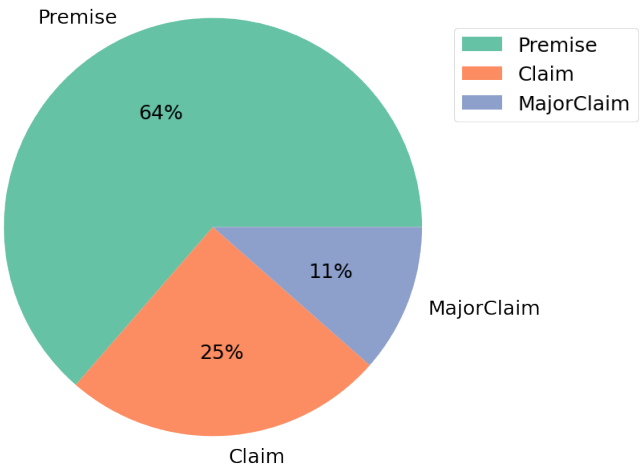



Figure 3.1: Persuasive essays dataset. Distribution of argument components.

Premise and support argument components are instantiated via the following subclasses: premise-support, premise-attack, claim-for, and claim-against (see Table 3.3).

In similar fashion we can observe that premises in support of claims and claims that are aimed to provide argument for the major claim outnumber in proportion other argument components.

All texts from persuasive essays dataset have peculiar features - these argumentative essays are written in academic style. Thus, they all share a similar structure. They have introduction, main part, and conclusion. We can also assume, that argument components may be distributed within text boundaries in a peculiar way. For example, we can expect that major claim appears early in the text and/or in its conclusion. If this hypothesis holds, we can use it as additional feature in our classification model.

We need a method to decide weather an argument component belongs

| Argument component | Frequency |
|---|---|
| Premise-Support | 3611 |
| Claim-For | 1226 |
| Major Claim | 692 |
| Claim-Against | 273 |
| Premise-Attack | 219 |

Table 3.3: Persuasive essays dataset. Argument components without stance.

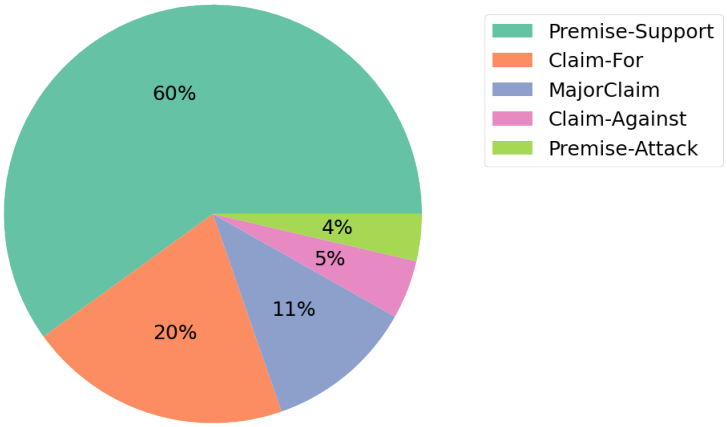



Figure 3.2: Persuasive essays dataset. Distribution of argument components 2.

to introduction, main part, or conclusion of an essay. After empirical examination of random texts from the persuasive essays dataset we came to the conclusion that there are no reliable ways to do this. Instead, we decided to take the following approach:

- For each sentence in a text we take its distance from the start of the text. In other words it is a serial number of the sentence in the text.
- We further normalize this number by the total number of sentences in the given text. As the result the distance of each sentence from the start of the text falls in range from 0 to 1.
- We further assign the distance of the respective sentence to each argument component contained in this sentence.
- Then, argument components are distributed among four ranges: 0 - 0.25, 0.25 - 0.5, 0.5 - 0.75, and 0.75 - 1. The first and the last range roughly represent introduction and conclusion, while the second and third ranges correspond to the main part of texts.
- We than take sums of all occurrences of argument components by their type within the defined ranges.

We can observe certain patterns in the distribution of argument components among these parts of the texts that we defined. For instance, *Premise-Support* argument component mainly occurs in the main part of the essays (Figure 3.4). *Claim-For* is rather evenly distributed across the text, with slightly less occurrences in the introduction (Figure 3.6). *Major Claim* almost exclusively appears only in the initial and conclusive parts of the texts (Figure 3.3). Interestingly, both *Claim-Against* and *Premise-Attack* have a tendency to appear towards the conclusion of the texts (Figures 3.7, 3.5). Thus, we can conclude that the distance of a candidate argument component can be considered a discriminating feature for the classification of argument components.

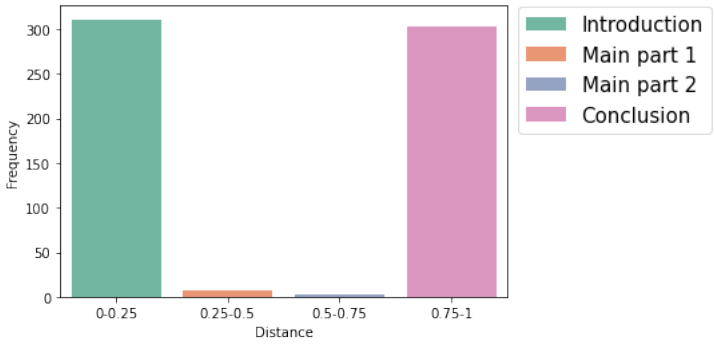



Figure 3.3: Persuasive essays dataset. Distribution of Major Claim argument components within text boundaries.

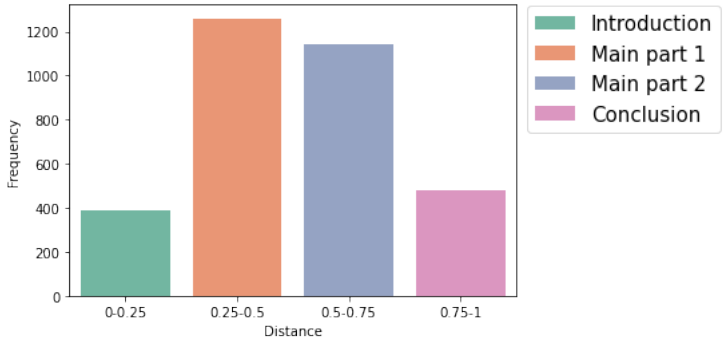



Figure 3.4: Persuasive essays dataset. Distribution of Premise-Support argument components within text boundaries.

Further, we perform similar analysis of the film reviews dataset. Film reviews dataset was created by Evensen (2020) based on random selection of texts from the screen category of Norwegian Reviews Corpus (Velldal et al., 2018) dataset. The latter includes 13,085 reviews of films. The reviews are written by a variety of authors, and do not adhere to a predefined schema or rules as compared to the argumentative essays.

Originally, the texts from NoReC dataset are presented in raw text format. Evensen (2020) performed preprocessing of the texts and converted them to the CoNLL-format format. The texts were further annotated by

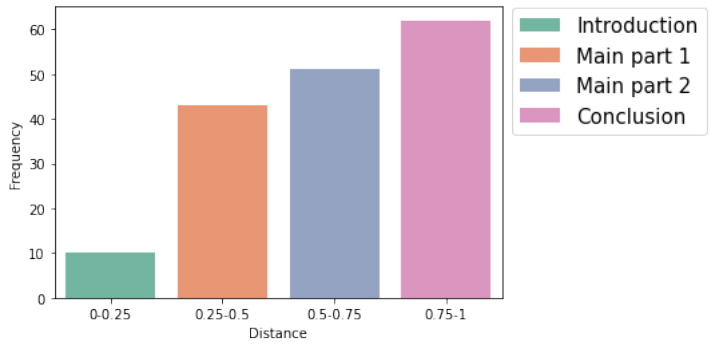



Figure 3.5: Persuasive essays dataset. Distribution of Premise-Attack argument components within text boundaries.

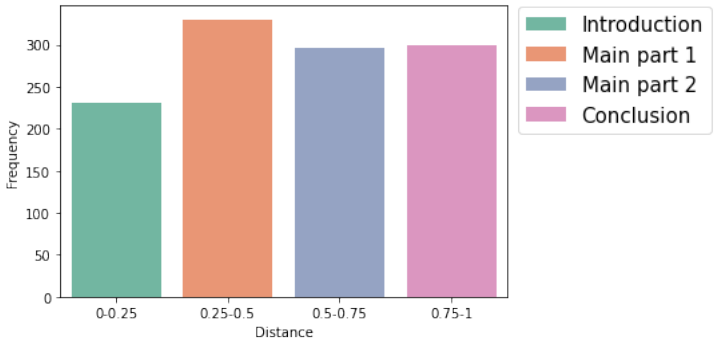



Figure 3.6: Persuasive essays dataset. Distribution of Claim-For argument components within text boundaries.

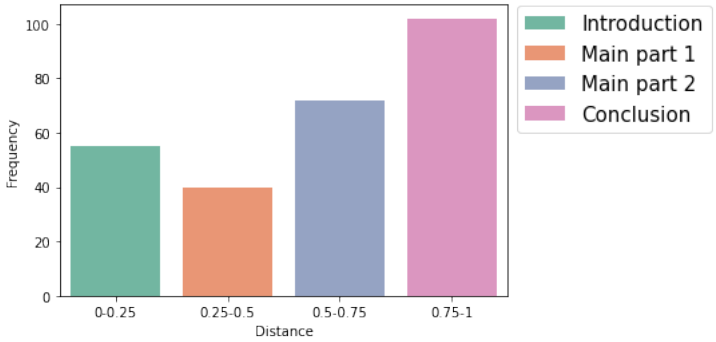



Figure 3.7: Persuasive essays dataset. Distribution of Claim-Against argument components within text boundaries.

non-professional annotators. The annotators used the annotation scheme and guidelines based on the work of Stab and Gurevych (2014).

Unlike persuasive essays dataset, the original dataset by Evensen (2020) includes not five but six argument component types. He also distinguishes *Claim* component along with *Claim-For* and *Claim-Against*. Evensen (2020) suggests to mark claims that barely describe the plot of a film as claims without stance. However, Bentahar et al. (2010) defines

| | |
|---|---|
| Total texts | 40 |
| Total tokens | 15878 |
| Total types | 4603 |
| Total argument components | 456 |

Table 3.4: Film reviews dataset statistics.

| Argument component | Frequency |
|---|---|
| Premise | 313 |
| Claim | 116 |
| Major claim | 27 |

Table 3.5: Film reviews dataset. Argument components without stance.

claim as an assertion or a conclusion presented to the audience and which has potentially a controversial nature. A bare description of the plot of a film falls under another argument component type, namely data, which is defined by Bentahar et al. (2010) as statements specifying facts or previously established beliefs related to a situation about which the claim is made. Since *Data* component is not part of the annotation model applied in persuasive essays dataset, we decided to treat all argument components marked as *Claim* in films dataset as non-argument component elements.

The total number of argument components in film reviews dataset amounts to 456 after the aforementioned adjustment was made (see Table 3.4), which is over ten times less than in persuasive reviews dataset.

Similarly to persuasive essays dataset, *Premise* argument components significantly outnumber other component types (see Table 3.5 and Figure 3.8).

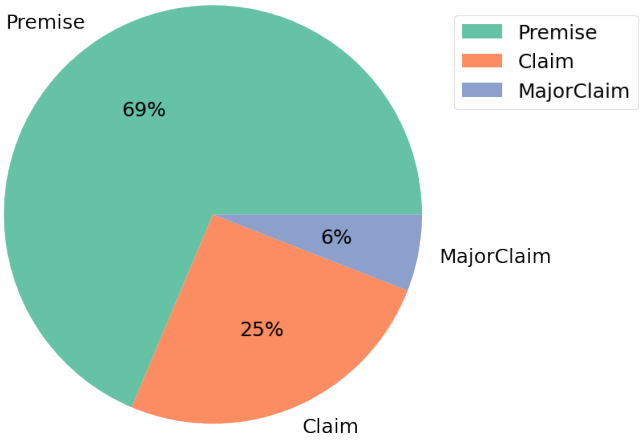



Figure 3.8: Film reviews dataset. Distribution of argument components.

Although texts from film data set generally do not follow any predefined structure and the authors were not constrained by any formal rules, we can observe that argument components show similar distribution patterns within the boundaries of a text. Namely, *Major Claims* appear mostly

| Argument component | Frequency |
|---|---|
| Premise-Support | 292 |
| Claim-For | 98 |
| Major Claim | 27 |
| Premise-Attack | 21 |
| Claim-Against | 18 |

Table 3.6: Persuasive essays dataset. Argument components without stance.

in the beginning and the end of the texts. Components with *-Support* and *-For* stance are rather evenly distributed, while components wit *-Attack* and *-Against* stance are more frequent in the concluding part.

Thus we can conclude that the despite the stylistic and language differences there exist structural similarities between the texts from the two datasets.

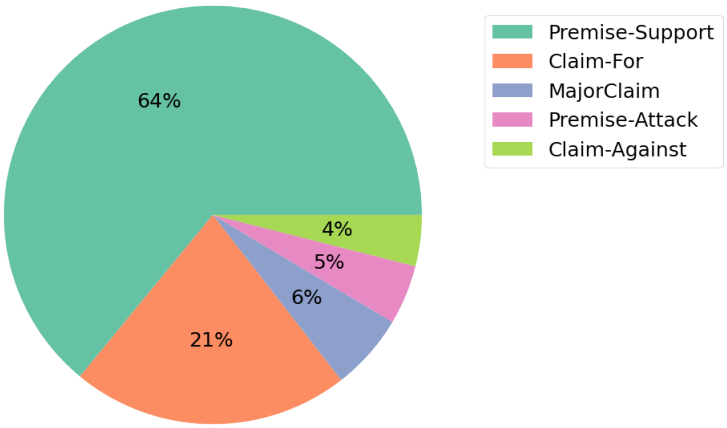



Figure 3.9: Film reviews dataset. Distribution of argument components 2.

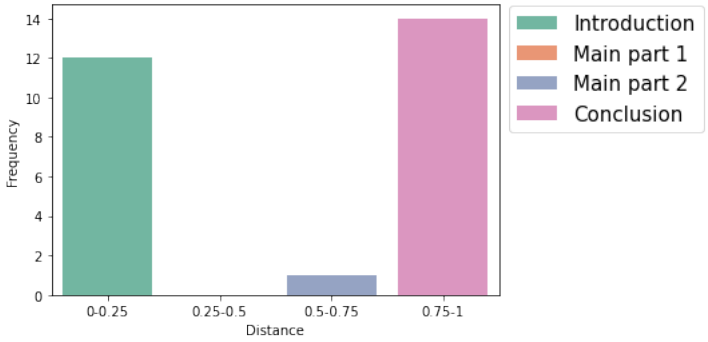



Figure 3.10: Film reviews dataset. Distribution of Major Claim argument components within text boundaries.

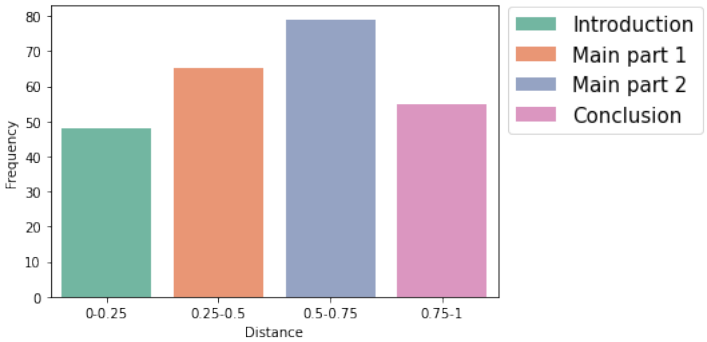



Figure 3.11: Film reviews dataset. Distribution of Premise-Support argument components within text boundaries.

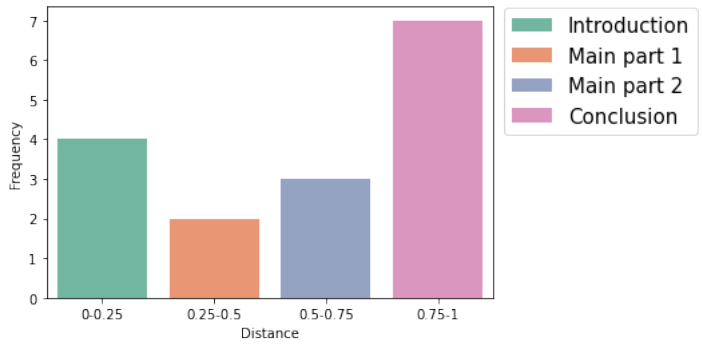



Figure 3.12: Film reviews dataset. Distribution of Premise-Attack argument components within text boundaries.

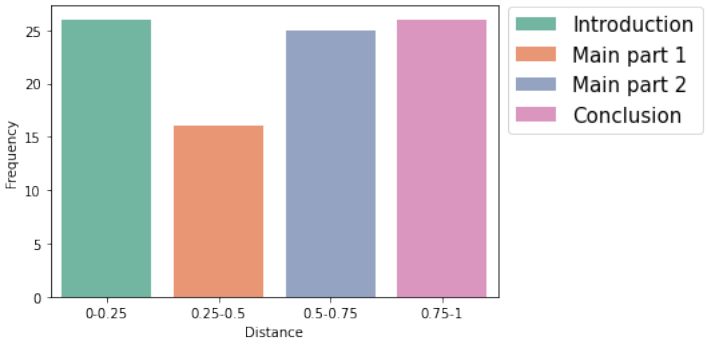



Figure 3.13: Film reviews dataset. Distribution of Claim-For argument components within text boundaries.

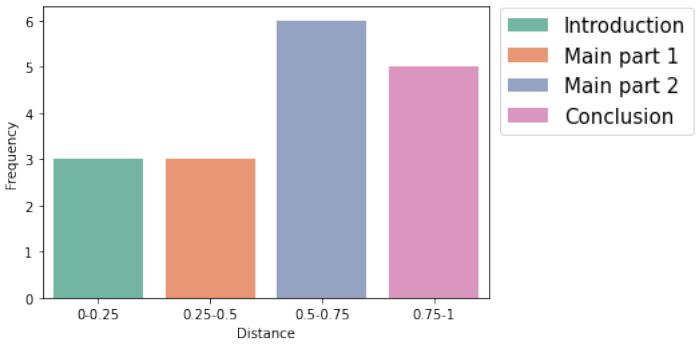



Figure 3.14: Film reviews dataset. Distribution of Claim-Against argument components within text boundaries.

# Chapter 4

# Experimental Set Up

## 4.1 Corpus Parsing

In our work we were using persuasive essay dataset files in CoNLL-format produced by Eger et al. (2017) and film reviews dataset files produced by Evensen (2020). The datasets are distributed as a collection of files where each file includes a stand alone document. Each token is presented on new line and annotated with a BIO-tag indicating the type of argument component, stance and eventually relation to another argument component. The relation is encoded as the distance from the current argument component to a related argument component. This distance is an offset in an array of argument components present in the current text. Sentences are dot-separated, each paragraph is separated from another by a line-break.

In order to parse these CoNLL files we created our own parser. Comprises of five classes: Text, Paragraph, Sentence, Argument Component, and Token. Each class is responsible for parsing and preprocessing the respective part of a document. See class diagram on Image 4.1. We chose this strategy instead of on-the-spot parsing because we plan to perform a number of experiments that would incur changes to the tagging scheme. For example, for argument component detection it is enough to annotate token with bare BIO tags, where for each tag we have Y:

$$Y = \{(b)|b \in \{B, I, O\}\}. \tag{4.1}$$

While for for the case of argument component detection and classification we will be using a scheme, where for each token we have a Y from:

$$\begin{aligned} Y = \{(b,t)|b \in \{B, I, O\}, \\ t \in \{MC, C, P, \perp\}\}. \end{aligned} \tag{4.2}$$

Or in more specific case as in:

$$
\begin{aligned}
Y = \{(b, t, s) | b \in \{B, I, O\}, \\
t \in \{MC, C, P, \perp\}, \\
s \in \{Sup, Att, F, A, \perp\}\}.
\end{aligned} \tag{4.3}
$$

Finally, in the case of argument component structure analysis one needs to annotate each token with a tag Y from:

$$
\begin{aligned}
Y = \{(b, t, s, d) | b \in \{B, I, O\}, \\
t \in \{MC, C, P, \perp\}, \\
s \in \{Sup, Att, F, A, \perp\}, \\
d \in \{..., -1, 0, 1, ..., \perp\}\}.
\end{aligned} \tag{4.4}
$$

Auxiliary classes that we produced are capable to generate tags with respective annotation scheme. Furthermore, they have methods that can produce documents of varying size: corresponding to the whole text, paragraphs or sentences.

## 4.2 Train and Test Datasets

Marsland (2009) recommends to split datasets into train, validation, and test subsets using the ratio 60:20:20 in situations where one does not have sufficient data. And he suggest using multi-fold cross-validation in situations where one is really short of training material.

We decided to follow this recommendation with some adjustments. First, we were not performing extensive hyper-parameters tuning, thus we decided not to use validation sets. Second, in experiments where only persuasive dataset was used for training such as zero-shot transfer. The persuasive essays data set was split into train and test subsets with ratio 80:20, and the whole film reviews dataset was used for the evaluation. In cases where the data from the film reviews dataset was used for training, we performed four fold cross-validation. The procedure was as follows:

- The whole persuasive essays dataset is used for training.
- Documents from the film reviews dataset are randomly split into four subsets.
- We performed four full cycles of model training evaluation for each of the four subsets. The training data comprised of the full persuasive essays data set and three subsets of the persuasive essays dataset.
- We reported average results of the four cross-validation experiments.

In order to evaluate the influence of the proportion of a low-resource language material in the training data we created fifteen additional training

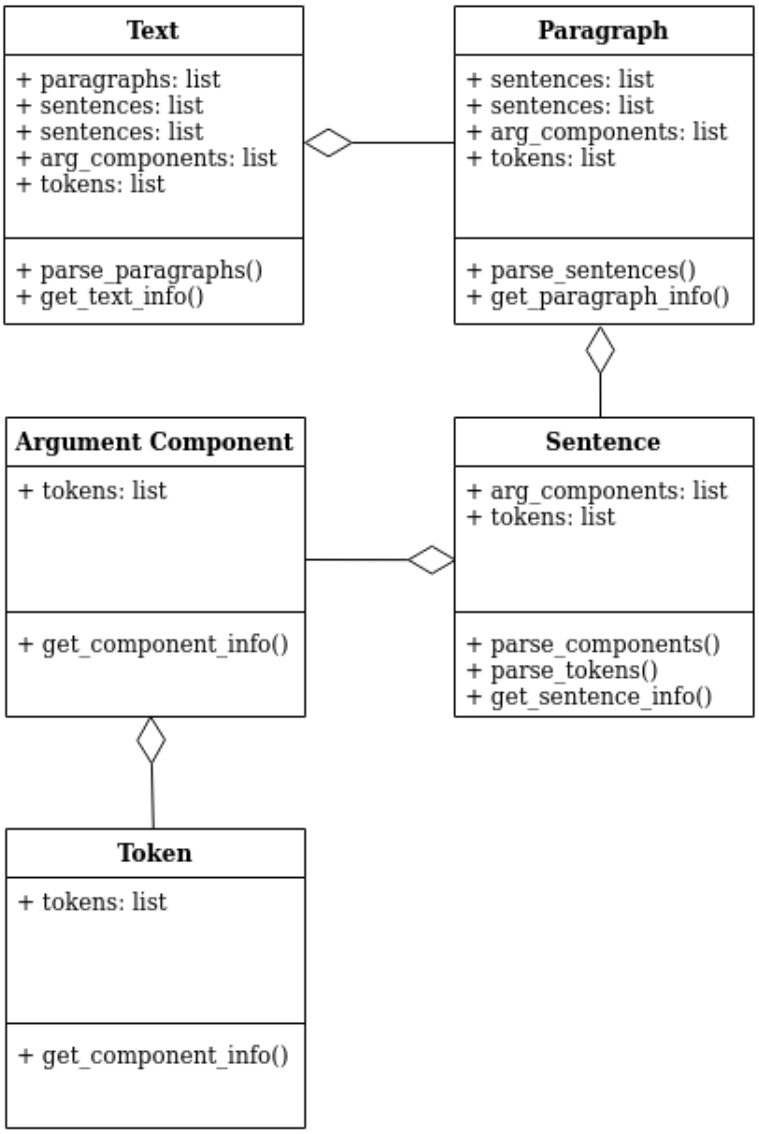



Figure 4.1: Data preprocessing. Class diagram.

datasets. They comprise of the whole persuasive essays dataset, and an increasing amount of the texts from the film review dataset: the first training set includes just one text from the film reviews dataset, the second training set includes two texts from the film reviews dataset and so on until the final fifteenth training set that includes fifteen texts from the film reviews dataset. Thus, each consequent dataset from this series includes a higher proportion of training material in Norwegian, which is a low-resource language in our set up. The models were evaluated on a test comprising of ten texts randomly chosen from the film reviews dataset.

## 4.3 PyTorch

Neural models are implemented using PyTorch (Paszke et al., 2019) machine learning library. PyTorch library includes tools for creating datasets and building deep neural architectures in imperative programming style. The library allows to run the code on CPUs as well as on GPUs. It is compatible with other packages used for machine learning tasks, such as NumPy (Harris et al., 2020) and Pandas (team, 2020). In our set-up, PyTorch handles the whole machine learning pipeline and includes the following components:

1. Dataset class. Is responsible for reading input texts and respective tags from CoNLL files. This class depends upon the parsing component presented in the previous section 4.1 of our work.

2. Model class. It is responsible for loading a respective pre-trained model.

3. Train procedure. This is a core procedure that handles training of a respective model. It handles learning epochs, batching, handling of hyper-parameters, intermediary evaluation of the learning process, storage of intermediary model state.

4. Evaluate procedure. This procedure handled evaluation of the models after additional learning and fine-tuning, as well as responsible for building charts and tables.

## 4.4 Neural Models

In our work we were using two pre-trained multilingual models. These are multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020). Both models are pre-trained models provided by HugginFace transformers library [1].

Both have transformer based architectures. Transformer based architecture was first introduced by Vaswani et al. (2017). It is a sequence transduction model. Before the transformer architecture was introduced, transduction models included complex recurrent or convolutional layers. The transformer architecture takes in use only attention mechanisms (Vaswani et al., 2017).

Multilingual BERT is a transformers model pre-trained on Wikipedia article in 104 languages. It was trained on raw texts without any sort of supervised input [2].

XLM-RoBERTa is pre-trained on 2.5TB of raw text data from CommonCrawl archive [3] in 100 languages.

RoBERTa has basically the same architecture as compared with BERT. However, the researches that developed RoBERTa model improved the

[1] https://huggingface.co/

[2] https://huggingface.co/bert-base-multilingual-cased

[3] https://commoncrawl.org/

approach to model training as compared with BERT. For instance, RoBERTa model is trained during a longer time-span using bigger batches as compared with BERT, RoBERTa model is trained on longer sequences. Next sentence prediction objective is removed in RoBERTa training. Furthermore, RoBERTa authors were applying dynamic masking patter to the input data (Liu et al., 2019).

### 4.4.1 Model Architecture and Hyper Parameters

We are experimenting with two model architecture setups. Both architectures share the following properties. They consist of Dataloader class. This class generates batches of inputs of a predefined size. In our case we chose to use batch of size 32. This choice was mostly dictated by performance considerations dictated by the hardware that was used for running the experiments. We experienced that larger batches often caused out of memory exceptions. Each batch consists of sentences of variable length, however, the sentences are padded to the size of the longest sentence in the given batch.

Each batch is passed through a specialized tokenizer that if necessary performs decomposition of non-frequent words into sub-words. Such tokenizers are shipped together with pre-trained transformer models. We are using the respective tokenizers provided by HugginFace transformers library [4].

The input then passes through pre-trained transformer layer. The output then passes a drop out layer. We chose drop out probability with the value 0.1. Drop out is a regularization technique. Its aim is to prevent overfitting of a model. The idea of this technique is as follows: with a certain chance (drop out rate) some elements of the input can be zeroed out. As the result inputs become more diverse, which is also helpful in the situations where less training data is available (Marsland, 2009). Finally, the input passes through linear classifier layer. Loss function is cross-entropy loss function. Learning rate is different for the transformer layer and the linear classifier. Transformer layer is trained with the learning rate of $3e-5$, while the linear classifier is trained with the learning rate of $1e-3$. We made this choice during the preliminary experiments, where we attempted to reduce the time required to train a model. We experienced that higher learning rate on the transformation layer produced less stable outputs. Performance scores could increase and decrease drastically from epoch to epoch. Thus we tried to achieve stability vs learning speed trade off by using different learning rates for the two layers.

Hyper parameters are summarized in Table 4.1. High-level architecture diagram is presented in Figure 4.2.

The architecture of the second model is similar to the architecture of the first one. However, it is accommodated for the usage of additional features. In Chapter 3 we pointed out that argument components of different types tend to appear more or less frequent in certain parts of a text.

[4] https://huggingface.co/

| Parameter | Value |
|---|---|
| Epochs | 35 |
| Batch size | 32 |
| Drop out rate | 0.1 |
| Learning rate (transformer layer) | $3e-5$ |
| Learning rate (linear layer) | $1e-3$ |

Table 4.1: Summary of hyper parameters.

Argument components of *Major Claim* type are observed more frequently in the beginning and the end of a text. We used this peculiarity as en extra structural feature. In the second type of architecture Dataloader class produces additional data. It is an array of the size of the batch, where each value represents the distance of the respective sentence from the beginning of a text. This data is concatenated with the output from the pre-trained transformer layer and fed further through the drop out layer and linear classifier. The higher level structure of this architecture is presented in Figure 4.3.

## 4.5 Model Selection

In order to avoid over-fitting of a model it is generally advised to used early stopping technique (Marsland, 2009). On the training stage, a model is evaluated each epoch on the validation subset. If the performance of the model starts deteriorating over next epochs, the training is stopped.

Given the fact that models involved in our work are of rather small-size and training a model over the next epoch does not take substantial time we decided to take an other approach. Each model is trained over 35 epochs. A model is evaluated and its state is saved for every epoch. Finally, we pick up the model that showed the best performance over 35 epochs based on the weighted average F1 score calculated over the labels (classes) of a current classification task.

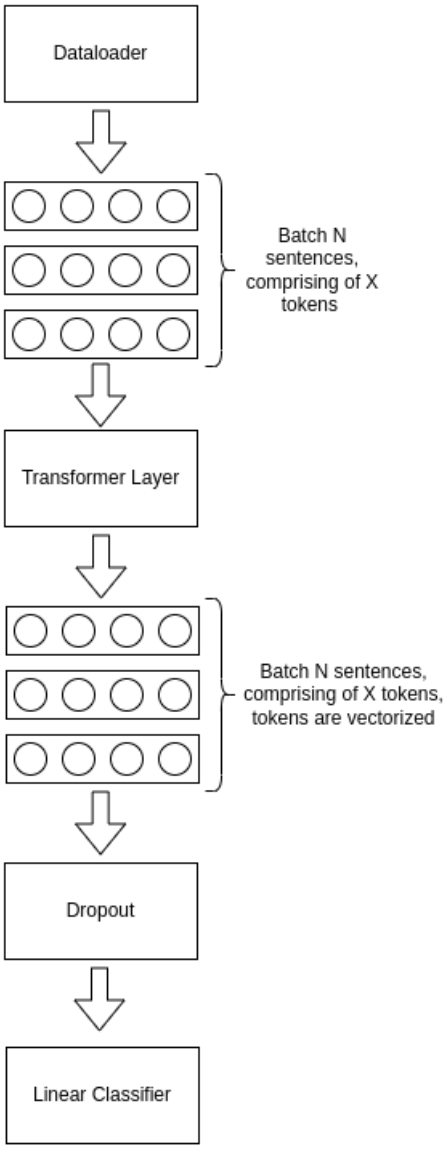



Figure 4.2: Model architecture 1.

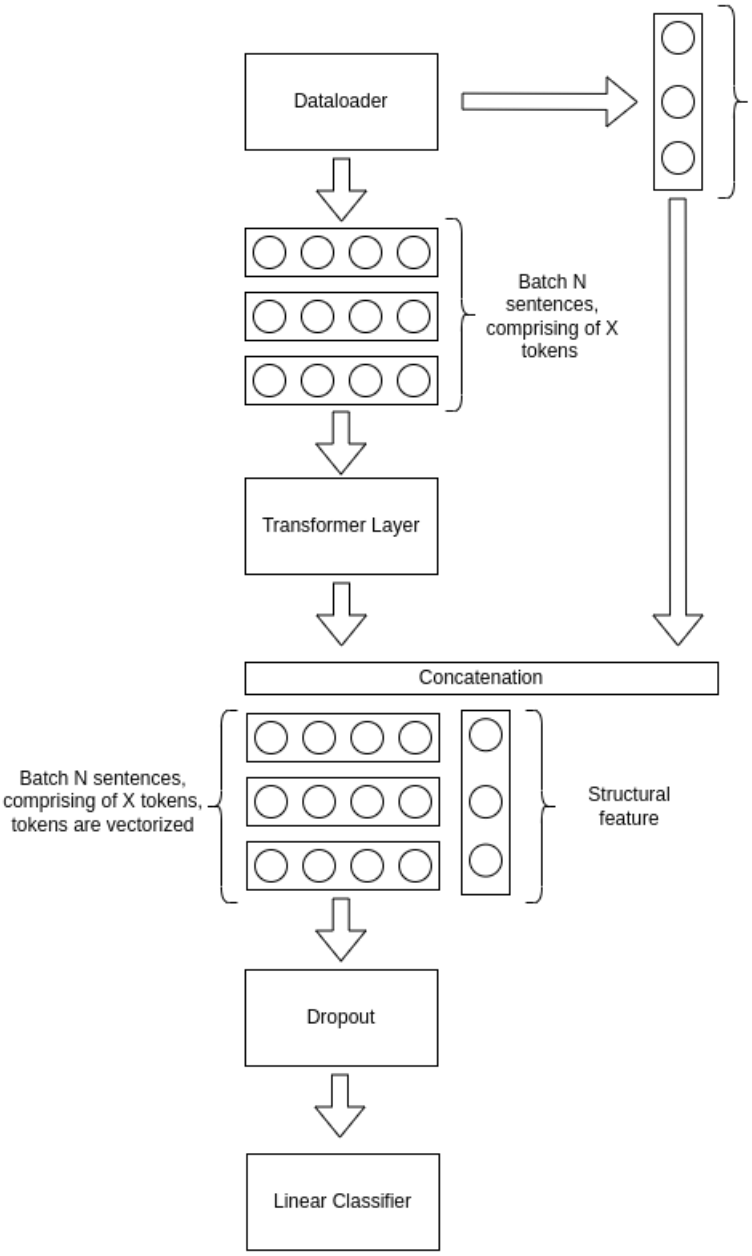



Figure 4.3: Model architecture 2.

# Chapter 5

# Results

## 5.1 General Notes

Each experiment is accompanied by the following descriptive material: a table containing a summary of model performance over 35 learning epochs, loss function chart, proportional (weighted) F1 score chart over 35 learning epochs, a summary table describing model performance, and a confusion matrix.

The performance summary table, loss function chart, and F1 score change chart were used in order to choose the best epoch that would be further analysed. Eventually, those could be used for fine tuning of model hyper-parameters. However, due to the limited amount of resources we decided to leave this task out of the scope of this work.

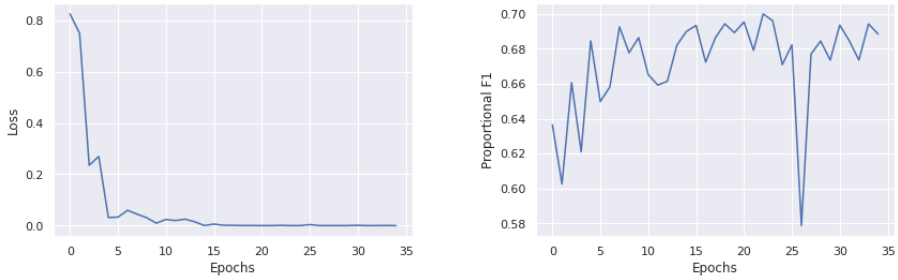
Each model performance summary table describes the performance of the best model chosen during the given experiment. It includes precision, recall, and F1-score for each label, accuracy, as well as macro, and micro (weighted) F1 score averages over all the labels.

In case of models where we applied k-fold cross-validation scheme, the tables that are summarizing model performance over the epochs include the data only from the first out of four folds. We omit the data for the remaining three steps because it does not contribute to the overall comprehension of the results. The final results summary tables, on the other hand, represent the mean over the four cross-validation experiments.

The data presented in confusion matrices is normalized so that each cell is marked with the percentage from total number of tokens included in the test dataset. This was done to improve readability and comprehension of the results, and to enable us to compare the results produced during zero-shot language transfer experiments vs few-shot transfer, and experiments where the models were both trained and evaluated on the Norwegian dataset.

## 5.2 Argument Component Identification

In this set of experiments we are training four models for the argument component detection task. Input sequences are split by sentences. Token




(a) Loss. (b) F1 score.

Figure 5.1: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component detection. Weighted F1 score and loss during model training.

labels are encoded using BIO tags. Tokens that are not part of an argument component are marked with O-label. Tokens that are part of an argument component are marked with B- and I-labels. Thus, there are three labels in total.

### 5.2.1 Models Trained and Evaluated on the Norwegian Dataset

The following models were trained and evaluated solely on the film reviews dataset in the Norwegian language.

#### Multilingual BERT - Model 1

While training the model, minimum we observed that weighted F1 score had was 0.5787, maximal value achieved was 0.6998, and it was 0.7623 on average. Highest weighted F1 score is achieved at epoch number 23. See Table A.1.

As seen on Figure 5.1, during the training of this model loss value substantially decreases after epoch number 5. Weighted F1 score reaches nearly maximal value after epoch number 5. After epoch number 24 weighted F1 score gradually reduces.

If we look at the confusion matrix (Figure 5.2) and the respective summary table (Table 5.3) we can notice that the model has a strong tendency to wrongly classify tokens labelled with O tag with I-Component tag. When it comes to I-Component tag the model reaches almost 0.89 in Recall score and 0.69 in Precision score.

#### Multilingual BERT - Model 2

Compared to the model trained without the use of extra feature (the relative distance of a token from the beginning of a given text) this model showed different behaviour during training. First, we observe that the difference between the minimum and maximum value of the weighted F1 score is almost two times bigger. Highest weighted F1 score is 0.4810 and highest is 0.7078. The average F1 score was 0.6758 on average. Highest weighted

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.82 | 0.59 | 0.68 | 1935 |
| B-Component | 0.61 | 0.61 | 0.6 | 128 |
| I-Component | 0.69 | 0.89 | 0.77 | 1905 |
| Accuracy | - | - | 0.73 | 3969 |
| Macro average | 0.71 | 0.70 | 0.69 | 3969 |
| Weighted average | 0.76 | 0.73 | 0.73 | 3969 |

Table 5.1: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component detection. Results evaluated on the epoch with best F1 score.

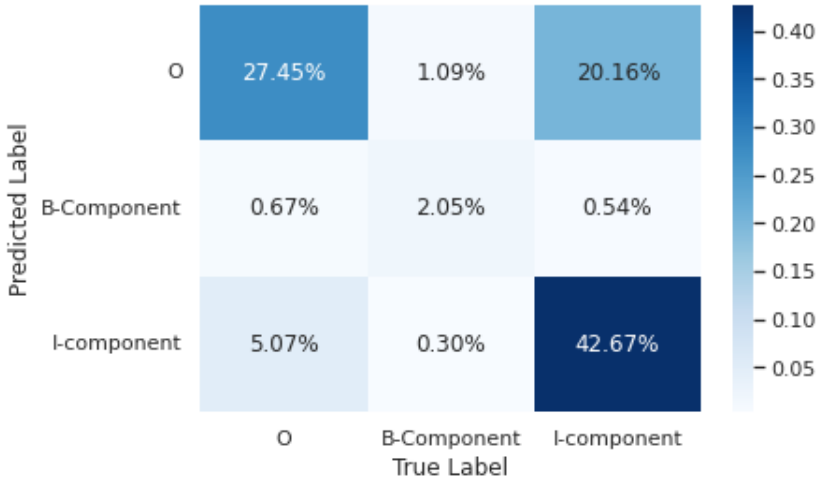
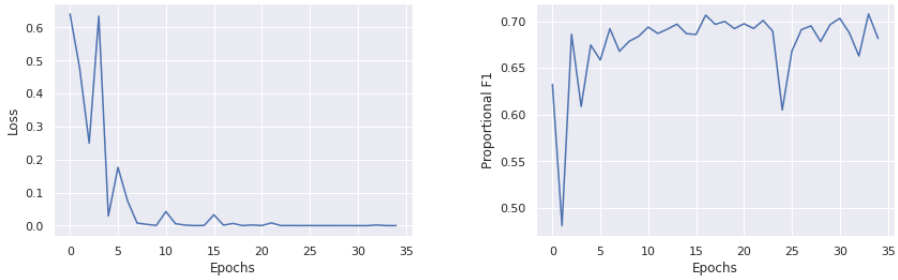



Figure 5.2: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features. Confusion matrix.

F1 score is achieved at epoch number 34. See Table See Table A.2. Second, the loss value drops and weighted F1 score reaches near maximal value at a later epoch. Third, weighted F1 score shows less fluctuation from epoch 6 through 23 (Figure 5.3).

Adding an extra feature (the relative distance of a token from the beginning of a given text) does not have a substantial effect on the results of the model in this task. We observe that the model has been slightly better at detecting tokens labelled with B-Component and I-Component tags (Table 5.2), however, simultaneously, we observe a slight decrease of F1 score for the tokens that are not part of argument component, and since these class is a major class it contributes more to the change of the weighted F1 score and we observe that the weighted average F1 score is less than for the previous model.

We observe a similar patter on the confusion matrix (Figure 5.4), namely, the model wrongly assigns I-Component label to quite a few tokens that are not part of any argument component.




(a) Loss. (b) F1 score.

Figure 5.3: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component detection. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.84 | 0.57 | 0.67 | 1935 |
| B-Component | 0.62 | 0.67 | 0.64 | 128 |
| I-Component | 0.69 | 0.9 | 0.78 | 1905 |
| Accuracy | - | - | 0.73 | 3969 |
| Macro average | 0.72 | 0.71 | 0.70 | 3969 |
| Weighted average | 0.77 | 0.73 | 0.72 | 3969 |

Table 5.2: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component detection. Results evaluated on the epoch with best binary F1 score.

### XLM-RoBERTa - Model 1

During the training of this model the loss value dropped to near zero values after 13 epochs of training (Figure 5.5). The spread between minimum and maximum values of weighted average F1 score is less than that of mBERT-based models. Minimum weighted F1 score achieved is 0.6049, maximal F1 score value during running the experiment amounted to 0.6986, weighted F1 score averaged to 0.6679. The best weighted average score was achieved during epoch number 27, as seen in Table A.3.

Based on the data from the model performance summary table (Table 5.3) and the confusion matrix (Figure 5.6) we can see that on the overall the model shows similar behaviour to the mBERT based models. The majority of the tokens that belong to I-Component class are labelled correctly. A small fraction of them is wrongly labelled with O class. Similarly to mBert based models, the model has a strong tendency to wrongly label O tokens with I-Component label.

### XLM-RoBERTa - Model 2

Adding an extra feature (the relative distance of a token from the beginning of a given text) changes drastically the behaviour of the model across the training epochs as compared to the same model trained without the extra

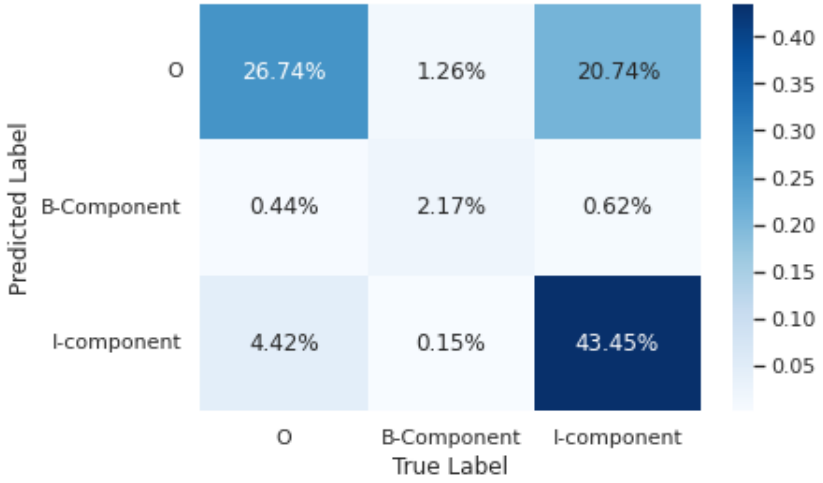
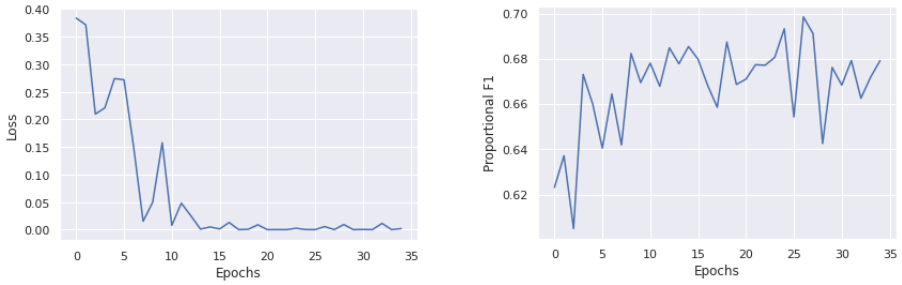



Figure 5.4: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.5: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component detection. Weighted F1 score and loss during model training.

feature. The spread between the F1 score of the worst and and the best epochs increases (Table A.4), the difference between minimal and maximal values is over 0.65, since F1 score of the worst epoch is as low as 0.01, while the maximal weighted F1 score that was reached during training of the model was 0.6692. Weighted F1 score average value is 0.5939. After epoch number 15 F1 score fluctuates only by a small margin and is almost unchanged (Figure 5.7).

XML-RoBERTa based model trained with extra features displays the same pattern of errors compared with the previously described models (Figure 5.8). It mostly correctly labels B- and I-Component tokens. However, it makes most errors in labelling tokens that are not part of any argument component. Recall score of B-Component class dropped from 0.74 to 0.68 (Table 5.4).

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.84 | 0.58 | 0.68 | 1935 |
| B-Component | 0.60 | 0.74 | 0.66 | 128 |
| I-Component | 0.69 | 0.89 | 0.77 | 1905 |
| Accuracy | - | - | 0.73 | 3969 |
| Macro average | 0.71 | 0.74 | 0.71 | 3969 |
| Weighted average | 0.76 | 0.73 | 0.73 | 3969 |

Table 5.3: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component detection.

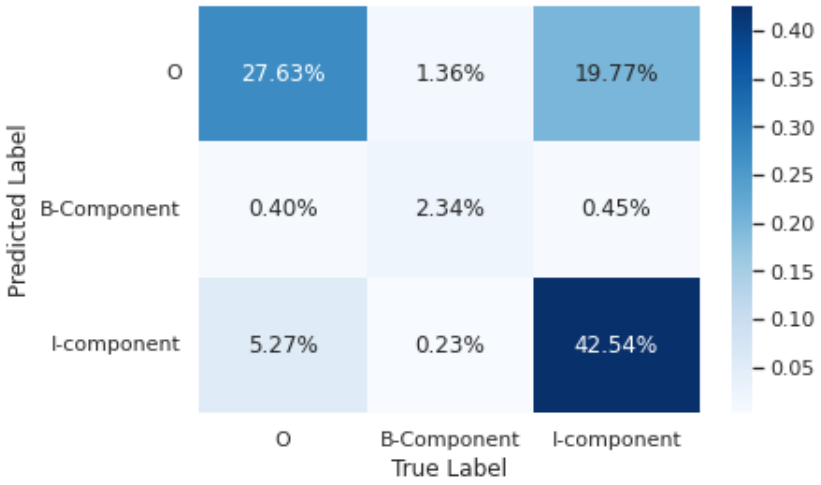



Figure 5.6: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features. Confusion matrix.

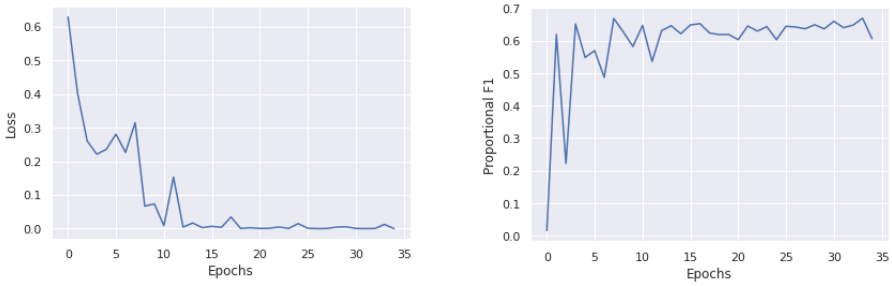
### Model Comparison

In general, the four models show similar results in argument component detection task (Table 5.5). F1 score for individual classes as well as weighted average F1 score values differ by about 0.01 across all the models.

Adding an extra feature (the relative distance of a token from the beginning of a given text) has a minor effect on the performance of the classifiers. However, they do show different behaviour if we compare the performance across the learning epochs. Both mBERT and XLM-RoBERTa based models with the extra feature reach near maximal weighted average F1 score after bigger number of epochs compared to the same models but without the extra feature. Thus, we can conclude that training a model with this extra feature will take extra time. However, since we do not get substantially higher F1 scores at the cost slower learning it does not pay back to use it provided current experimental set up and the task at hand.

All the four models tend to wrongly classify tokens that are not part of any argument component with I-Component label.

Three models (mBERT - Model 1, XLM-RoBERTa - Model 1, and XLM-




(a) Loss.

(b) F1 score.

Figure 5.7: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component detection. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.82 | 0.59 | 0.68 | 1935 |
| B-Component | 0.61 | 0.68 | 0.64 | 128 |
| I-Component | 0.69 | 0.88 | 0.77 | 1905 |
| Accuracy | - | - | 0.73 | 3969 |
| Macro average | 0.71 | 0.72 | 0.70 | 3969 |
| Weighted average | 0.76 | 0.73 | 0.73 | 3969 |

Table 5.4: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component detection.

RoBERTa - Model 2) showed the same weighted average F1 score. Thus we do not have a clear winner in this set up.

### 5.2.2 Zero-Shot Language Transfer

The following models were trained on the persuasive essays dataset in English and evaluated on the film reviews dataset in Norwegian. With this set of experiments we are evaluating the potential of using zero-shot language transfer for the task of argument component detection.

#### Multilingual BERT - Model 1

During the training of the model the loss value reached near zero values and weighted average F1 score reached near maximal values after epoch number 5 (Figure 5.9). However, if we compare the changes of loss value and weighted average F1 score to the ones observed during training of the models trained on the films dataset we can notice that they show more volatility. Thus, it might be problematic to apply early stopping techniques during the model training.

Minimal F1 score observed is 0.6177, maximal weighted F1 score achieved during the training was 0.6714. The average value of weighted F1 score amounts to 0.6518 (Table A.5).

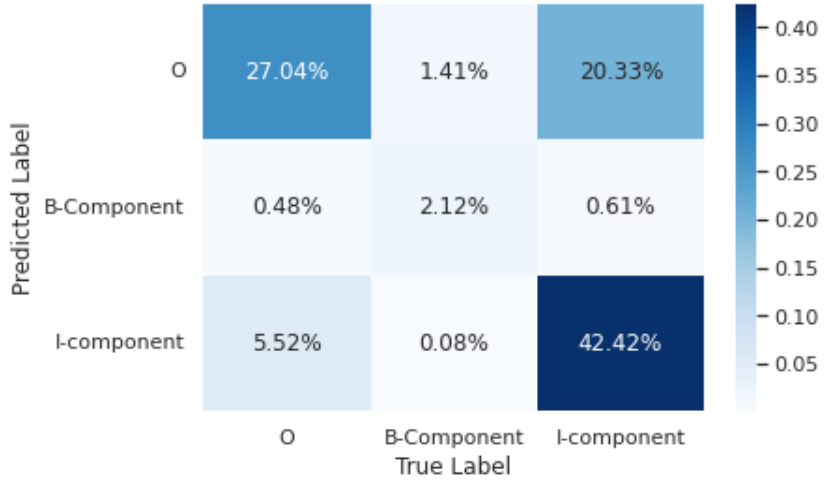



Figure 5.8: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features. Confusion matrix.

| | F1 - Score | | | |
|---|---|---|---|---|
| | mBERT 1 | mBERT 2 | XLM-RoBERTa 1 | XLM-RoBERTa 2 |
| O | 0.68 | 0.67 | 0.68 | 0.68 |
| B-Component | 0.6 | 0.64 | 0.66 | 0.64 |
| I-Component | 0.77 | 0.78 | 0.77 | 0.77 |
| Accuracy | 0.73 | 0.73 | 0.73 | 0.73 |
| Macro average | 0.69 | 0.70 | 0.71 | 0.70 |
| Weighted average | **0.73** | 0.72 | **0.73** | **0.73** |

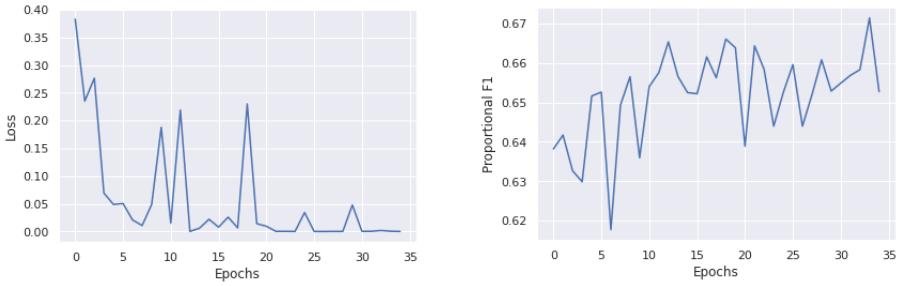
Table 5.5: F1 score comparison of models trained and evaluated on film reviews dataset in Norwegian, argument component detection.

The confusion matrix (Figure 5.10) and model performance summary table (Table 5.6) show that the model has a strong tendency to classify most of the tokens as I-Component. Compared to the same model trained solely on the film reviews dataset it has substantially worse recall when it comes to the classification of tokens that are not part of any argument components: 0.21 vs 0.59.

#### Multilingual BERT - Model 2

Adding an extra feature (the relative distance of a token from the beginning of a given text) to the model does not have any substantial influence on the learning process. Minimum and maximum weighted average F1 score change within 0.01. Minimal value of the weighted F1 score achieved during training amounted to 0.6168, maximal weighted F1 score reached 0.6671, while weighted F1 score averaged to 0.6472 (Table A.6).

The model is less stable. There are some spikes of loss value after




(a) Loss. (b) F1 score.

Figure 5.9: Model: mBERT, zero-shot transfer, no extra features, argument component detection. Weighted F1 score and loss during model training.

| | **Precision** | **Recall** | **F1-Score** | **Support** |
|---|---|---|---|---|
| O | 0.69 | 0.21 | 0.32 | 7740 |
| B-Component | 0.49 | 0.66 | 0.56 | 513 |
| I-Component | 0.54 | 0.91 | 0.68 | 7623 |
| Accuracy | - | - | 0.56 | 15876 |
| Macro average | 0.58 | 0.59 | 0.52 | 15876 |
| Weighted average | 0.61 | 0.56 | 0.50 | 15876 |

Table 5.6: Model: mBERT, zero-shot transfer, no extra features, argument component detection.

training epoch 25 (Figure A.6). The highest F1 score was achived by the model after the twelveth training epoch.

Adding an extra feature (the relative distance of a token from the beginning of a given text) has an adverse effect on the model performance. The recall for O-label drop by 0.02 (Table 5.11). We can observe that even more token that are not part of any argument component are being classified as I-Component. Simultaneously, the precision of labelling tokens that are part of an argument component decreased by 0.01.

We observe that about 40% of all tokens are wrongly classified as being a part of argument component (Figure 5.12).

### XLM-RoBERTa - Model 1

During the training of this model we observed that its performance reached nearly maximal values after epoch number 5 (Figure 5.13). It gradually increased from epoch number 5 though epoch 25. After epoch 25 the performance of the model becomes less stable and decreases. Minimal weighted F1 score that we achieved during training of this model was 0.6076, maximal weightded F1 score we observed was 0.6577, on the average weighted F1 score amounted to 0.6461, and standard deviation of the value is as low as 0.001 (Table A.7).

XML-RoBERTa based model displays nearly similar behaviour when it comes to the classification of tokens as compared with the mBERT based

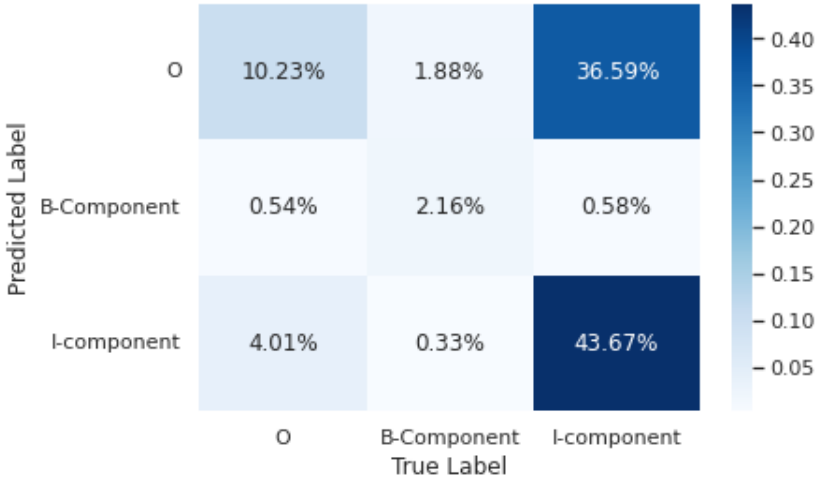
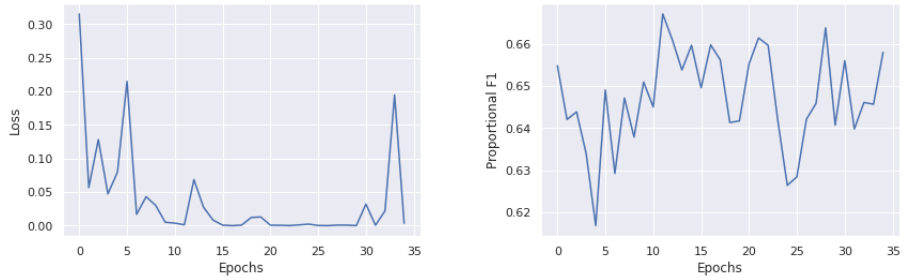



Figure 5.10: Model: mBERT, zero-shot transfer, no extra features. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.11: Model: mBERT, zero-shot transfer, with extra features, argument component detection. Weighted F1 score and loss during model training.

models. For instance, the majority of errors stem from misclassifications of tokens that are not part of any argument component (Figure 5.14). The recall for this type of label is as low as 0.21 (Table 5.8).

### XLM-RoBERTa - Model 2

Adding extra feature (the relative distance of a token from the beginning of a given text) does not substantially influence the behaviour of the model through learning epochs. Minimal weighted F1 score is 0.6336, maximal weighted F1 score observed during training is 0.6596, average weighted F1 score we observed is 0.6505 (Table A.8). The model reaches near maximal weighted average F1 score during the first two training epochs (Figure 5.15).

Adding extra feature (the relative distance of a token from the beginning of a given text) to the XLM-RoBERTa based model has a negative influence on its performance. The recall of tokens marked with O-label

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.67 | 0.18 | 0.29 | 7740 |
| B-Component | 0.49 | 0.66 | 0.55 | 513 |
| I-component | 0.53 | 0.92 | 0.67 | 7623 |
| Accuracy | - | - | 0.55 | 15876 |
| Macro average | 0.57 | 0.58 | 0.50 | 15876 |
| Weighted average | 0.60 | 0.55 | 0.48 | 15876 |

Table 5.7: Model: mBERT, zero-shot transfer, with extra features, argument component detection.

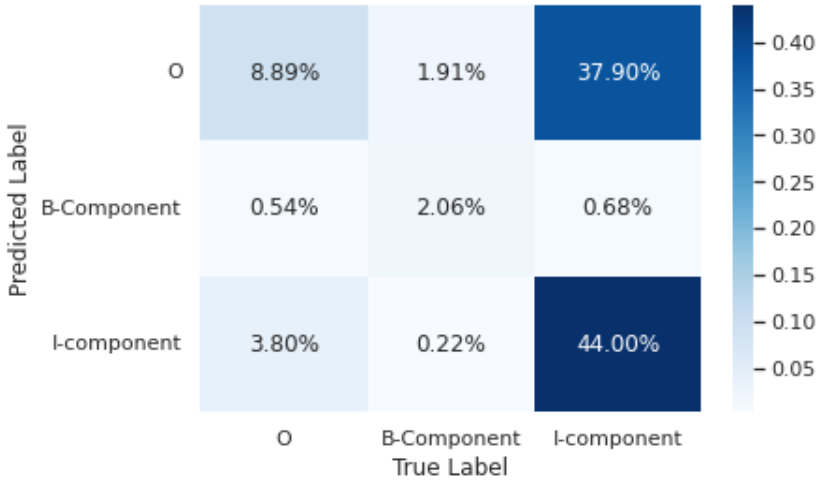



Figure 5.12: Model: mBERT, zero-shot transfer, with extra features. Confusion matrix.

further reduced. It dropped to 0.18 (Table 5.9). The majority of token that should have been labelled with O are being marked with B-Component and I-Component (Figure 5.16).

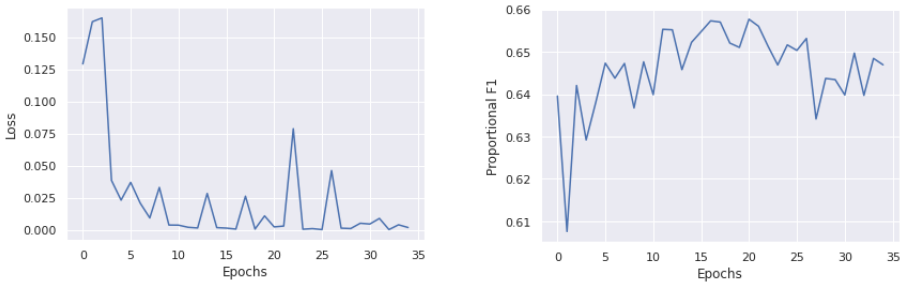
### Model Comparison

All the models trained on the persuasive essays dataset and evaluated on the film reviews dataset showed similar behaviour during training across 35 training epochs. They reach near maximal weighted average F1 score after 5 training epochs. The performance deteriorates after 25 training epoch.

All the models tend to classify most of the tokens with I-Component and B-Component labels.

Adding an extra feature (the relative distance of a token from the beginning of a given text) had an adverse effect on the model performance both for mBERT and XLM-RoBERTa based models.

Out of these four models mBERT - Model 1 showed best performance with weighted average F1 of 0.5. While XLM-RoBERTa - Model 2 was the worst with weighted average F1 score of 0.45 (Table 5.10).




(a) Loss.

(b) F1 score.

Figure 5.13: Model: XLM-RoBERTa, zero-shot transfer, no extra features, argument component detection. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.62 | 0.21 | 0.31 | 7740 |
| B-Component | 0.50 | 0.67 | 0.57 | 513 |
| I-Component | 0.53 | 0.88 | 0.66 | 7623 |
| Accuracy | - | - | 0.55 | 15876 |
| Macro average | 0.55 | 0.58 | 0.51 | 15876 |
| Weighted average | 0.58 | 0.55 | 0.49 | 15876 |

Table 5.8: Model: XLM-RoBERTa, zero-shot transfer, no extra features, argument component detection.

### 5.2.3 Few-Shot Language Transfer

The following models were trained on combination of the persuasive essays dataset in English and evaluated on the film reviews dataset in Norwegian. Each experiment was evaluated with 4-fold cross-validation. With this set of experiments we are evaluating the potential of using few-shot language transfer for the task of argument component detection.

#### Multilingual BERT - Model 1

Unlike other models we observed so far this showed significant drops of weighted average F1 score during some learning epochs (Figure 5.17). For example, during epochs 15, 25, and 30. Also, this model reaches nearly 0 value of the loss and nearly maximal weighted average F1 score earlier - after epoch number 3.

Minimal weighted average F1 score is 0.56, maximal weighted average F1 score is - 0.70. The average of weighted average F1 across all the training epochs 0.67. See Table A.9.

The model has a tendency to wrongly label tokens that do not belong to any argument component with I-Component label. Simultaneously, relatively small proportion of I-Component token is wrongly marked with O-label (Figure 5.18). This behaviour is generally in line with the models that we have trained and evaluated up to this point.

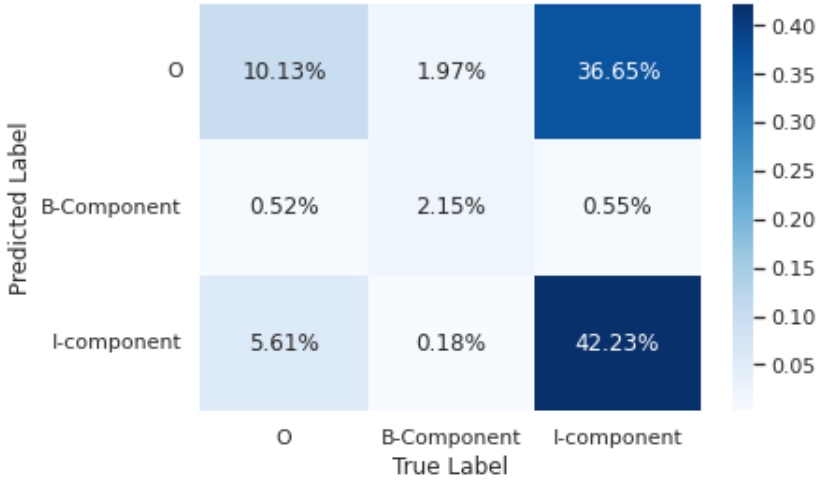
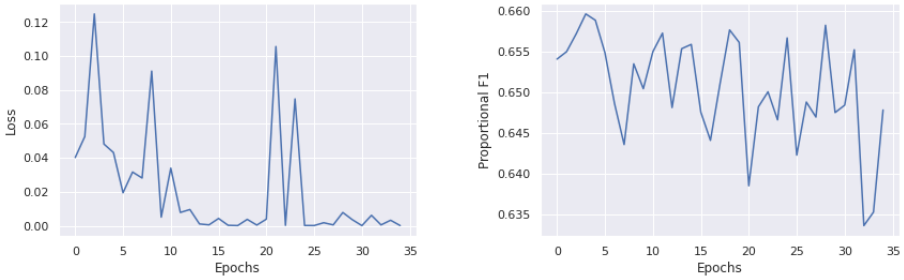



Figure 5.14: Model: XLM-RoBERTa, zero-shot transfer, no extra features. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.15: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component detection. Weighted F1 score and loss during model training.

This is further illustrated by Table 5.11. We can see that I-Component label has 0.87 in recall value, while B-Component and O have 0.67 and 0.67 respectively.

### Multilingual BERT - Model 2

Adding an extra feature (the relative distance of a token from the beginning of a given text) has some visible effect on the behaviour of the model across different learning epochs. The value of loss drops to near zero value and we do not observe any spikes through the remaining epochs (Figure 5.19a).

However, the value of weighted average F1 score is rather unstable and we can observe multiple peaks and drops across the learning epochs. Which in turn makes it problematic to apply early stopping techniques (Figure 5.19b).

Weighted average F1 score averages to 0.66 across all the training epochs with the standard deviation of 0.03. Maximal weighted average

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.62 | 0.14 | 0.23 | 7740 |
| B-Component | 0.44 | 0.70 | 0.54 | 513 |
| I-Component | 0.52 | 0.92 | 0.67 | 7623 |
| Accuracy | - | - | 0.53 | 15876 |
| Macro average | 0.53 | 0.59 | 0.48 | 15876 |
| Weighted average | 0.57 | 0.53 | 0.45 | 15876 |

Table 5.9: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component detection.

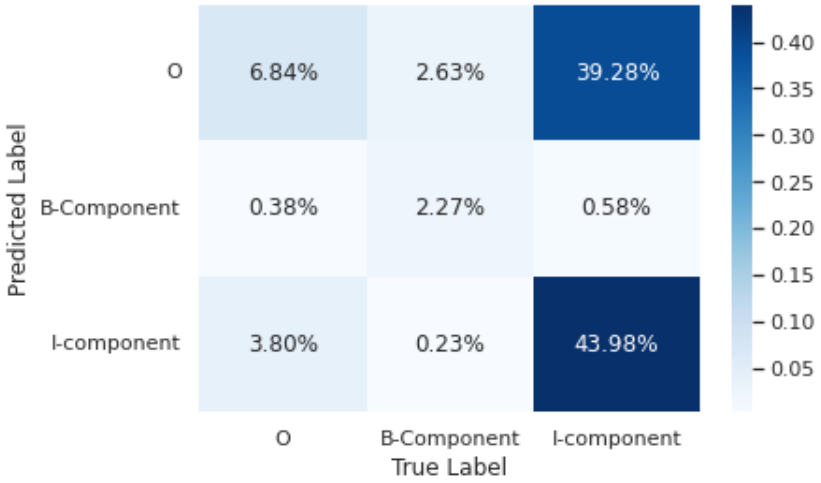



Figure 5.16: Model: XLM-RoBERTa, zero-shot transfer, with extra features. Confusion matrix.

F1 score achieved during training is 0.70, while the minimal value that we observed is 0.58 (Table A.10).

Adding an extra feature (the relative distance of a token from the beginning of a given text) does not cause substantial difference in the way the model classifies the tokens. The model has a better recall for B-Component labels (higher by 0.02) and I-Component labels (higher by 0.01). The recall and precision scores remain unchanged for O-label. See Table 5.12.

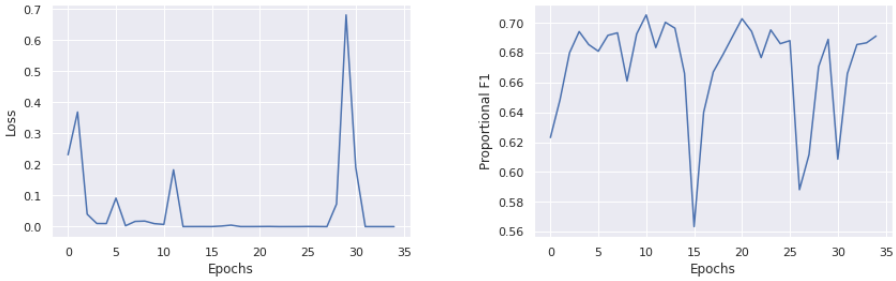
Confusion matrix (Figure 5.20) shows that the error pattern that we have observed in previous experiments persists.

#### XLM-RoBERTa - Model 1

The model shows a behaviour similar to the one we observed for the XLM-RoBERTa based models during the previous experiments. It reaches nearly maximal weighted average F1 score after epoch number 7 and the value remains rather stable up to the epoch number 25 (Figure 5.21). Minimal weighted average F1 score is 0.61, while weighted average F1 score maximum value that we observed is 0.70. The average value of

| | F1 - Score | | | |
|---|---|---|---|---|
| | mBERT 1 | mBERT 2 | XLM-RoBERTa 1 | XLM-RoBERTa 2 |
| O | 0.32 | 0.29 | 0.31 | 0.23 |
| B-Component | 0.55 | 0.64 | 0.57 | 0.54 |
| I-Component | 0.68 | 0.67 | 0.66 | 0.67 |
| Accuracy | 0.56 | 0.55 | 0.55 | 0.53 |
| Macro average | 0.52 | 0.50 | 0.51 | 0.48 |
| Weighted average | **0.50** | 0.48 | 0.49 | 0.45 |

Table 5.10: F1 score comparison of models trained on persuasive essays dataset in English and evaluated on film reviews dataset in Norwegian, argument component detection.




(a) Loss.

(b) F1 score.

Figure 5.17: Model: mBERT, few-shot transfer, with no extra features, argument component detection. Weighted F1 score and loss during model training.

weighted average F1 score across all the training epochs is 0.68 (Table A.11).

XLM-RoBERTa based model has higher precision (0.79) and recall (0.69) in classifying tokens that are not part of any argument component. However, compared to the mBERT based models the recall for I-Component is slightly worse. And we can observe that more I-Component tokens are wrongly marked with O-label. See Table 5.13.

Confusion matrix (Figure 5.22) further demonstrates that almost 20% of all tokens are wrongly classified as being part of an argument component and over 8% of tags are wrongly marked with O-label.

#### XLM-RoBERTa - Model 2

After adding an extra feature (the relative distance of a token from the beginning of a given text) to the model we observe that the value of the loss function settles down at later epochs. Namely after epoch number 10 and the model reaches its maximal weighted average F1 score value later, at epoch number 34 (Figure 5.23).

Maximal weighted average F1 score is 0.70, minimal weighted average

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.82 | 0.60 | 0.69 | 1935 |
| B-Component | 0.61 | 0.67 | 0.64 | 128 |
| I-Component | 0.69 | 0.87 | 0.77 | 1905 |
| Accuracy | - | - | 0.55 | 3969 |
| Macro average | 0.70 | 0.71 | 0.70 | 3969 |
| Weighted average | 0.75 | 0.74 | 0.73 | 3969 |

Table 5.11: Model: mBERT, few-shot transfer, with no extra features, argument component detection. 4-fold validation averages.

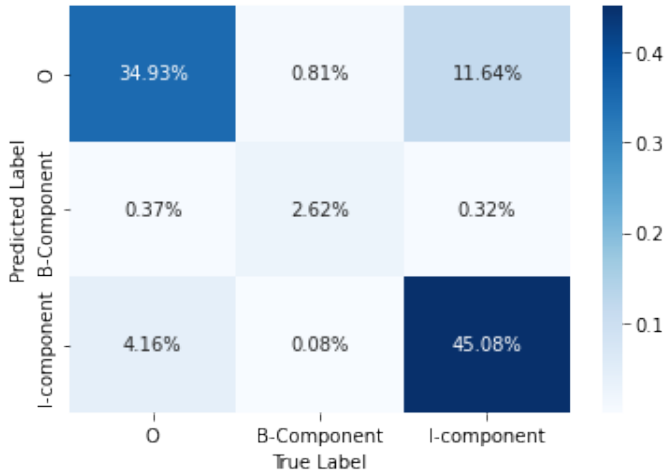



Figure 5.18: Model: mBERT, few-shot transfer, with no extra features. Confusion matrix.

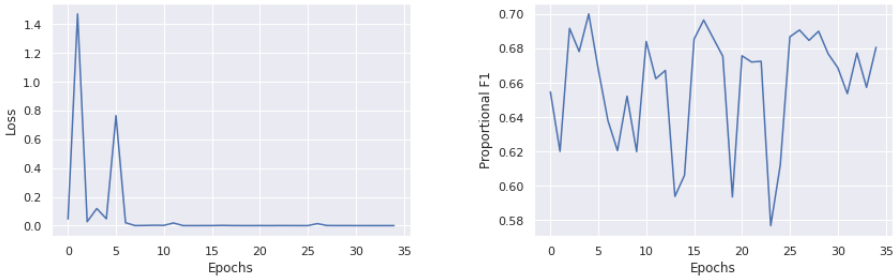
F1 score is 0.58, and it amounted to 0.67 on average across 35 training epochs (Table A.12).

Adding an extra feature (the relative distance of a token from the beginning of a given text) improves the precision for O labelled tokens: 0.83 vs 0.79. It also has a significant positive impact on the recall of I-Component labelled tokens: 0.87 versus 0.81 (Table 5.14). Although, we still observe the same kind of pattern, the model tends to wrongly classify O labelled tokens with I-Component label (Figure 5.24).

### Model Comparison

All the four models trained on the combination of English and Norwegian texts and evaluated on the Norwegian texts display similar error patterns. Most errors stem from the fact that the models wrongly classify quite a number of tokens that are not part of any argument component, they are marked with I-Component label.

Adding an extra feature (the relative distance of a token from the beginning of a given text) influences the behaviour of the models across the learning epochs. For instance, it takes more epoch to reach near




(a) Loss. (b) F1 score.

Figure 5.19: Model: mBERT, few-shot transfer, with extra features, argument component detection. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.82 | 0.60 | 0.69 | 1935 |
| B-Component | 0.60 | 0.69 | 0.64 | 128 |
| I-Component | 0.69 | 0.88 | 0.77 | 1905 |
| Accuracy | - | - | 0.73 | 3969 |
| Macro average | 0.71 | 0.72 | 0.70 | 3969 |
| Weighted average | 0.76 | 0.73 | 0.73 | 3969 |

Table 5.12: Model: mBERT, few-shot transfer, with extra features, argument component detection. 4-fold validation averages.

maximal weighted average F1 score. When it comes to the performance, the scores for mBERT based models are almost completely unchanged. The performance of XLM-RoBERTa based model with the additional feature turned out to be worse than the one without the additional feature.

The overall based result was demonstrated by XML-RoBERTa - Model 1. It reached 0.75 weighted average F1 score (Table 5.15).

## 5.3 Argument Component Classification

In this set of experiments we are training four models for the argument component classification task. Input sequences are split by sentences. Token labels are encoded using BIO labels. There are eleven labels in total. The goal of the modal is to identify argument component, assign argument component type, and stance to a token.

We are following the same procedure as before. We will carry out three sets of experiments with models trained and evaluated on Norwegian texts, zero-shot language transfer, and few-shot language transfer. In each set we are training four models: mBERT based, XLM-RoBERTa based, with and without additional features.

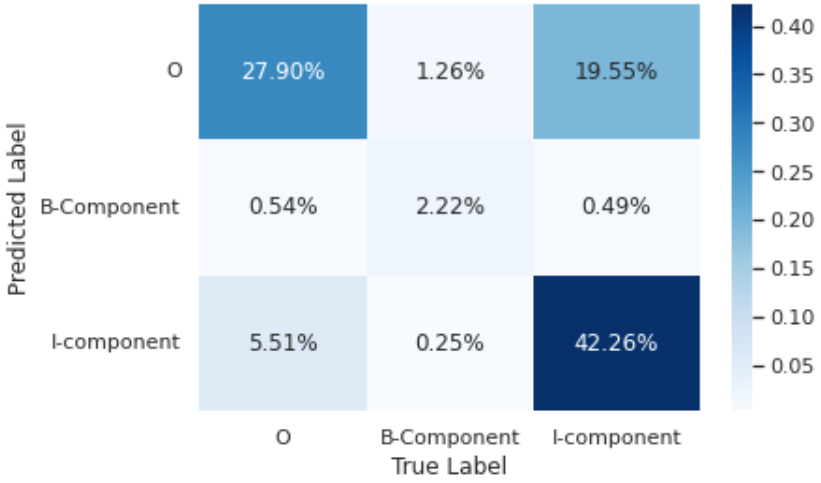
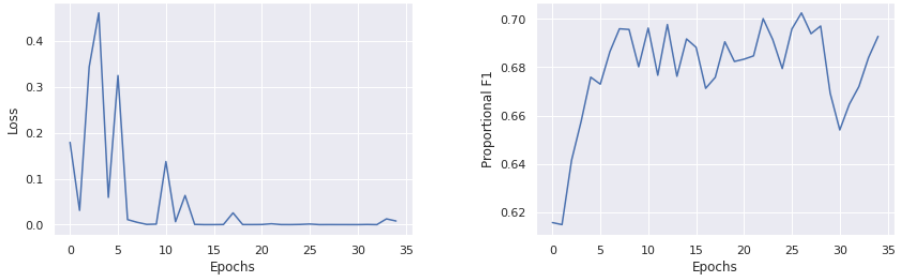



Figure 5.20: Model: mBERT, few-shot transfer, with extra features. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.21: Model: XLM-RoBERTa, few-shot transfer, with no extra features, argument component detection. Weighted F1 score and loss during model training.

### 5.3.1 Models Trained and Evaluated on the Norwegian Dataset

The following models were trained and evaluated solely on the film reviews dataset in the Norwegian language.

#### Multilingual BERT - Model 1

During the training of the model loss values dropped to nearly zero after ten training epochs (Figure 5.25a). During the first ten training epochs weighted average F1 score grows steadily. After that it gradually declines (Figure 5.25b). Minimal weighted average F1 score is zero, unlike the models that were trained for the task of argument component detection (Table A.13). In the latter case we observed that minimal F1 score values were observed rather early during training and they were always non-zero. Maximal weighted F1 score achieved during training over 35 epochs is 0.29, the average value of weighted average F1 score that we observed is 0.18

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.79 | 0.69 | 0.73 | 1935 |
| B-Component | 0.61 | 0.66 | 0.63 | 128 |
| I-Component | 0.73 | 0.81 | 0.77 | 1905 |
| Accuracy | - | - | 0.75 | 3969 |
| Macro average | 0.71 | 0.72 | 0.71 | 3969 |
| Weighted average | 0.76 | 0.75 | 0.75 | 3969 |

Table 5.13: Model: XLM-RoBERTa, few-shot transfer, with no extra features, argument component detection. 4-fold validation averages.

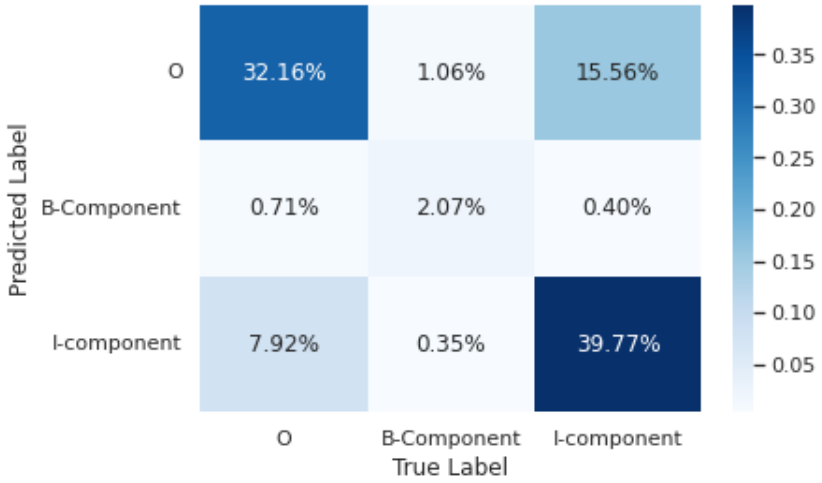


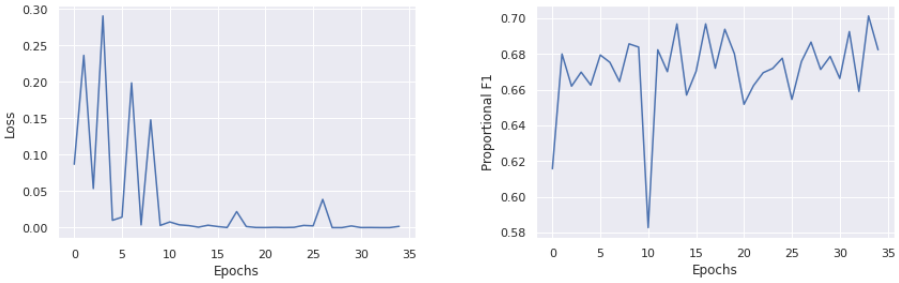

Figure 5.22: Model: XLM-RoBERTa, few-shot transfer, with no extra features. Confusion matrix.

(Table A.13).

The analysis of the confusion matrix (Figure 5.26) and the summary table (Table 5.16) shows that the model is not capable to distinguish components with stances "Attack" and "Against". Precision and recall values are zero for all four labels carrying these types of stance. It is also worth to notice that the model does not tend to wrongly label with these types of stance except for most frequent labels, namely O, I-Premise-Support, and I-Claim-For. Most typical errors are caused by misclassification of among the aforementioned most frequent labels.

### Multilingual BERT - Model 2

When we added an extra feature (the relative distance of a token from the beginning of a given text) we observed that the model needed more training epochs to reach nearly zero loss values and nearly maximal weighted average F1 score (Figure A.14). Thus, minimal training time of the model extends. Also, the results across the epochs became less stable. Standard deviation raised from 0.03 to 0.05. Minimal weighted average F1 score is 0, while maximal weighted average F1 score is 0.31, and we




(a) Loss. (b) F1 score.

Figure 5.23: Model: XLM-RoBERTa, few-shot transfer, with extra features, argument component detection. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.83 | 0.62 | 0.71 | 1935 |
| B-Component | 0.55 | 0.65 | 0.59 | 128 |
| I-Component | 0.70 | 0.87 | 0.78 | 1905 |
| Accuracy | - | - | 0.75 | 3969 |
| Macro average | 0.70 | 0.71 | 0.69 | 3969 |
| Weighted average | 0.76 | 0.75 | 0.74 | 3969 |

Table 5.14: Model: XLM-RoBERTa, few-shot transfer, with extra features, argument component detection. 4-fold validation averages.

observed about 0.02 improvement as compared with the result obtained during the previous experiment. Weighted average F1 score averaged to 0.21 across 35 training epochs (Table A.14).

Adding an extra feature (the relative distance of a token from the beginning of a given text) had a substantial positive effect on the precision of the model for the classification of B-MajorClaim and I-MajorClaim. The precision value raised from 0.12 to 0.22 for B-MajorClaim and from 0.25 to 0.37 for I-MajorClaim. Although simultaneously we observe the decrease in recall by 0.02 and 0.06 respectively. We also observe some improvements for I-Premise-Support label. Also, this model detects I-Premise-Attack tokens. Although precision and recall are still very low: 0.03 and 0.02 respectively. See Table 5.17.

A total of 9.37% of tokens are labelled with O-label instead of I-Premise-Support. And 12.10% of tags that should have been marked with I-Promise-Support get O-labelled. These are the most prominent errors, as seen in the confusion matrix in Figure 5.28.

#### XLM-RoBERTa - Model 1

As compared to mBERT based models that we considered for the current task, the performance of XLM-RoBERTa based model remains close to zero during the first five training epochs. Afterwords it grows steadily. The

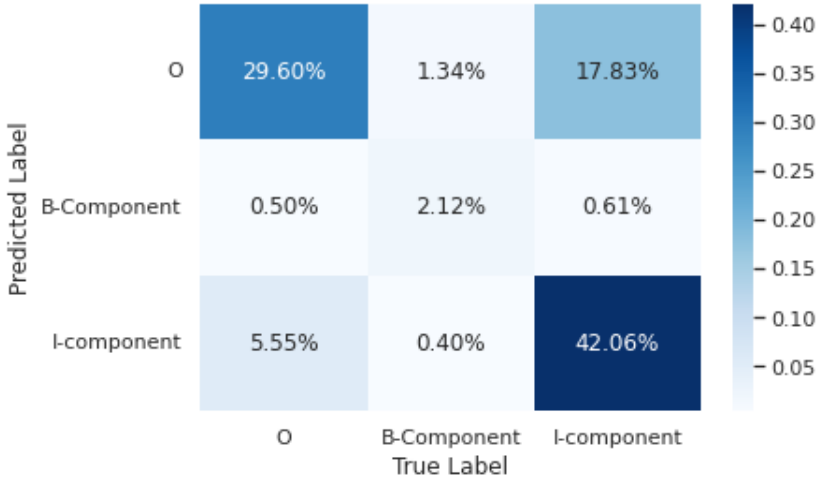



Figure 5.24: Model: XLM-RoBERTa, few-shot transfer, with extra features. Confusion matrix.

| | F1 - Score | | | |
|---|---|---|---|---|
| | mBERT 1 | mBERT 2 | XLM-RoBERTa 1 | XLM-RoBERTa 2 |
| O | 0.69 | 0.69 | 0.73 | 0.71 |
| B-Component | 0.64 | 0.64 | 0.63 | 0.59 |
| I-Component | 0.77 | 0.77 | 0.77 | 0.78 |
| Accuracy | 0.55 | 0.53 | 0.75 | 0.75 |
| Macro average | 0.70 | 0.70 | 0.71 | 0.69 |
| Weighted average | 0.73 | 0.73 | **0.75** | 0.74 |

Table 5.15: F1 score comparison of models trained on the mix of persuasive essays dataset in English and film reviews dataset in Norwegian, evaluated on film reviews dataset in Norwegian, argument component detection.

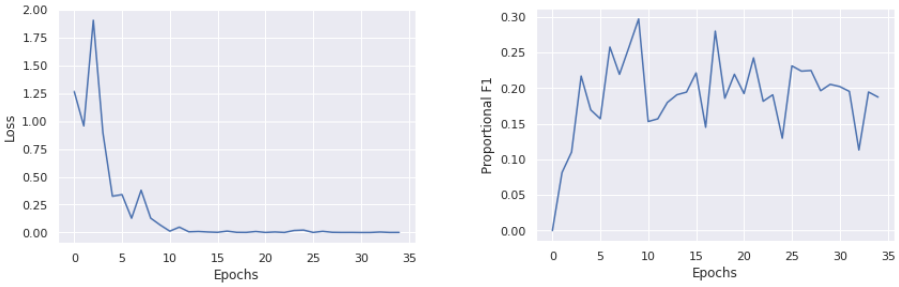
results stop improving after epoch number 23 as seen in Figure A.15.

The average value of weighted F1 score for 35 learning epochs is 0.1626. Maximal average F1 score value that we observed amounts to 0.23, while the minimal value is 0.16 (Table A.15).

The model poorly classifies tokens that belong to B-MajorClaim class (Table 5.18). This is something we have not observed in mBERT based models. However, it is somewhat capable to classify labels with "Against" stance. It is worth noting that the confusing happens mainly among the three most numerous classes, which is similar to the behaviour of mBERT based models (Figure 5.30).

#### XLM-RoBERTa - Model 2

Adding an extra feature (the relative distance of a token from the beginning of a given text) made the performance of the model less stable across the epochs. While the weighted average F1 score of the previous model tends




(a) Loss.

(b) F1 score.

Figure 5.25: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification. Weighted F1 score and loss during model training.

to grow steadily, in this case we observe some peaks with the difference up to 0.1 (Figure 5.31).

Standard deviation of weighted F1 score across training epochs is somewhat higher than during the previous experiments and it amounts to 0.063. Weighted average F1 score on average amounts to 0.13 across 35 training epochs, maximal value achieved is 0.23, while minimal value is zero (Table A.16).

The most prominent positive effect of adding an extra feature (the relative distance of a token from the beginning of a given text) is observed when it comes to the classification of Major Claim argument components. There is a substantial increase in precision and recall both for B-MajorClaim and I-MajorClaim labels. We also observe an in improvement in classification of claims with Against stance. On the overall, weighted average F1 score improved by 0.03 as compared with the model with no extra features included. See Table 5.19 and Figure 5.32.

**Model Comparison**

When it comes to the training process, we observed that XLM-RoBERTa based models required more training epochs to reach near maximal weighted average F1 score. Adding an extra feature (the relative distance of a token from the beginning of a given text) after the transformer layer makes performance of both mBERT and XLM-RoBERTa based models more volatile across training epochs.

All the models demonstrate similar confusion pattern: errors mostly occur among most numerous classes. For example, many O labelled tokens are wrongly labelled with I-Claim-For or I-Premise-Support. All the four models are poor at classifying token with Against and Attack stances.

Adding extra feature (the relative distance of a token from the beginning of a given text) positively influenced precision and recall for Major Claim type of argument component. XLM-RoBERTa based model showed higher sensitivity to adding the extra feature. XLM-RoBERTa - Model 1 surpassed XLM-RoBERTa - Model 2 in terms of F1 score across

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.73 | 0.66 | 0.69 | 2143 |
| B-MajorClaim | 0.12 | 0.13 | 0.13 | 6 |
| I-MajorClaim | 0.25 | 0.24 | 0.22 | 107 |
| B-Claim-For | 0.18 | 0.23 | 0.18 | 24 |
| I-Claim-For | 0.22 | 0.38 | 0.27 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 50 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 5 |
| I-Premise-Attack | 0.0 | 0.0 | 0.0 | 72 |
| B-Premise-Support | 0.37 | 0.34 | 0.33 | 73 |
| I-Premise-Support | 0.5 | 0.5 | 0.49 | 1159 |
| Accuracy | - | - | 0.55 | 3969 |
| Macro average | 0.22 | 0.23 | 0.21 | 3969 |
| Weighted average | 0.58 | 0.55 | 0.55 | 3969 |

Table 5.16: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification.

all the classes (Table 5.20).

### 5.3.2 Zero-Shot Language Transfer

In the section below we describe the models trained on the persuasive datasets model in English language and evaluated on the film reviews dataset in Norwegian language for the task of argument component classification.

#### Multilingual BERT - Model 1

Zero-shot model as compared with its counterpart that was trained on Norwegian film reviews behaves differently across the training epochs. F1 score fluctuates across the training epochs. However, we can notice that a general pattern that we observed in previous models holds. Namely, the model achieves best results in the range from epoch 10 to epoch 25 (Figure 5.33). Weighted F1 score averaged to 0.0381 across 35 training epochs. Maximal weighted average F1 score that was achieved during training of the model is 0.1083, minimal value of the weighted average F1 score is zero (Table A.17).

Unlike the mBERT based model that was trained on Norwegian film reviews dataset, this one somewhat detects argument components with Attack stance, but F1 scores are still very low: 0.04 for and 0.07 for B-Premise-Attack and I-Premise-Attack labels respectively (Table 5.21).

The confusion matrix (Figure 5.34) demonstrates that the majority of tokens that are not part of any argument component are marked with I-Premise-Support label and this accounts for most model errors.

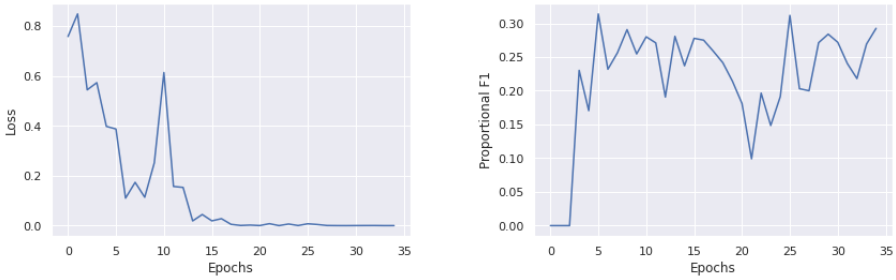

Figure 5.26: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification. Confusion matrix.

### Multilingual BERT - Model 2

Looking at loss and F1 score charts (Figure 5.35) During training the model shows similar behaviour as we observed in the previous model without any specific phenomena.

Maximal F1 score achieved is 0.1079 and it was 0.0522 on the average, while the minimal weighted average F1 score is zero (Table A.18).

The influence of an extra feature (the relative distance of a token from the beginning of a given text) to this model are very different from what we observed after the evaluation of the mBERT based model trained on the persuasive essays. While in the latter case we observed almost no changes in terms of performance per label, in this case we see that the model fails to classify Major Claim components (Figure 5.36), however, it performs better when it comes to the classification of components with Against and Attack




(a) Loss. (b) F1 score.

Figure 5.27: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification. Weighted F1 score and loss during model training.

stance. It is also worth to mention, that I-Claim-For class, the third most numerous class, has very low recall value of 0.1 (Table 5.22).

Almost 40% of all tokens where wrongly tagged with I-Promise-Support label.

#### XLM-RoBERTa - Model 1

This XLM-RoBERTa based model reaches higher F1 scores at later epochs (Figure 5.37). It reaches the maximum F1 score with value 0.1051 only after the 25-th training epoch. The average weighted F1 score is as low as 0.0309, while the minimal value of weighted average F1 score is zero (Table A.19).

The model does not detect Major Claim components and components with Against stance (Figure 5.38). The majority of tokens are labelled with I-Premise-Support label. Thus it demonstrates high recall for this class - 0.81 and low precision - 0.34 (Table 5.23).

#### XLM-RoBERTa - Model 2

With the extra feature (the relative distance of a token from the beginning of a given text) added the model shows higher F1 scores during earlier learning epochs, namely, from epoch 5 to epoch 15 (Figure 5.39). Average weighted F1 score across training epochs is as low as 0.0273 with a maximum of 0.0792, and similarly to other experiments performed with this set up minimal weighted average F1 score is zero (Table A.20).

In terms of performance the only noteworthy change is that the model can detect Major Claim components, however, recall value is very low - 0.03 and precision is 0.31 (Table 5.24). The model preserves the tendency to be biased towards classifying the majority of tokens as Premise Support components (Table 5.40).

#### Model Comparison

As seen in Table 5.30 all zero-shot models trained for the task of argument component classification fail to detect tokens with B-MajorClaim label, and

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.72 | 0.68 | 0.69 | 2143 |
| B-MajorClaim | 0.22 | 0.11 | 0.14 | 6 |
| I-MajorClaim | 0.37 | 0.18 | 0.23 | 107 |
| B-Claim-For | 0.21 | 0.16 | 0.17 | 24 |
| I-Claim-For | 0.27 | 0.26 | 0.25 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 50 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 5 |
| I-Premise-Attack | 0.03 | 0.02 | 0.02 | 72 |
| B-Premise-Support | 0.39 | 0.43 | 0.4 | 73 |
| I-Premise-Support | 0.45 | 0.55 | 0.48 | 1159 |
| Accuracy | - | - | 0.55 | 3969 |
| Macro average | 0.24 | 0.22 | 0.22 | 3969 |
| Weighted average | 0.57 | 0.55 | 0.55 | 3969 |

Table 5.17: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification.

they perform poorly when it comes to the classification of components Against and Attack stance.

Adding an extra feature (the relative distance of a token from the beginning of a given text) further reduces F1 score for I-MajorClaim label. The mBERT based model with the extra feature detected some Claim components, but the F1 score is still pretty low: 0.1 for B-Claim-Against and 0.03 for I-Claim-Against.

Overall, the mBERT based model with the additional feature showed the best result with weighted average F1 score of 0.38.

### 5.3.3 Few-Shot Language Transfer

In the section below we describe the models trained on the combination of persuasive essays dataset in English language and film reviews dataset in Norwegian, that were evaluated on the film reviews dataset in Norwegian language for the task of argument component classification.

#### Multilingual BERT - Model 1

As shown in Figure 5.41 the model is rather unstable during the learning process. The loss value decrease from epoch 1 through epoch 5, however, we can observe a number of spikes afterwords. Maximal weighted average F1 score value is 0.3070, average is 0.1971, and standard deviation of the weighted average F1 score across training epochs is 0.0714 (Table A.21).

As seen in Table 5.26 and confusion matrix in Figure 5.42 the model is performing poorly at the classification of components with Against and Attack stance. Many errors happen because tokens with true O-label get


Figure 5.28: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification. Confusion matrix.

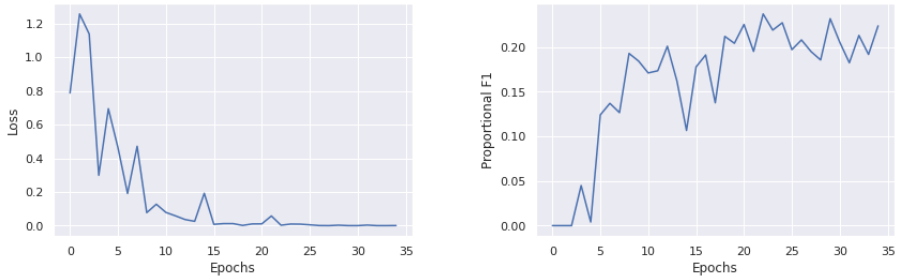
classified as I-Promise-Support and vice versa. Tokens with true Against and Attack stance are most often labelled by the model with O-label.

### Multilingual BERT - Model 2

Adding extra feature (the relative distance of a token from the beginning of a given text) makes the model more stable as seen in Figure 5.43. Except for a sharp decrease observed during epoch 25, F1 score tends to grow steadily across the training epochs.

Minimal value of weighted average F1 score is zero. Maximal weighted average F1 score achieved is 0.3318, the average of the weighted average F1 score across 35 learning epochs is 0.2103 with standard deviation of 0.0812 (Table A.22).

With extra feature (the relative distance of a token from the beginning




(a) Loss. (b) F1 score.

Figure 5.29: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification. Weighted F1 score and loss during model training.

of a given text) included, Recall value for I-MajorComponent raised from 0.22 to 0.29, precision raised from 0.31 to 0.41. Precision and recall values for B-MajorClaim label also raised by 0.05. Although, we observe low performance when it comes to the classification of the components with Against and Attack stance (Table 5.27).

As seen in the confusion matrix in Figure 5.44 many errors happen because tokens with true O-label get classified as I-Promise-Support and vice versa. Tokens with true Against and Attack stance are most often labelled by the model with O-label.

**XLM-RoBERTa - Model 1**

During training of the XLM-RoBERTa based model we observed that the loss value fluctuated during training training epoch 0 through 15 and then settled down while F1 score reached nearly maximum value at epoch number 5 and afterwords showed changes within the range of 0.20-0.25 (Figure 5.45).

Minimal weighted average F1 score that we observed is zero. Maximum weighted average F1 score value is 0.3070, while its average value across 35 training epochs is 0.2168 (Table A.23).

As seen in Table 5.28, the model has zero precision and recall for B-Claim-Against, I-Claim-Against, and B-Premise-Attack labels. Precision and recall for I-Premise-Attack label is nearly zero. As compared with the counterpart mBERT based model, this model performs decently at the classification of Major Claim component.

As seen in confusion matrix in Figure 5.46 many errors happen because tokens with true O-label get classified as I-Promise-Support and vice versa. Tokens with true Against and Attack stance are most often labelled by the model with O-label. Tokens with true Against and Attack stance are most often wrongly marked with either O-label or I-Promise-Support.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.68 | 0.73 | 0.7 | 2143 |
| B-MajorClaim | 0.06 | 0.04 | 0.05 | 6 |
| I-MajorClaim | 0.44 | 0.18 | 0.24 | 107 |
| B-Claim-For | 0.15 | 0.14 | 0.14 | 24 |
| I-Claim-For | 0.18 | 0.24 | 0.2 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.05 | 0.02 | 0.02 | 50 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 5 |
| I-Premise-Attack | 0.0 | 0.0 | 0.0 | 72 |
| B-Premise-Support | 0.28 | 0.26 | 0.25 | 73 |
| I-Premise-Support | 0.43 | 0.44 | 0.43 | 1159 |
| Accuracy | - | - | 0.55 | 3969 |
| Macro average | 0.21 | 0.19 | 0.19 | 3969 |
| Weighted average | 0.53 | 0.55 | 0.53 | 3969 |

Table 5.18: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification.

#### XLM-RoBERTa - Model 2

This model reached near maximum results at a later training stage, namely, around epoch number 10 (Figure 5.47) as compared to the model without extra feature. Minimal weighted average F1 score that we observed during training of this model is zero, similar to the previous model. Maximal weighted average F1 score value that we managed to achieve across the 35 training epochs equals to 0.23 and it is substantially lower (almost by 0.07) as compared with the counter model without an extra feature (the relative distance of a token from the beginning of a respective text). The weighted average F1 score averaged to 0.11 across the 35 training epochs (Table A.24).

Adding an extra feature had an adverse effect on model performance when it comes to Major Claim classification (Table 5.29). Weighted average F1 score for B-MajorClaim reduced from 0.3 to 0.15, for I-MajorClaim from 0.33 to 0.27. Simultaneously, the model has become more biased towards O-label and I-Premise-Support label. They both got higher recall values and lower precision.

As seen in Figure 5.48 true O-labelled tokens and I-Premise-Support are rather often wrongly classified as I-Claim-For that is something we have not observed in previous models.

#### Model Comparison

As seen in Table 5.30 all the few-shot language transfer models trained for the task of argument component classification suffer from the low performance when in comes to the classification of components with Attack and Against stance. Although, mBERT based models have non zero F1

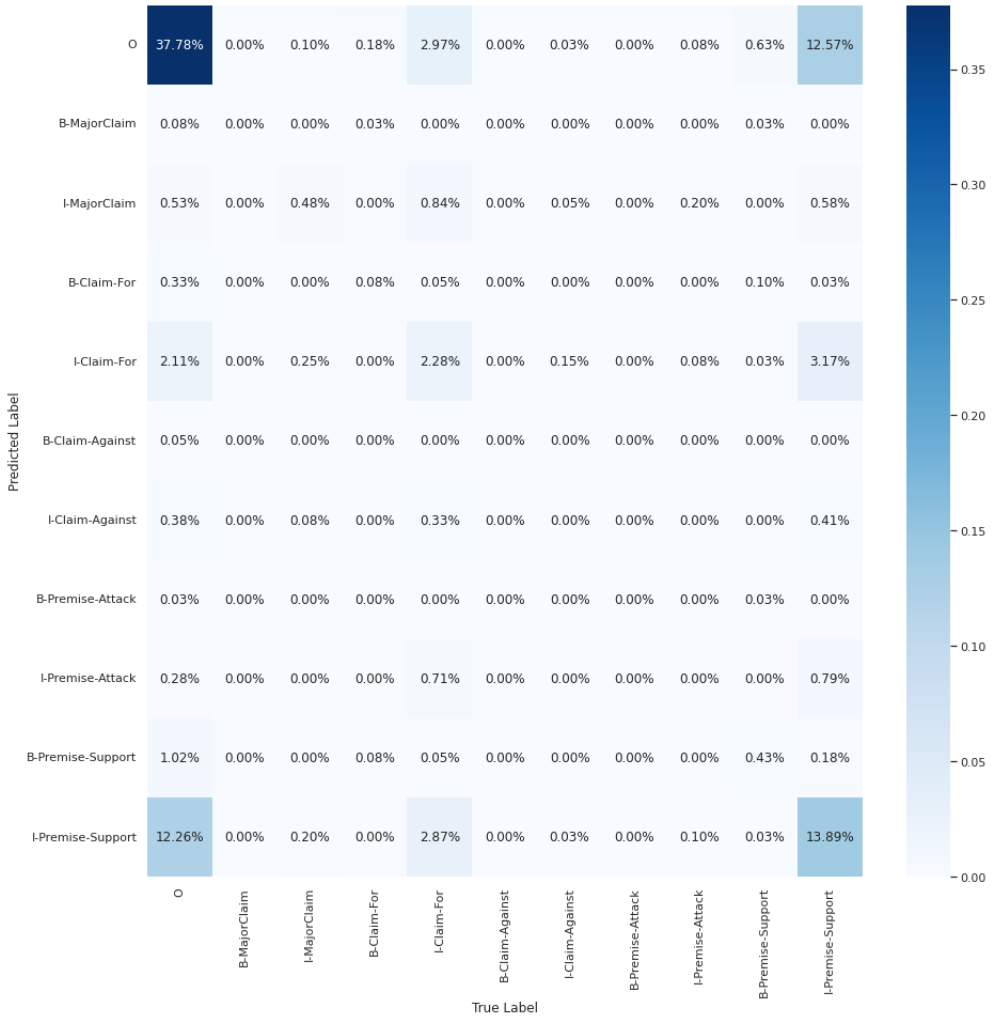
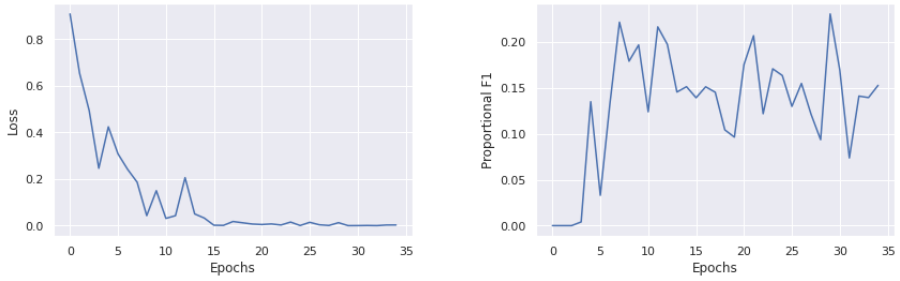



Figure 5.30: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.31: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.73 | 0.69 | 0.7 | 2143 |
| B-MajorClaim | 0.34 | 0.23 | 0.25 | 6 |
| I-MajorClaim | 0.5 | 0.25 | 0.32 | 107 |
| B-Claim-For | 0.2 | 0.2 | 0.2 | 24 |
| I-Claim-For | 0.27 | 0.27 | 0.25 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.06 | 0.04 | 0.04 | 50 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 5 |
| I-Premise-Attack | 0.0 | 0.0 | 0.0 | 72 |
| B-Premise-Support | 0.33 | 0.44 | 0.37 | 73 |
| I-Premise-Support | 0.47 | 0.56 | 0.51 | 1159 |
| Accuracy | - | - | 0.57 | 3969 |
| Macro average | 0.26 | 0.24 | 0.24 | 3969 |
| Weighted average | 0.58 | 0.57 | 0.56 | 3969 |

Table 5.19: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification.

score for B-Premise-Attack and I-Premise-Attack, the values are quite low, from 0.04 to 0.07.

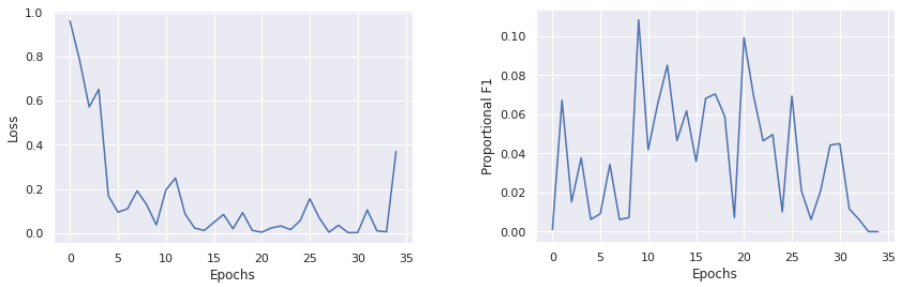
Interestingly, mBERT based model improves weighted average F1 score for Major claim component when we add extra feature (the relative distance of a token from the beginning of a given text) to it, while we observe the reverse effect for the XLM-RoBERTa based models.

Overall, we achieved the best result with XLM-RoBERTa based model without the extra feature included, with a weighted average F1 score achieved is 0.57.

### 5.3.4 Influence of the Proportion of Low-Resource Language Training Material in Training Data on Few-Shot Language Transfer

The aim of this set of experiments is to observe how the proportion of low-resource language training material in the training dataset influences the performance of few-shot language transfer. We ran in total 32 experiments. There are two series of experiments consisting of 16 experiments respectively for the task of argument component detection and 16 for the argument component classification task. In each of the series we are training an XLM-RoBERTa model without any additional features included. There are 16 training sets used in total. The first training set consists of only the persuasive dataset, i.e it includes only English texts. Each subsequent training set includes an additional text from the Norwegian film reviews dataset. Thus, a training set used for the second experiment in the series includes one text in Norwegian, while the last sixteenth training set includes 15 texts in Norwegian.


Figure 5.32: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.33: Model: mBERT, zero-shot transfer, with no extra features, argument component classification. Weighted F1 score and loss during model training.

| | F1 - Score | | | |
|---|---|---|---|---|
| | mBERT 1 | mBERT 2 | XLM-RoBERTa 1 | XLM-RoBERTa 2 |
| O | 0.69 | 0.69 | 0.7 | 0.7 |
| B-MajorClaim | 0.13 | 0.14 | 0.05 | 0.25 |
| I-MajorClaim | 0.22 | 0.23 | 0.24 | 0.32 |
| B-Claim-For | 0.18 | 0.17 | 0.14 | 0.2 |
| I-Claim-For | 0.27 | 0.25 | 0.2 | 0.25 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 0.0 |
| I-Claim-Against | 0.0 | 0.0 | 0.02 | 0.04 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 0.0 |
| I-Premise-Attack | 0.0 | 0.02 | 0.0 | 0.0 |
| B-Premise-Support | 0.33 | 0.4 | 0.25 | 0.37 |
| I-Premise-Support | 0.49 | 0.48 | 0.43 | 0.51 |
| Accuracy | 0.55 | 0.55 | 0.55 | 0.57 |
| Macro average | 0.21 | 0.22 | 0.19 | 0.24 |
| Weighted average | 0.55 | 0.55 | 0.53 | **0.56** |

Table 5.20: F1 score. Comparison of models trained and evaluated on film reviews dataset in Norwegian, argument component classification.

For the purpose of these experiments we were logging just the maximal weighted average F1 score for all labels and summarized the results in two line charts, one for the task of argument component detection, another - for the task of argument component classification.

**Argument Component Detection**

During running the first experiment (we can consider this one as the zero-shot language transfer, since the training set in this case included only text in English from the persuasive essays dataset) the model reached weighted average F1 score of 0.49 as is in line with the results we achieved when experimenting with the zero-short transfer models in the sections above.

It was enough to include just one additional text in the Norwegian language in the training set in order to see a substantial improvement in the performance of the model. The second model in the series reached weighted average F1 score amounting to 0.58, which 0.09 more as compared with purely zero-shot language transfer model. The models trained on the training sets including 2, 3, and 4 text in Norwegian reached weighted average F1 scores of 0.58, 0.59, 0.63 respectively. We observed that with each additional text the performance increases by 0.01-0.04. After we added a fifth Norwegian text to the training set we observed that the models began to yield better results, however, the improvement value gradually decreased.

For the overview of the models' performance with varying amount of training data in Norwegian language see Figure 5.50.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.63 | 0.28 | 0.38 | 8575 |
| B-MajorClaim | 0.0 | 0.0 | 0.0 | 27 |
| I-MajorClaim | 0.11 | 0.02 | 0.04 | 431 |
| B-Claim-For | 0.12 | 0.05 | 0.07 | 98 |
| I-Claim-For | 0.17 | 0.1 | 0.13 | 1288 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 18 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 200 |
| B-Premise-Attack | 0.03 | 0.05 | 0.04 | 21 |
| I-Premise-Attack | 0.05 | 0.14 | 0.07 | 289 |
| B-Premise-Support | 0.3 | 0.55 | 0.39 | 292 |
| I-Premise-Support | 0.35 | 0.74 | 0.48 | 4637 |
| Accuracy | - | - | 0.39 | 15876 |
| Macro average | 0.16 | 0.17 | 0.15 | 15876 |
| Weighted average | 0.47 | 0.39 | 0.37 | 15876 |

Table 5.21: Model: mBERT, zero-shot transfer, with no extra features, argument component classification.

#### Argument Component Classification

The experiments for the task of argument component classification were performed in the similar fashion as in the case of argument component detection. The fist model was trained on a training set with no linguistic material in Norwegian language included. We observed the weighted average F1 score of 0.086. The addition of one text in the Norwegian language allowed us to reach F1 score of 0.141, which more than 0.04 improvement over the base case. Weighted average F1 score changes less evenly as compared with the experiments we ran for the task of argument component detection. Our intuition is that for more complicated natural language processing tasks it is required to mix in more material in a low-resource language to a training set in order to observe tangible improvements as compared with less complicated natural language processing tasks.

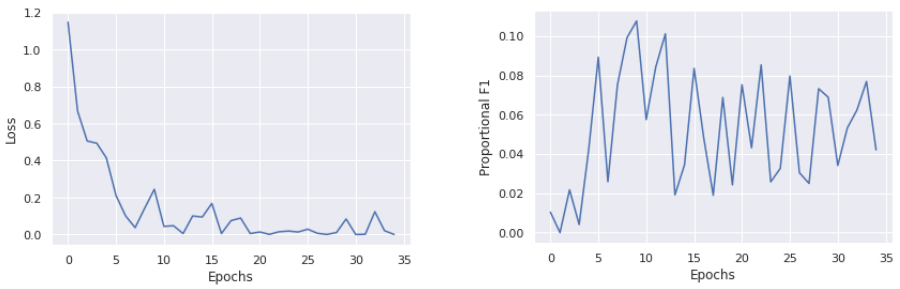
For the overview of the models' performance with varying amount of training data in Norwegian language see Figure 5.50.

### 5.3.5 Summary of Findings

In Table 5.31 we provide a summary of models trained for solving the task of argument component detection. And in Table 5.32 we provide a summary of experiments carried out with the models trained for the task of argument component identification.

Although, it is not correct to directly compare the results from zero-shot language transfer models with the few-shot language models, and models trained and evaluated on Norwegian dataset due to the differences in experimental set up (for the full description see Section 4.2), we feel that the output of zero-shot transfer experiments, provided we carried them


Figure 5.34: Model: mBERT, zero-shot transfer, with no extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.35: Model: mBERT, zero-shot transfer, with extra features, argument component classification. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.62 | 0.3 | 0.41 | 8575 |
| B-MajorClaim | 0.0 | 0.0 | 0.0 | 27 |
| I-MajorClaim | 0.0 | 0.0 | 0.0 | 431 |
| B-Claim-For | 0.11 | 0.02 | 0.03 | 98 |
| I-Claim-For | 0.29 | 0.1 | 0.14 | 1288 |
| B-Claim-Against | 0.5 | 0.06 | 0.1 | 18 |
| I-Claim-Against | 0.04 | 0.02 | 0.03 | 200 |
| B-Premise-Attack | 0.05 | 0.14 | 0.07 | 21 |
| I-Premise-Attack | 0.04 | 0.19 | 0.06 | 289 |
| B-Premise-Support | 0.25 | 0.25 | 0.25 | 292 |
| I-Premise-Support | 0.35 | 0.71 | 0.47 | 4637 |
| Accuracy | - | - | 0.39 | 15876 |
| Macro average | 0.20 | 0.16 | 0.14 | 15876 |
| Weighted average | 0.47 | 0.39 | 0.38 | 15876 |

Table 5.22: Model: mBERT, zero-shot transfer, with extra features, argument component classification.

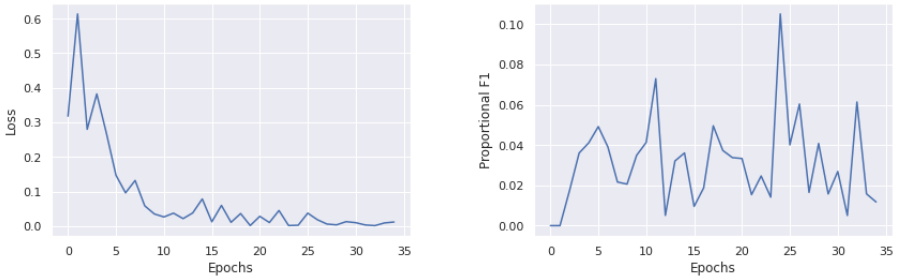
out following n-fold cross validation methodology, would have produced comparable results. In general, they prove that zero-shot language transfer is a viable option for English and Norwegian language pairs for the task of argument component detection, although the performance of zero-short models fall well behind few-shot language models and models trained and evaluated in the same language. In our experiments, zero-shot transfer models reached weighted F1 scores about 0.20 less than the respective few-shot models and the models trained and evaluated on Norwegian language both for the task of argument component detection and argument component classification.

Out of the 12 models trained for the task of argument component detection XLM-RoBERTa based zero-shot language transfer model showed the best result. It reached weighted averaged F1 score of 0.75, which is 0.02 improvement over the best result among the models trained and evaluated on the Norwegian dataset. If we draw a comparison line between mBERT and XLM-RoBERTa based models, we can notice that except for the zero-shot transfer models XLM-RoBERTa based models slightly outperform the mBERT based ones. It is also worth to notice that adding an extra feature (relative distance of a token from document start, for the complete description see Subsection 4.4) negatively effects the performance of all models.

XLM-RoBERTa few-shot transfer model with no extra features achieved the best result for the task of argument component classification. Its weighted average F1 score is 0.57. The runner-up is XLM-RoBERTa model trained on the film reviews dataset in Norwegian with extra feature (the relative distance of a token from the beginning of a given text) included. This model scored 0.56.

XLM-RoBERTa based zero-shot transfer models scored by 0.05 less in


Figure 5.36: Model: mBERT, zero-shot transfer, with extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.37: Model: XLM-RoBERTa, zero-shot transfer, with no extra features, argument component classification. Weighted F1 score and loss during model training.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.63 | 0.2 | 0.3 | 8575 |
| B-MajorClaim | 0.0 | 0.0 | 0.0 | 27 |
| I-MajorClaim | 0.0 | 0.0 | 0.0 | 431 |
| B-Claim-For | 0.14 | 0.06 | 0.08 | 98 |
| I-Claim-For | 0.17 | 0.11 | 0.13 | 1288 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 18 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 200 |
| B-Premise-Attack | 0.06 | 0.05 | 0.05 | 21 |
| I-Premise-Attack | 0.06 | 0.09 | 0.07 | 289 |
| B-Premise-Support | 0.29 | 0.63 | 0.39 | 292 |
| I-Premise-Support | 0.34 | 0.81 | 0.48 | 4637 |
| Accuracy | - | - | 0.37 | 15876 |
| Macro average | 0.15 | 0.18 | 0.14 | 15876 |
| Weighted average | 0.46 | 0.37 | 0.32 | 15876 |

Table 5.23: Model: XLM-RoBERTa, zero-shot transfer, with no extra features, argument component classification.

F1 score compared to the mBERT based counterparts. The difference in performance of mBERT and XLM-RoBERTa based few-shot models and models trained on the film reviews dataset in Norwegian is within 0.01-0.2 (see Table 5.32).

Adding an extra feature (the relative distance of a token from the beginning of a given text) has a positive effect on the overall performance of XLM-RoBERTa based zero-shot language transfer models and models trained and evaluated on the Norwegian dataset. We discussed this phenomena above in the previous sections.

It is also worth to point out that all the models that we trained for the task of argument component classification suffer from the same problem. They show low performance or totally fail when it comes to the identification of the components with Against and Attack stance. This might be partially attributed to the fact that the components of these types are less numerous in our datasets. However, Major Claim component has the lowest frequency in the datasets and the models we trained still identify it. Furthermore, we observed that adding an extra feature (the relative distance of a token from the beginning of a text) that was specifically designed to descriminate this type of an argument component was beneficial. Thus, we may assume that the introduction of extra hand-crafted features might also contribute to better performance of the transformer based models when it comes to classification of the components with Against or Attack stance.

Our experiments with a few-language models with a varying amount of low-resource language material (Norwegian in our case) in training sets mainly consisting of high-resource language material (English) showed that adding even one text in low-resource language to a training set yields significant performance gain as compared to the results obtained

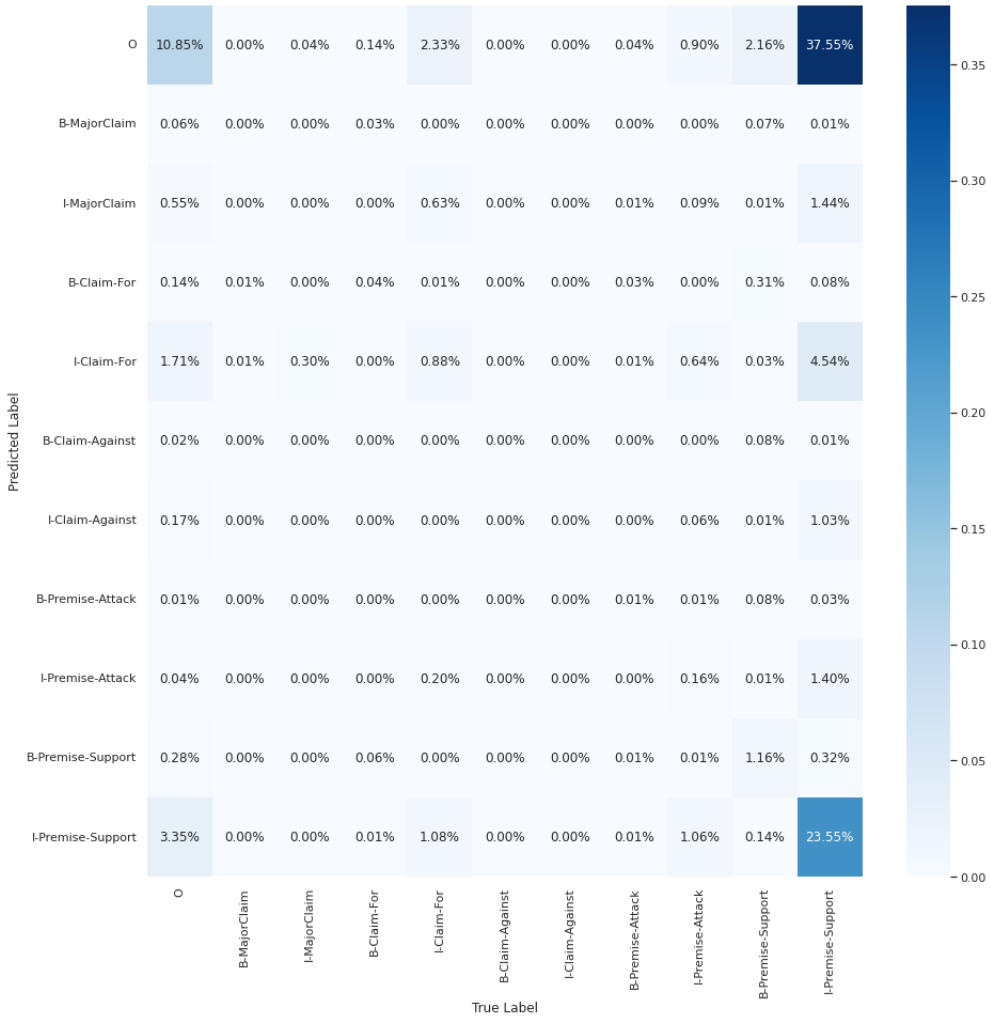
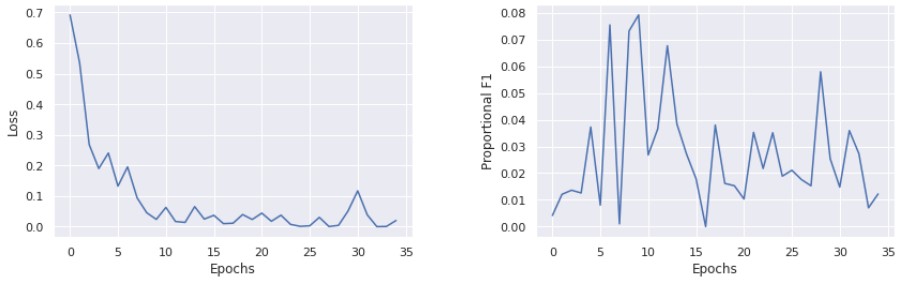



Figure 5.38: Model: XLM-RoBERTa, zero-shot transfer, with no extra features, argument component classification. Confusion matrix.

in a purely low-resource language set up. For the task of argument component detection adding one text in Norwegian to the training dataset improved average weighted F1 score by 0.09. The improvement for the task of argument component classification was 0.04. Further growth of the proportion of a low-resource language material in a training set yielded diminishing results.




(a) Loss.

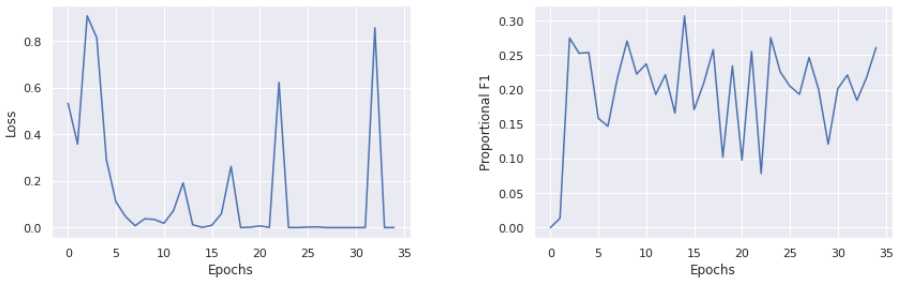
(b) F1 score.

Figure 5.39: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component classification. Weighted F1 score and loss during model training.

| | **Precision** | **Recall** | **F1-Score** | **Support** |
|---|---|---|---|---|
| O | 0.63 | 0.22 | 0.33 | 8575 |
| B-MajorClaim | 0.0 | 0.0 | 0.0 | 27 |
| I-MajorClaim | 0.31 | 0.03 | 0.06 | 431 |
| B-Claim-For | 0.03 | 0.01 | 0.02 | 98 |
| I-Claim-For | 0.12 | 0.07 | 0.09 | 1288 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 18 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 200 |
| B-Premise-Attack | 0.04 | 0.05 | 0.04 | 21 |
| I-Premise-Attack | 0.04 | 0.1 | 0.05 | 289 |
| B-Premise-Support | 0.29 | 0.57 | 0.39 | 292 |
| I-Premise-Support | 0.35 | 0.78 | 0.48 | 4637 |
| Accuracy | - | - | 0.37 | 15876 |
| Macro average | 0.16 | 0.17 | 0.13 | 15876 |
| Weighted average | 0.46 | 0.37 | 0.33 | 15876 |

Table 5.24: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component classification.


Figure 5.40: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.41: Model: mBERT, few-shot transfer, with no extra features, argument component classification. Weighted F1 score and loss during model training.

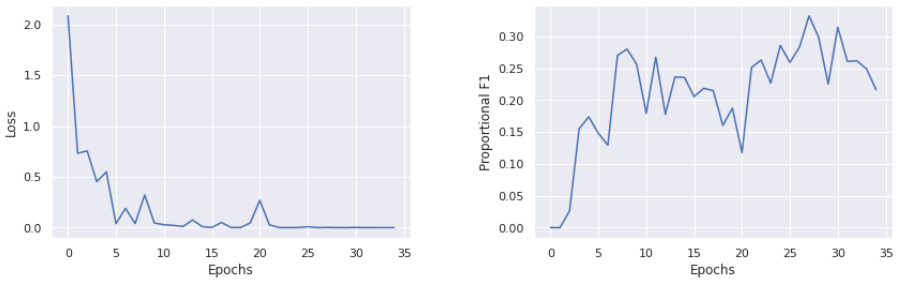
| | F1 - Score | | | |
|---|---|---|---|---|
| | mBERT 1 | mBERT 2 | XLM-RoBERTa 1 | XLM-RoBERTa 2 |
| O | 0.38 | 0.41 | 0.3 | 0.33 |
| B-MajorClaim | 0.0 | 0.0 | 0.0 | 0.0 |
| I-MajorClaim | 0.04 | 0.0 | 0.0 | 0.06 |
| B-Claim-For | 0.07 | 0.03 | 0.08 | 0.02 |
| I-Claim-For | 0.13 | 0.14 | 0.13 | 0.09 |
| B-Claim-Against | 0.0 | 0.1 | 0.0 | 0.0 |
| I-Claim-Against | 0.0 | 0.03 | 0.0 | 0.0 |
| B-Premise-Attack | 0.04 | 0.07 | 0.05 | 0.04 |
| I-Premise-Attack | 0.07 | 0.06 | 0.07 | 0.05 |
| B-Premise-Support | 0.39 | 0.25 | 0.39 | 0.39 |
| I-Premise-Support | 0.48 | 0.47 | 0.48 | 0.48 |
| Accuracy | 0.39 | 0.39 | 0.37 | 0.37 |
| Macro average | 0.15 | 0.14 | 0.14 | 0.13 |
| Weighted average | 0.37 | **0.38** | 0.32 | 0.33 |

Table 5.25: F1 score. Comparison of models trained on persuasive essays dataset in English and evaluated on film reviews dataset in Norwegian, argument component classification.

| | **Precision** | **Recall** | **F1-Score** | **Support** |
|---|---|---|---|---|
| O | 0.68 | 0.7 | 0.68 | 2143 |
| B-MajorClaim | 0.16 | 0.18 | 0.17 | 5 |
| I-MajorClaim | 0.31 | 0.22 | 0.21 | 107 |
| B-Claim-For | 0.26 | 0.19 | 0.22 | 24 |
| I-Claim-For | 0.34 | 0.27 | 0.29 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 50 |
| B-Premise-Attack | 0.03 | 0.12 | 0.05 | 5 |
| I-Premise-Attack | 0.09 | 0.08 | 0.07 | 72 |
| B-Premise-Support | 0.37 | 0.34 | 0.34 | 73 |
| I-Premise-Support | 0.46 | 0.51 | 0.47 | 1159 |
| Accuracy | - | - | 0.56 | 3969 |
| Macro average | 0.25 | 0.24 | 0.23 | 3969 |
| Weighted average | 0.56 | 0.56 | 0.55 | 3969 |

Table 5.26: Model: mBERT, few-shot transfer, with no extra features, argument component classification.


Figure 5.42: Model: mBERT, few-shot transfer, with no extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.43: Model: mBERT, few-shot transfer, with extra features, argument component classification. Weighted F1 score and loss during model training.

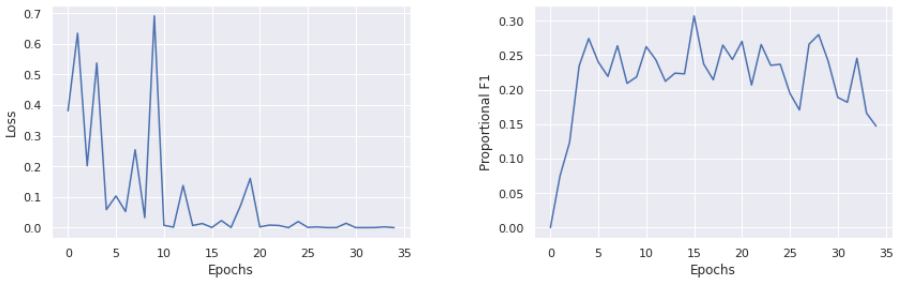
| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.69 | 0.75 | 0.72 | 2143 |
| B-MajorClaim | 0.21 | 0.23 | 0.22 | 6 |
| I-MajorClaim | 0.41 | 0.29 | 0.33 | 107 |
| B-Claim-For | 0.14 | 0.14 | 0.12 | 24 |
| I-Claim-For | 0.26 | 0.26 | 0.23 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.03 | 0.04 | 0.03 | 50 |
| B-Premise-Attack | 0.08 | 0.03 | 0.04 | 5 |
| I-Premise-Attack | 0.13 | 0.03 | 0.05 | 72 |
| B-Premise-Support | 0.34 | 0.31 | 0.33 | 73 |
| I-Premise-Support | 0.45 | 0.44 | 0.44 | 1159 |
| Accuracy | - | - | 0.57 | 3969 |
| Macro average | 0.25 | 0.23 | 0.23 | 3969 |
| Weighted average | 0.55 | 0.57 | 0.55 | 3969 |

Table 5.27: Model: mBERT, few-shot transfer, with extra features, argument component classification.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| O | 0.72 | 0.7 | 0.71 | 2143 |
| B-MajorClaim | 0.28 | 0.35 | 0.3 | 6 |
| I-MajorClaim | 0.38 | 0.35 | 0.33 | 107 |
| B-Claim-For | 0.17 | 0.2 | 0.18 | 24 |
| I-Claim-For | 0.26 | 0.27 | 0.25 | 322 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 50 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 5 |
| I-Premise-Attack | 0.01 | 0.02 | 0.01 | 72 |
| B-Premise-Support | 0.39 | 0.37 | 0.37 | 73 |
| I-Premise-Support | 0.5 | 0.54 | 0.52 | 1159 |
| Accuracy | - | - | 0.58 | 3969 |
| Macro average | 0.25 | 0.25 | 0.24 | 3969 |
| Weighted average | 0.58 | 0.58 | 0.57 | 3969 |

Table 5.28: Model: XLM-RoBERTa, few-shot transfer, with no extra features, argument component classification.


Figure 5.44: Model: mBERT, few-shot transfer, with extra features, argument component classification. Confusion matrix.

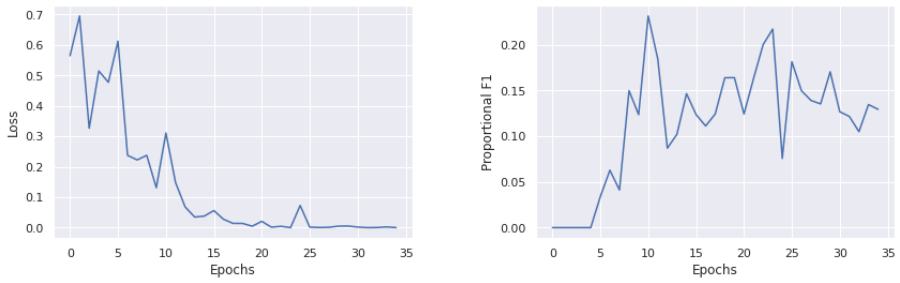



(a) Loss.

(b) F1 score.

Figure 5.45: Model: XLM-RoBERTa, few-shot transfer, with no extra features, argument component classification. Weighted F1 score and loss during model training.


Figure 5.46: Model: XLM-RoBERTa, few-shot transfer, with no extra features, argument component classification. Confusion matrix.




(a) Loss.

(b) F1 score.

Figure 5.47: Model: XLM-RoBERTa, few-shot transfer, with extra features, argument component classification. Weighted F1 score and loss during model training.

| | **Precision** | **Recall** | **F1-Score** | **Support** |
|---|---|---|---|---|
| O | 0.74 | 0.66 | 0.69 | 2143.75 |
| B-MajorClaim | 0.29 | 0.1 | 0.15 | 6.75 |
| I-MajorClaim | 0.55 | 0.21 | 0.27 | 107.75 |
| B-Claim-For | 0.16 | 0.17 | 0.14 | 24.5 |
| I-Claim-For | 0.28 | 0.36 | 0.28 | 322.0 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 4.5 |
| I-Claim-Against | 0.0 | 0.0 | 0.0 | 50.0 |
| B-Premise-Attack | 0.0 | 0.0 | 0.0 | 5.25 |
| I-Premise-Attack | 0.04 | 0.01 | 0.01 | 72.25 |
| B-Premise-Support | 0.38 | 0.3 | 0.28 | 73.0 |
| I-Premise-Support | 0.45 | 0.54 | 0.49 | 1159.25 |
| Accuracy | - | - | 0.55 | 3969 |
| Macro average | 0.26 | 0.21 | 0.21 | 3969 |
| Weighted average | 0.58 | 0.55 | 0.55 | 3969 |

Table 5.29: Model: XLM-RoBERTa, few-shot transfer, with extra features, argument component classification.

| | F1 - Score | | | |
|---|---|---|---|---|
| | mBERT 1 | mBERT 2 | XLM-RoBERTa 1 | XLM-RoBERTa 2 |
| O | 0.68 | 0.72 | 0.71 | 0.69 |
| B-MajorClaim | 0.17 | 0.22 | 0.3 | 0.15 |
| I-MajorClaim | 0.21 | 0.33 | 0.33 | 0.27 |
| B-Claim-For | 0.22 | 0.12 | 0.18 | 0.14 |
| I-Claim-For | 0.29 | 0.23 | 0.25 | 0.28 |
| B-Claim-Against | 0.0 | 0.0 | 0.0 | 0.0 |
| I-Claim-Against | 0.0 | 0.03 | 0.0 | 0.0 |
| B-Premise-Attack | 0.05 | 0.04 | 0.0 | 0.0 |
| I-Premise-Attack | 0.07 | 0.05 | 0.01 | 0.01 |
| B-Premise-Support | 0.34 | 0.33 | 0.37 | 0.28 |
| I-Premise-Support | 0.47 | 0.44 | 0.52 | 0.49 |
| Accuracy | 0.56 | 0.57 | 0.58 | 0.55 |
| Macro average | 0.23 | 0.23 | 0.24 | 0.21 |
| Weighted average | 0.55 | 0.55 | **0.57** | 0.55 |

Table 5.30: F1 score. Comparison of models trained on the mix of persuasive essays dataset in English and film reviews dataset in Norwegian and evaluated on film reviews dataset in Norwegian, argument component classification.

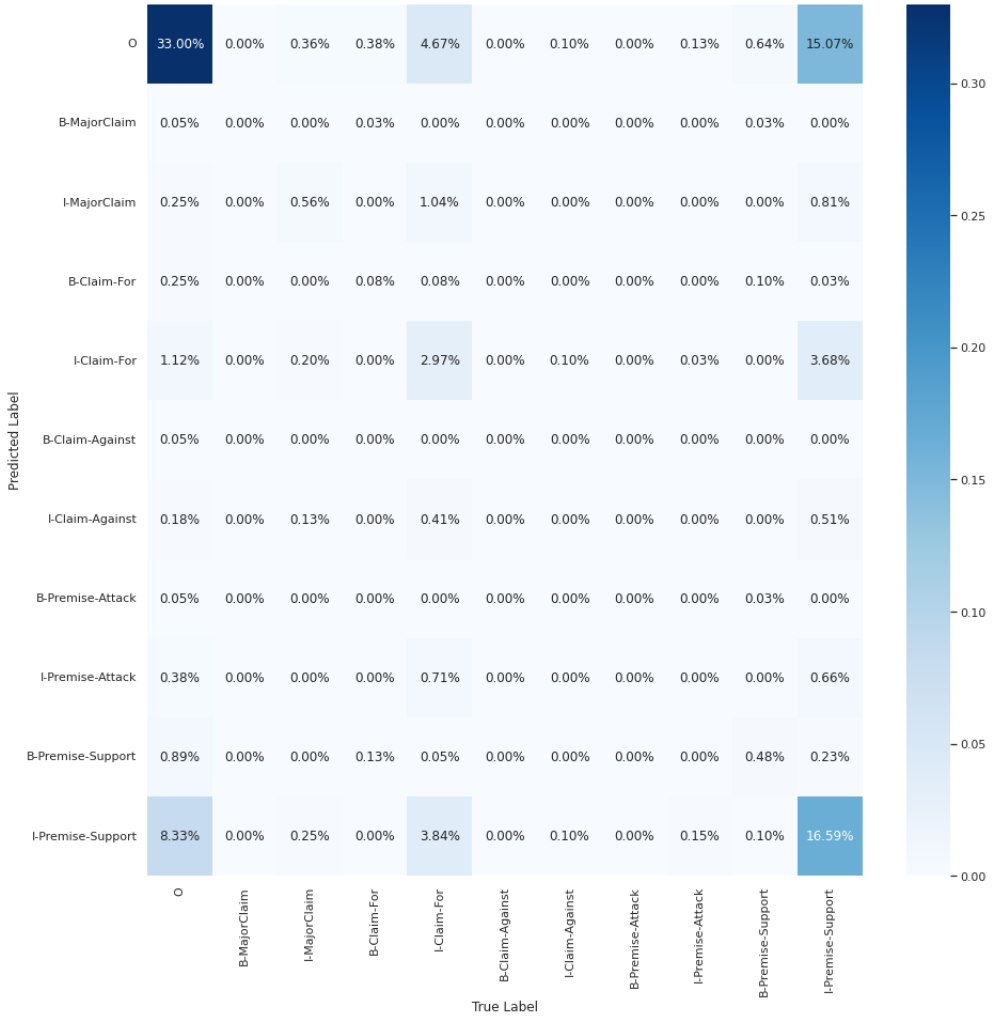



Figure 5.48: Model: XLM-RoBERTa, few-shot transfer, with extra features, argument component classification. Confusion matrix.

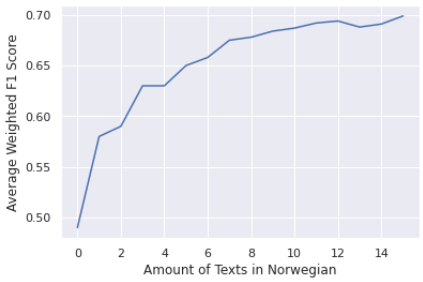



Figure 5.49: Weighted average F1 score. Argument component detection. The influence of proportion of Norwegian texts in training set on the performance of few-shot language transfer.

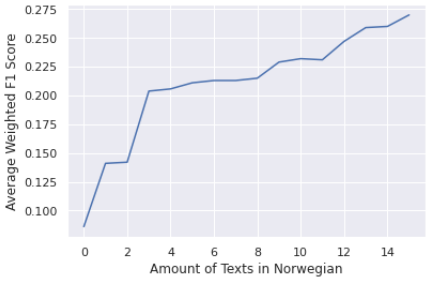



Figure 5.50: Weighted average F1 score. Argument component classification. The influence of proportion of Norwegian texts in training set on the performance of few-shot language transfer.

| | **F1 - Score** | |
|---|---|---|
| | **Macro average** | **Weighted average** |
| **Models trained and evaluated on Norwegian dataset.** | | |
| Multilingual BERT - Model 1 | 0.69 | 0.73 |
| Multilingual BERT - Model 2 | 0.70 | 0.72 |
| XLM-RoBERTa - Model 1 | 0.71 | 0.73 |
| XLM-RoBERTa - Model 2 | 0.70 | 0.73 |
| **Zero-shot language transfer.** | | |
| Multilingual BERT - Model 1 | 0.52 | 0.50 |
| Multilingual BERT - Model 2 | 0.50 | 0.48 |
| XLM-RoBERTa - Model 1 | 0.51 | 0.49 |
| XLM-RoBERTa - Model 2 | 0.48 | 0.45 |
| **Few-shot language transfer.** | | |
| Multilingual BERT - Model 1 | 0.70 | 0.73 |
| Multilingual BERT - Model 2 | 0.70 | 0.73 |
| XLM-RoBERTa - Model 1 | 0.71 | **0.75** |
| XLM-RoBERTa - Model 2 | 0.69 | 0.74 |

Table 5.31: Comparison of all models trained and evaluated based on F1 score. Argument component detection task.

| | F1 - Score | |
|---|---|---|
| | **Macro average** | **Weighted average** |
| **Models trained and evaluated on Norwegian dataset.** | | |
| Multilingual BERT - Model 1 | 0.21 | 0.55 |
| Multilingual BERT - Model 2 | 0.22 | 0.55 |
| XLM-RoBERTa - Model 1 | 0.19 | 0.53 |
| XLM-RoBERTa - Model 2 | 0.24 | 0.56 |
| **Zero-shot language transfer.** | | |
| Multilingual BERT - Model 1 | 0.15 | 0.37 |
| Multilingual BERT - Model 2 | 0.14 | 0.38 |
| XLM-RoBERTa - Model 1 | 0.14 | 0.32 |
| XLM-RoBERTa - Model 2 | 0.13 | 0.33 |
| **Few-shot language transfer.** | | |
| Multilingual BERT - Model 1 | 0.23 | 0.55 |
| Multilingual BERT - Model 2 | 0.23 | 0.55 |
| XLM-RoBERTa - Model 1 | 0.24 | **0.57** |
| XLM-RoBERTa - Model 2 | 0.21 | 0.55 |

Table 5.32: Comparison of all models trained and evaluated based on F1 score. Argument component classification task.

# Chapter 6

# Conclusion

In this thesis we experimentally evaluated the potential for application of zero-shot language transfer techniques for the task of argument component detection and argument component classification and compared them with the respective few-shot transfer models and models trained on sparse data where the training language matches the target language. The textual material we used was in English and Norwegian. The detailed overview of the results is provided in Chapter 5. In general, we can conclude that zero-shot language transfer can potentially be applied for the task of argument component identification, while for the task of argument component classification zero-shot models are barely usable due to low performance.

When it comes to few-shot language transfer models we managed to achieve 0.02-0.03 improvement as compared to the models trained and evaluated in the same language. Our results are in line with the the experiments carried out by Lauscher et al. (2020), who showed that few-shot language transfer models yield by 2-4.5 percentage points higher results for more complex language understanding tasks. We can conclude that in a situation where one has at one's disposal larger datasets in a high-resource language and a smaller dataset in a low-resource language it is beneficial to apply few-shot language transfer technique.

We also performed an experimental comparison of multilingual BERT and XLM-RoBERTa based models for the tasks of argument component detection and argument component classification. Out results show that mBERT based models perform on the same level or slightly better than XLM-RoBERTa based models in argument component detection in zero-shot language transfer and for models trained and evaluated in the same language, while XML-RoBERTa based models surpass multilingual BERT based models in few-shot transfer. Similar differences were captured for the task of argument component classification, however, in this case, the differences were more pronounced in zero-shot transfer models: mBERT based models surpassed XLM-RoBERTa based models by 0.03-0.05 in weighted average F1 score. We can conclude that it is more beneficial to apply XLM-RoBERTa based models in combination with few-shot language transfer technique.

We further evaluated the influence of a hand-crafted feature, the relative distance of a token from the beginning of a text, on the performance of the models. The intuition behind this feature is illustrated in Chapter 3, where we show that some argument components appear more often in certain parts of a text. For instance, Major Claim components appear mostly either in the first or last quarter of a text. This feature had no positive effect on the models trained for the task of argument component detection. However, it significantly improved in most cases the capacity of the models to classify Major Claim components. During our experiments, we logged the performance of all models across training epochs. The models with this extra feature included generally reached nearly maximal results at later epochs, as compared to the models without this hand-crafted feature. We can conclude that token distance from the start of a text can be used as a feature for the classification of Major Claim, this holds true for the genres of student argumentative essays and film reviews, since in the texts of other genres the distribution of Major claim components in a text may differ.

Finally, we experimented with a few-language models with a varying amount of low-resource language material (Norwegian in our case) in training sets mainly consisting of high-resource language material (English). We found out that adding even one text in low-resource language to a training set yields significant performance gain as compared to the results obtained in a purely low-resource language set up. Further growth of the proportion of a low-resource language material in a training set yielded diminishing results. For the task of argument mining detection the improvements were more prominent than for the task of argument mining classification.

## 6.1 Future Work

Our main set of few-shot language transfer models was trained with a fixed amount of data in Norwegian combined with the training data in English. We performed additional experiments with a growing amount of linguistic material in Norwegian included in the training set in order to evaluate how this would influence the performance of the models, however it is necessary to refine experimental set up for the latter in order to get more descriptive results and work out measurable recommendations for the implementation of few-shot transfer models.

While analyzing the results of the experiments we noticed that all the models trained for the task of argument component classification poorly classify components with stances Attack and Against. It might be beneficial to perform additional comparative analysis of such components with their For and Support counterparts and try to identify additional features that could help discriminate better between the components with different stance.

It might be furthermore beneficial to run experiments with more heuristics and additional features. We mentioned some of them in Chapter

2.

We were using the same set of hyperparameters in all the experiments. It might be beneficial to perform more thorough hyperparameters search, separately for multilingual BERT and XLM-RoBERTa based models, for models with and without extra features and for zero-shot transfer models, this could potentially yield better results.

# Appendix A

# Appendix

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.5787 |
| Maximal Weighted F1 Score | 0.6998 |
| Average Weighted F1 Score | 0.6723 |
| Standard Deviation Weighted F1 Score | 0.0270 |
| Best Epoch (Weighted F1 Score) | 23 |

Table A.1: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component detection. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.4810 |
| Maximal Weighted F1 Score | 0.7078 |
| Average Weighted F1 Score | 0.6758 |
| Standard Deviation Weighted F1 Score | 0.0414 |
| Best Epoch (Weighted F1 Score) | 34 |

Table A.2: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component detection. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.6049 |
| Maximal Weighted F1 Score | 0.6986 |
| Average Weighted F1 Score | 0.6679 |
| Standard Deviation Weighted F1 Score | 0.0201 |
| Best Epoch (Weighted F1 Score) | 27 |

Table A.3: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component detection. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.01561 |
| Maximal Weighted F1 Score | 0.6692 |
| Average Weighted F1 Score | 0.5939 |
| Standard Deviation Weighted F1 Score | 0.1272 |
| Best Epoch (Weighted F1 Score) | 34 |

Table A.4: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component detection. Summary of model performance over training epochs.

| | |
|---|---|
| Minimal Weighted F1 Score | 0.6177 |
| Maximal Weighted F1 Score | 0.6714 |
| Average Weighted F1 Score | 0.6518 |
| Standard Deviation Weighted F1 Score | 0.0113 |
| Best Epoch (Weighted F1 Score) | 34 |

Table A.5: Model: mBERT, zero-shot transfer, no extra features, argument component detection. Summary of model performance over training epochs.

| | |
|---|---|
| Minimal Weighted F1 Score | 0.6168 |
| Maximal Weighted F1 Score | 0.6671 |
| Average Weighted F1 Score | 0.6472 |
| Standard Deviation Weighted F1 Score | 0.0115 |
| Best Epoch (Weighted F1 Score) | 12 |

Table A.6: Model: mBERT, zero-shot transfer, with extra features, argument component detection. Summary of model performance over training epochs.

| | |
|---|---|
| Minimal Weighted F1 Score | 0.6076 |
| Maximal Weighted F1 Score | 0.6577 |
| Average Weighted F1 Score | 0.6461 |
| Standard Deviation Weighted F1 Score | 0.0010 |
| Best Epoch (Weighted F1 Score) | 21 |

Table A.7: Model: XLM-RoBERTa, zero-shot transfer, no extra features, argument component detection. Summary of model performance over training epochs.

| **Parameter** | **Value** |
|---|---|
| Minimal Weighted F1 Score | 0.6336 |
| Maximal Weighted F1 Score | 0.6596 |
| Average Weighted F1 Score | 0.6505 |
| Standard Deviation Weighted F1 Score | 0.0066 |
| Best Epoch (Weighted F1 Score) | 4 |

Table A.8: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component detection. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.56 |
| Maximal Weighted F1 Score | 0.70 |
| Average Weighted F1 Score | 0.67 |
| Standard Deviation Weighted F1 Score | 0.03 |
| Best Epoch (Weighted F1 Score) | 11 |

Table A.9: Model: mBERT, few-shot transfer, no extra features, argument component detection. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.58 |
| Maximal Weighted F1 Score | 0.70 |
| Average Weighted F1 Score | 0.66 |
| Standard Deviation Weighted F1 Score | 0.03 |
| Best Epoch (Weighted F1 Score) | 5 |

Table A.10: Model: mBERT, few-shot transfer, no extra features, argument component detection. Summary of model performance over training epochs.

| | |
|---|---|
| Minimal Weighted F1 Score | 0.61 |
| Maximal Weighted F1 Score | 0.70 |
| Average Weighted F1 Score | 0.68 |
| Standard Deviation Weighted F1 Score | 0.02 |
| Best Epoch (Weighted F1 Score) | 27 |

Table A.11: Model: XLM-RoBERTa, few-shot transfer, no extra features, argument component detection. Summary of model performance over training epochs.

| | |
|---|---|
| Minimal Weighted F1 Score | 0.58 |
| Maximal Weighted F1 Score | 0.70 |
| Average Weighted F1 Score | 0.67 |
| Standard Deviation Weighted F1 Score | 0.02 |
| Best Epoch (Weighted F1 Score) | 34 |

Table A.12: Model: XLM-RoBERTa, few-shot transfer, no extra features, argument component detection. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.2969 |
| Average Weighted F1 Score | 0.1886 |
| Standard Deviation Weighted F1 Score | 0.0567 |
| Best Epoch (Weighted F1 Score) | 10 |

Table A.13: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.3139 |
| Average Weighted F1 Score | 0.2187 |
| Standard Deviation Weighted F1 Score | 0.0827 |
| Best Epoch (Weighted F1 Score) | 6 |

Table A.14: Model: mBERT, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.2373 |
| Average Weighted F1 Score | 0.1626 |
| Standard Deviation Weighted F1 Score | 0.0708 |
| Best Epoch (Weighted F1 Score) | 23 |

Table A.15: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with no extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.2308 |
| Average Weighted F1 Score | 0.1320 |
| Standard Deviation Weighted F1 Score | 0.0630 |
| Best Epoch (Weighted F1 Score) | 30 |

Table A.16: Model: XLM-RoBERTa, model trained and evaluated on film reviews dataset in Norwegian, with extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.1083 |
| Average Weighted F1 Score | 0.0381 |
| Standard Deviation Weighted F1 Score | 0.0304 |
| Best Epoch (Weighted F1 Score) | 12 |

Table A.17: Model: mBERT, zero-shot transfer, with no extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Proportional F1 Score | 0.0 |
| Maximal Proportional F1 Score | 0.1079 |
| Average Proportional F1 Score | 0.0522 |
| Standard Deviation Proportional F1 Score | 0.03 |
| Best Epoch (Proportional F1 Score) | 10 |

Table A.18: Model: mBERT, zero-shot transfer, with extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.1051 |
| Average Weighted F1 Score | 0.0309 |
| Standard Deviation Weighted F1 Score | 0.0218 |
| Best Epoch (Weighted F1 Score) | 25 |

Table A.19: Model: XLM-RoBERTa, zero-shot transfer, with no extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.0792 |
| Average Weighted F1 Score | 0.0273 |
| Standard Deviation Weighted F1 Score | 0.0210 |
| Best Epoch (Weighted F1 Score) | 10 |

Table A.20: Model: XLM-RoBERTa, zero-shot transfer, with extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.3070 |
| Average Weighted F1 Score | 0.1971 |
| Standard Deviation Weighted F1 Score | 0.0714 |
| Best Epoch (Weighted F1 Score) | 15 |

Table A.21: Model: mBERT, few-shot transfer, with no extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.3318 |
| Average Weighted F1 Score | 0.2103 |
| Standard Deviation Weighted F1 Score | 0.0812 |
| Best Epoch (Weighted F1 Score) | 28 |

Table A.22: Model: mBERT, few-shot transfer, with extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Weighted F1 Score | 0.0 |
| Maximal Weighted F1 Score | 0.3070 |
| Average Weighted F1 Score | 0.2168 |
| Standard Deviation Weighted F1 Score | 0.0600 |
| Best Epoch (Weighted F1 Score) | 16 |

Table A.23: Model: XLM-RoBERTa, few-shot transfer, with no extra features, argument component classification. Summary of model performance over training epochs.

| Parameter | Value |
|---|---|
| Minimal Proportional F1 Score | 0.0 |
| Maximal Proportional F1 Score | 0.2314 |
| Average Proportional F1 Score | 0.1149 |
| Standard Deviation Proportional F1 Score | 0.0640 |
| Best Epoch (Proportional F1 Score) | 11 |

Table A.24: Model: XLM-RoBERTa, few-shot transfer, with extra features, argument component classification. Summary of model performance over training epochs.

# Bibliography


Artetxe, Mikel, Gorka Labaka and Eneko Agirre (July 2017). 'Learning bilingual word embeddings with (almost) no bilingual data'. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 451–462. DOI: 10.18653/v1/P17-1042. URL: https://www.aclweb.org/anthology/P17-1042.

Bentahar, Jamal, Bernard Moulin and Micheline Belanger (Mar. 2010). 'A Taxonomy of Argumentation Models Used for Knowledge Representation'. In: *Artificial Intelligence Review* 33, pp. 211–259. DOI: 10.1007/s10462-010-9154-1.

Bohnet, Bernd and Joakim Nivre (July 2012). 'A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing'. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, pp. 1455–1465. URL: https://www.aclweb.org/anthology/D12-1133.

Bojanowski, Piotr et al. (2016). 'Enriching Word Vectors with Subword Information'. In: *arXiv preprint arXiv:1607.04606*.

Choi, Jinho (Aug. 2012). 'Optimization of Natural Language Processing Components for Robustness and Scalability'. PhD thesis.

Conneau, Alexis et al. (2020). *Unsupervised Cross-lingual Representation Learning at Scale*. arXiv: 1911.02116 [cs.CL].

Devlin, Jacob et al. (June 2019a). '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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

— (2019b). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv: 1810.04805 [cs.CL].

Donkers, Tim and Jürgen Ziegler (2020). 'Leveraging Arguments in User Reviews for Generating and Explaining Recommendations'. In: *Datenbank-Spektrum* 20.2, pp. 181–187. ISSN: 1618-2162. DOI: 10.1007/s13222-020-00350-y.

Dyer, Chris et al. (July 2015). 'Transition-Based Dependency Parsing with Stack Long Short-Term Memory'. In: *Proceedings of the 53rd Annual*

*Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 334–343. DOI: 10.3115/v1/P15-1033. URL: https://www.aclweb.org/anthology/P15-1033.

Eger, Steffen, Johannes Daxenberger and Iryna Gurevych (July 2017). 'Neural End-to-End Learning for Computational Argumentation Mining'. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 11–22. DOI: 10.18653/v1/P17-1002. URL: https://www.aclweb.org/anthology/P17-1002.

Evensen, Anders Næss (2020). 'Annotation Projection and Cross-Lingual approaches to Argument Mining for Norwegian'. MA thesis. The University of Oslo.

Ghosh, Debanjan et al. (June 2014). 'Analyzing Argumentative Discourse Units in Online Interactions'. In: DOI: 10.3115/v1/W14-2106.

Habernal, Ivan and Iryna Gurevych (Apr. 2017). 'Argumentation Mining in User-Generated Web Discourse'. In: *Computational Linguistics* 43.1, pp. 125–179. DOI: 10.1162/COLI_a_00276. URL: https://www.aclweb.org/anthology/J17-1004.

Harris, Charles R. et al. (Sept. 2020). 'Array programming with NumPy'. In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Huang, Zhiheng, Wei Xu and Kai Yu (2015). *Bidirectional LSTM-CRF Models for Sequence Tagging*. arXiv: 1508.01991 [cs.CL].

Joachims, Thorsten, Thomas Finley and Chun-Nam John Yu (Oct. 2009). 'Cutting-Plane Training of Structural SVMs'. In: *Mach. Learn.* 77.1, pp. 27–59. ISSN: 0885-6125. DOI: 10.1007/s10994-009-5108-8. URL: https://doi.org/10.1007/s10994-009-5108-8.

Kiperwasser, Eliyahu and Yoav Goldberg (2016). 'Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations'. In: *Transactions of the Association for Computational Linguistics* 4, pp. 313–327. DOI: 10.1162/tacl_a_00101. URL: https://www.aclweb.org/anthology/Q16-1023.

Lauscher, Anne et al. (Nov. 2020). 'From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers'. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 4483–4499. DOI: 10.18653/v1/2020.emnlp-main.363. URL: https://www.aclweb.org/anthology/2020.emnlp-main.363.

Lawrence, John and Chris Reed (Dec. 2019). 'Argument Mining: A Survey'. In: *Computational Linguistics* 45.4, pp. 765–818. DOI: 10.1162/coli_a_00364. URL: https://www.aclweb.org/anthology/J19-4006.

— (Jan. 2020). 'Argument Mining: A Survey'. In: *Computational Linguistics* 45.4, pp. 765–818. ISSN: 0891-2017, 1530-9312. DOI: 10.1162/coli_a_00364. URL: https://direct.mit.edu/coli/article/45/4/765-818/93362 (visited on 02/04/2021).

Lawrence, John et al. (June 2014). 'Mining Arguments from 19th Century Philosophical Texts Using Topic Based Modelling'. In: *Proceedings of the First Workshop on Argumentation Mining*. Baltimore, Maryland: Association for Computational Linguistics, pp. 79–87. DOI: 10.3115/v1/W14-2111. URL: https://www.aclweb.org/anthology/W14-2111.

Lee, Heeyoung et al. (2013). 'Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules'. In: *Computational Linguistics* 39.4, pp. 885–916. DOI: 10.1162/COLI_a_00152. URL: https://www.aclweb.org/anthology/J13-4004.

Levy, Omer, Yoav Goldberg and Ido Dagan (2015). 'Improving Distributional Similarity with Lessons Learned from Word Embeddings'. In: *Transactions of the Association for Computational Linguistics* 3, pp. 211–225. DOI: 10.1162/tacl_a_00134. URL: https://www.aclweb.org/anthology/Q15-1016.

Lindahl, Anna, Lars Borin and Jacobo Rouces (Aug. 2019). 'Towards Assessing Argumentation Annotation - A First Step'. In: *Proceedings of the 6th Workshop on Argument Mining*. Florence, Italy: Association for Computational Linguistics, pp. 177–186. DOI: 10.18653/v1/W19-4520. URL: https://www.aclweb.org/anthology/W19-4520.

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. DOI: 10.48550/ARXIV.1907.11692. URL: https://arxiv.org/abs/1907.11692.

Ma, Xuezhe and Eduard Hovy (Aug. 2016). 'End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF'. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074. DOI: 10.18653/v1/P16-1101. URL: https://www.aclweb.org/anthology/P16-1101.

Marsland, Stephen (2009). *Machine Learning - An Algorithmic Perspective.* Chapman and Hall / CRC machine learning and pattern recognition series. CRC Press, pp. I–XVI, 1–390. ISBN: 978-1-4200-6718-7.

McCallum, Andrew Kachites (2002). 'MALLET: A Machine Learning for Language Toolkit'. http://www.cs.umass.edu/ mccallum/mallet.

McDonald, Ryan et al. (Oct. 2005). 'Non-Projective Dependency Parsing using Spanning Tree Algorithms'. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 523–530. URL: https://www.aclweb.org/anthology/H05-1066.

Mikolov, Tomas, Quoc V. Le and Ilya Sutskever (2013). *Exploiting Similarities among Languages for Machine Translation*. arXiv: 1309.4168 [cs.CL].

Mikolov, Tomas et al. (June 2011). 'Extensions of recurrent neural network language model'. In: pp. 5528–5531. DOI: 10.1109/ICASSP.2011.5947611.

Miwa, Makoto and Mohit Bansal (Aug. 2016). 'End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures'. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Compu-

tational Linguistics, pp. 1105–1116. DOI: 10.18653/v1/P16-1105. URL: https://www.aclweb.org/anthology/P16-1105.

Mochales, Raquel and Marie-Francine Moens (Jan. 2009). 'Argumentation mining: The detection, classification and structure of arguments in text'. In: pp. 98–107. DOI: 10.1145/1568234.1568246.

Moens, Marie-Francine et al. (Jan. 2007). 'Automatic Detection of Arguments in Legal Texts'. In: Proceedings of the International Conference on Artificial Intelligence and Law, pp. 225–230. DOI: 10.1145/1276318.1276362.

Morio, Gaku and Katsuhide Fujita (Nov. 2018). 'End-to-End Argument Mining for Discussion Threads Based on Parallel Constrained Pointer Architecture'. In: *Proceedings of the 5th Workshop on Argument Mining*. Brussels, Belgium: Association for Computational Linguistics, pp. 11–21. DOI: 10.18653/v1/W18-5202. URL: https://www.aclweb.org/anthology/W18-5202.

Musi, Elena, Debanjan Ghosh and Smaranda Muresan (Aug. 2016). 'Towards Feasible Guidelines for the Annotation of Argument Schemes'. In: *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. Berlin, Germany: Association for Computational Linguistics, pp. 82–93. DOI: 10.18653/v1/W16-2810. URL: https://www.aclweb.org/anthology/W16-2810.

Paszke, Adam et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. arXiv: 1912.01703 [cs.LG].

Pennington, Jeffrey, Richard Socher and Christopher D. Manning (2014). 'GloVe: Global Vectors for Word Representation'. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Persing, Isaac and Vincent Ng (May 2020). 'Unsupervised Argumentation Mining in Student Essays'. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 6795–6803. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.839.

Potash, Peter, Alexey Romanov and Anna Rumshisky (Sept. 2017). 'Here's My Point: Joint Pointer Architecture for Argument Mining'. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1364–1373. DOI: 10.18653/v1/D17-1143. URL: https://www.aclweb.org/anthology/D17-1143.

Řehůřek, Radim and Petr Sojka (2010). 'Software Framework for Topic Modelling with Large Corpora'. eng. In: *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*. Valletta, Malta: University of Malta, pp. 46–50. ISBN: 2-9517408-6-7. URL: http://www.fi.muni.cz/usr/sojka/presentations/lrec2010-poster-rehurek-sojka.pdf.

Rigotti, E. and S. Greco (2018). *Inference in Argumentation: A Topics-Based Approach to Argument Schemes*. Argumentation Library. Springer International Publishing. ISBN: 978-3-030-04568-5. URL: https://books.google.no/books?id=44J-DwAAQBAJ.

Socher, Richard et al. (Oct. 2013). 'Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank'. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642. URL: https://www.aclweb.org/anthology/D13-1170.

Stab, Christian and Iryna Gurevych (Aug. 2014a). '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, pp. 1501–1510. URL: https://www.aclweb.org/anthology/C14-1142.

— (Oct. 2014b). 'Identifying Argumentative Discourse Structures in Persuasive Essays'. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 46–56. DOI: 10.3115/v1/D14-1006. URL: https://www.aclweb.org/anthology/D14-1006.

— (Sept. 2017a). 'Parsing Argumentation Structures in Persuasive Essays'. In: *Computational Linguistics* 43.3, pp. 619–659. DOI: 10.1162/COLI_a_00295. URL: https://www.aclweb.org/anthology/J17-3005.

— (Sept. 2017b). 'Parsing Argumentation Structures in Persuasive Essays'. In: *Computational Linguistics* 43.3, pp. 619–659. DOI: 10.1162/COLI_a_00295. URL: https://www.aclweb.org/anthology/J17-3005.

Sutton, Charles and Andrew McCallum (2010). *An Introduction to Conditional Random Fields*. arXiv: 1011.4088 [stat.ML].

team, The pandas development (Feb. 2020). *pandas-dev/pandas: Pandas*. Version latest. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

Toulmin, S.E. and Dawsonera (2003). *The Uses of Argument*. Cambridge University Press. ISBN: 978-0-521-53483-3. URL: https://books.google.no/books?id=8UYgegaB1S0C.

Trautmann, Dietrich (Dec. 2020). 'Aspect-Based Argument Mining'. In: *Proceedings of the 7th Workshop on Argument Mining*. Online: Association for Computational Linguistics, pp. 41–52. URL: https://www.aclweb.org/anthology/2020.argmining-1.5.

Trautmann, Dietrich et al. (2019). 'Robust Argument Unit Recognition and Classification'. In: *CoRR* abs/1904.09688. arXiv: 1904.09688. URL: http://arxiv.org/abs/1904.09688.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

Velldal, Erik et al. (May 2018). 'NoReC: The Norwegian Review Corpus'. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1661.

Wambsganß, Thiemo, Nikolaos Molyndris and Matthias Söllner (Mar. 2020). 'Unlocking Transfer Learning in Argumentation Mining: A Domain-Independent Modelling Approach'. In: pp. 341–356. ISBN: 9783955453350. DOI: 10.30844/wi_2020_c9-wambsganss.