



PADERBORN UNIVERSITY
The University for the Information Society

Department of Electrical Engineering,
Computer Science and Mathematics
Warburger Straße 100
33098 Paderborn

Master's Thesis

**Assessing the Argument Quality of Persuasive
Essays using Neural Text Generation**

Timon Gurcke

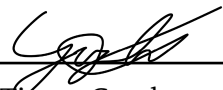
- 1. Reviewer* **Jun. Prof. Dr. Henning Wachsmuth**
Department of Computer Science
Paderborn University
- 2. Reviewer* **Prof. Dr. Oliver Müller**
Department of Business Information Systems
Paderborn University
- Supervisor* **Milad Alshomary**

January 4, 2021

Declaration

I hereby declare that the thesis I am submitting is entirely my own original work except where otherwise indicated.

Paderborn, January 4, 2021



Timon Gurcke

Acknowledgement

I would like to thank my supervisor Milad Alshomary for keeping me on track throughout this thesis and supporting me with his critical thoughts and suggestions. Furthermore, I would like to thank Jun. Prof. Dr. Henning Wachsmuth for giving me the opportunity to work on such an interesting topic, and Prof. Dr. Oliver Müller for taking the time to review this thesis. I would also like to thank the Web Technology and Information Systems (WEBIS) group at Bauhaus University, led by Prof. Dr. Benno Stein, for providing the computational resources to perform the experiments of this thesis. Furthermore, I would like to thank Prof. Dr. Iryna Gurevych and Dr. Christian Stab for giving me access to their code to build the baselines in this thesis. I would also like to thank Katharina Brenning, Le Tuan Anh Ha, Abdullah Burak, and Dr. Maurice Gurcke for taking the time to participate in the annotation study. I would also like to thank Stefan Werner and Abdullah Burak for many late-night conversations regarding my thesis. Finally, my biggest thanks go to Leah Ziegenbein for motivating me and providing an environment at home where I can focus.

Abstract

In this thesis, we will explore the potential of large pre-trained Language Models to assess the *Local Sufficiency* quality dimension of an argument. We study two different approaches: (1) Directly assessing the *Local Sufficiency* of an argument as a (binary) classification task and (2) Indirectly assessing *Local Sufficiency* of an argument by generating a conclusion based on a set of premises first and afterward use the generated conclusion to augment the (binary) classification task in (1). We establish a new state-of-the-art *Local Sufficiency* assessment approach using the BERT model achieving 96.7% of human performance. Subsequently, we show that leveraging *Argument Mining* to obtain argumentative units and thus, following the *Local Sufficiency* definition more strictly, decreases the assessment performance. We also investigate the reasons for this behavior. In addition, we study the task of conclusion generation and its similarity to other tasks in the NLP domain using the BART model. We show that the conclusions generated by our models are of equal quality and could not be discriminated from those written by humans in a manual ranking study. Furthermore, we find that multiple different conclusions are equally likely to be drawn without further context of the conclusion target given a set of premises. Finally, using the generated conclusion to augment the *Local Sufficiency* assessment approach led to no performance improvements. Still, it revealed that, given the currently available datasets, the *Local Sufficiency* assessment of arguments relies mostly on the given premises and not on the corresponding conclusions, displaying the need for further research in this direction to better fulfill the task of *Local Sufficiency* assessment.

Contents

1	Introduction	1
1.1	Research Questions	2
1.2	Approaches	3
1.3	Results and Contributions	4
1.4	Outline	6
2	Background and Related Work	7
2.1	Natural Language Processing and its Objectives	7
2.1.1	Text Classification, Summarization and Translation	8
2.2	Neural Networks in Natural Language Processing	10
2.2.1	Recurrent Neural Networks	10
2.2.2	Sequence-to-Sequence	12
2.2.3	Attention	13
2.2.4	Transformers	15
2.3	Language Modeling, Text Classification and Text Summarization	18
2.3.1	Language Modeling	18
2.4	Argumentation, Argument Mining, and Argument Quality	28
2.4.1	Argumentation	28
2.4.2	Computational Argumentation	32
3	Data for Local Sufficiency Assessment and Conclusion Generation	37
3.1	Existing Corpora	37
3.1.1	Persuasive Essays	37
3.1.2	Local Sufficiency Assessment	40
3.2	Corpus Transformation and Creation	42
3.2.1	Corpus Transformation	42
4	Approaches and Implementation	45
4.1	Direct Local Sufficiency Assessment	45
4.2	Conclusion Generation	47
4.3	Indirect Local Sufficiency Assessment	50
5	Experiments and Evaluation	51
5.1	Direct Local Sufficiency Assessment	51
5.2	Conclusion Generation	57

5.2.1	Automatic Evaluation	57
5.2.2	Manual Evaluation	62
5.3	Indirect Local Sufficiency Assessment	68
6	Conclusion	71
6.1	Future Outlook	73
A	Appendix	75
A.1	Experiments and Evaluation: Conclusion Generation	75
A.1.1	Automatic Evaluation	75
A.1.2	Manual Evaluation	79
	Bibliography	83

Introduction

In recent years, deep learning and the ability to process large amounts of data have accelerated progress in Natural Language Processing (NLP). Numerous work focuses on a general representation and generation of human language, that can be applied to approach a wide range of problems involving natural language text. Similar to Language Modeling approaches, the analysis of arguments strives to understand underlying concepts, relationships, and logic. This field referred to as *Computational Argumentation* focuses on the analysis and synthesis of arguments, which are used in many applications, e.g., virtual assistants (Rinott et al., 2015) and search engines (Stab et al., 2018; Wachsmuth et al., 2017b).

In the past, researchers have developed approaches to extract arguments from natural language text. These approaches belong to the task of *Argument Mining*, which not only aims to extract argumentative discourse units (Ajjour et al., 2017) but also tries to find different sub-structures within arguments (Stab, 2017). While the problem of *Argument Mining* has not yet been solved completely, it has supported the exploration of several downstream tasks.

One of the most important tasks in Argumentation Theory is the automatic evaluation of student essays (Shermis and Burstein, 2003), which aims to measure the quality of an argumentation written by a student on a specific topic. However, argument search engines (Stab et al., 2018; Wachsmuth et al., 2017b), which provide pro and contra arguments for a particular topic, consider not only student essays but also arguments from numerous other domains, e.g., comments or newspaper articles. In addition, these search engines must assess arguments to decide how to present them to the user. Existing approaches either use a holistic assessment scheme (Gretz et al., 2019a) or approach the problem through a deviation into several subproblems, often referred to as quality dimensions (Wachsmuth et al., 2017a). While a holistic approach is promising due to its simplicity, it is less useful in areas that require feedback (Shermis and Burstein, 2003). This is of interest not only from computer scientists' perspective but also from a social and philosophical point of view (Aristotle, 2007). Hence, the evaluation of the quality of arguments is sometimes regarded as the "ultimate question" in *Argument Mining* (Stede and Schneider, 2018).

In this work, following the definitions in Wachsmuth et al., (2017a), we assess the *Local Sufficiency* of an argument, measuring whether the premises given in an argument are sufficient and together make it rational to draw the proposed conclusion. For this purpose, we study two approaches that build upon the success of state-of-the-art (SOTA) NLP models in recent years.

1.1 Research Questions

In recent years, the newly proposed methods, which are now widely used in NLP tasks, have changed considerably. Models have become more refined, and the amount of data processed greatly improved. Research has shifted from the predominant use of hand-crafted features to the use of deep learning. Newly developed models can not only generate language representations, which can be used for direct classification or regression but can generate natural language text on its own. Furthermore, they allow transferring general language understanding knowledge, obtained through pre-training on massive corpora, to tasks where this knowledge is of use but cannot be fully inferred due to a lack of available data. Although these models have so far been adapted towards the holistic assessment of argumentation quality (Gretz et al., 2019b; Toledo et al., 2019), there is still great potential.

Considering the definitions of argument quality dimensions proposed in Wachsmuth et al., (2017a), one dimension is particularly useful to explore this potential, i.e., *Local Sufficiency*. As of now only Stab and Gurevych, (2017b) assessed the *Local Sufficiency* in argumentative essays. However, the approach is mainly based on hand-crafted features and model semantics only as n-grams or universal word embeddings. Moreover, the authors did not use argumentative features, i.e., only conclusions and premises instead of the whole text containing them, to create their features. In contrast, Wachsmuth et al., (2016) showed that the use of argumentative features created based on argumentative units improves the performance of assessing the quality of an argument in multiple dimensions. This raises the question of how else argumentative units can be used to improve performance in this domain. Other LSTM-based approaches have focused on the relative *convincingness* (Habernal and Gurevych, 2016a; Potash and Rumshisky, 2017; Simpson and Gurevych, 2018) of arguments or their *evidence* (Gleize et al., 2019). However, approaches that assess the quality of arguments using transformer-based architectures are still very limited (Gretz et al., 2019b; Toledo et al., 2019) and only focused on a holistic assessment in contrast to the more fine-grained view of Wachsmuth et al., (2017a).

To assess *Local Sufficiency*, Language Modeling based models, which have achieved SOTA results in other NLP areas, seem to be a promising choice, as they avoid the

necessity of crafting features by hand and allow to directly operate on the level of argumentative units in natural language form. The potential of these methods to directly output *Local Sufficiency* scores based on given argumentative units could thereby lead to substantial improvements. In addition, their ability to generate text allows studying a wide range of augmented assessment approaches. In our case, that is to first try to generate a conclusion based on a set of premises and in a second step infer a quality dimension score, e.g., by comparing the generated conclusion to the ground truth conclusion. Finally, it is important to investigate how well an augmented assessment (using generation) performs compared to a more direct approach to capture the trade-off between accurately predicting the quality dimension score and providing interpretable intermediate results (the generated conclusion). Therefore, the idea of directly assessing quality dimensions of arguments using contextual embedding methods and an augmented assessment through generation seem to be a novel and promising tasks, which can contribute to the field of *Computational Argumentation* in different ways. In particular, our work tries to answer and explore the following questions:

1. Can *Language Models* incorporating implicit knowledge gained through pre-training and fine-tuning help improve the assessment of *Local Sufficiency*?
2. To what extent can pre-trained Language Models be used to learn to generate proper conclusions given a set of premises?
3. How can text generation, i.e., conclusion generation be used to assess the *Local Sufficiency* of an argument?

1.2 Approaches

In this work, we create two different *Local Sufficiency* assessment approaches. Figure 1.1 shows an overview of the components used to create the final assessment models. Our models can be distinguished according to their basic approach, namely direct *Local Sufficiency* assessment and indirect *Local Sufficiency* assessment through generation. For direct *Local Sufficiency* assessment, we used the BERT (Devlin et al., 2018) transformer model and adapted it to the task of binary text classification. In total, we evaluate our approach by comparing it to baseline of Stab and Gurevych, (2017b), on the AAE-v2 dataset (Stab and Gurevych, 2017a) and two adaptations of it. For indirect *Local Sufficiency* assessment we create three models based on the BART (Lewis et al., 2019) transformer model: one without fine-tuning, one fine-tuned on the CNN-DailyMail (Nallapati et al., 2016) extractive news summarization dataset and one fine-tuned on the XSum (Narayan et al., 2018) abstract news summarization

dataset. We then investigate the performance of conclusion generation models using a wide range of automated metrics combined with a manual ranking study involving five annotators. Finally, we select the best of the conclusion generation models to produce conclusions for the entire dataset and use these to train a BERT based model that takes them as an extra input to predict the *Local Sufficiency* of an argument.

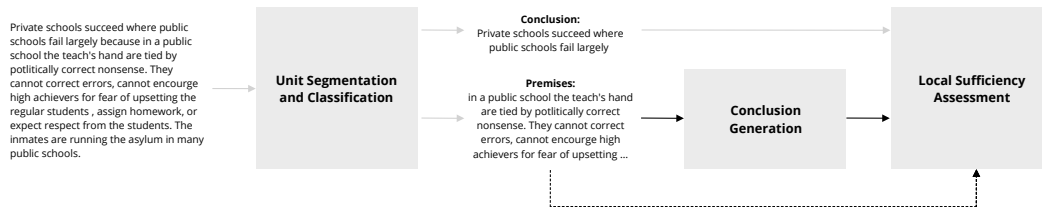


Figure 1.1.: Overview of the planned approaches: After extracting the conclusion and premises from the input text, the premises are used to generate a conclusion so that the *Local Sufficiency* assessment is performed either directly based on the conclusion and premises (dashed black lines) or indirectly using the conclusion and the generated conclusion (solid black lines).

1.3 Results and Contributions

Our results show that large-scale, pre-trained Language Models can successfully improve the prior SOTA of *Local Sufficiency* assessment. This said we found that the removal of non-argumentative text from arguments decreases the performance of *Local Sufficiency* assessment models in general due to the loss of contextual and connectivity information, i.e., textual markers for opposing views or the number of premises. In contrast, removing textual markers indicating the use of an example improves performance. Overall, however, the decrease significantly outweighs the increase in performance. Augmenting the task of assessing the *Local Sufficiency* of an argument using generated conclusions, we found that given a set of premises, multiple conclusion targets, and thus different conclusions, are viable choices for the human observer. In addition, we found that while the task of abstractive news summarization comes closest to generating conclusions, the fine-tuning on our dataset makes its potential transfer learning benefit negligible. Ultimately, given a set of premises, five human annotators could not distinguish between machine-generated conclusions and conclusions written by humans. Finally, comparing the generated conclusions to the ground truth conclusions and using the generated conclusions as an additional input to our evaluation model did not improve the *Local Sufficiency* classification performance. With that said, we found that premises are significantly more important than the corresponding conclusions in assessing the *Local Sufficiency* of an argument, which is likely a product of the available datasets and its, on average, high quality of arguments.

Leveraging the potential of large-scale pre-trained transformer models in combination with the theoretical background of argument quality assessment, in this work, we contribute to solving the research questions in the following way:

1. Combining the work of Stab and Gurevych, (2017a) and Stab and Gurevych, (2017b), we created a new dataset that presents a single argument as a conclusion, a set of premises, and a binary *Local Sufficiency* score.
2. We studied the effects of removing non-argumentative text from arguments in *Local Sufficiency* assessment.
3. We created a *Local Sufficiency* assessment model that outperforms the previous SOTA model by Stab and Gurevych, (2017b) and achieves 96.7% of human-level performance.
4. We performed a first analysis of the relationship between well-known NLP tasks and the conclusion generation task.
5. We created automated models that can generate conclusions that, given a set of premises, are equally probable to be inferred as conclusions written by a human.
6. We explored a variety of ideas to use argumentative features, i.e., the generated conclusions, to improve the *Local Sufficiency* assessment of arguments.

In summary, we hope to contribute valuable insights into the potential of text generation approaches in the context of *Computational Argumentation* and demonstrate the general potential of large-scale pre-trained Language Models in this area.

1.4 Outline

In Chapter 2, we provide the necessary background knowledge we use in our work. We give a brief introduction to Natural Language Processing and its objectives in Section 2.1, followed by an explanation of the technical parts of the models used in this thesis in Section 2.2 and of the models themselves in Section 2.3. Finally, we conclude the background chapter by introducing important related ideas in Argumentation Theory and *Computational Argumentation* and provide an overview of the prior state-of-the-art (SOTA) models that attempt to assess *Local Sufficiency* in Section 2.4. In Chapter 3, we provide a detailed introduction to the data used in our work, introducing relevant corpus criteria based on characteristics introduced in the background chapter. We explain currently available datasets in Section 3.1, followed by a description of our new dataset in Section 3.2. Next, we provide implementation details as well as our experiment setups in Chapter 4, which are divided into the direct *Local Sufficiency* assessment approach in Section 4.1, the conclusion generation approach in Section 4.2 and the indirect *Local Sufficiency* assessment approach in Section 4.3. Next, we evaluate all our experiments in Chapter 5, starting with the *Local Sufficiency* assessment approach in Section 5.1, the conclusion generation approach in Section 5.2 and the indirect *Local Sufficiency* assessment approach in Section 5.3. Finally, we discuss and conclude our work in Chapter 6 and state ideas that we think are interesting to explore in the future in Section 6.1.

Background and Related Work

This chapter will introduce the background knowledge we use in this work: (1) We will introduce Natural Language Processing (NLP) and its tasks related to this work. (2) We will explain techniques and model architectures, which are prerequisites for understanding the models that we will use later. (3) We will describe these models in detail, have a closer look at how they are trained, and how they tackle the tasks introduced before. (4) We will present the domain of *Computational Argumentation* and the domain-specific tasks, including concepts regarding arguments' quality.

2.1 Natural Language Processing and its Objectives

Natural Language Processing (NLP) is one of the core parts of text mining. Its ultimate goal is to discover, identify and structure previously unknown information from natural language text¹. Chowdhary, (2020) define the tasks of NLP researches the following: "NLP researchers aim to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform the desired tasks." During the past decades, the amount of tasks and data in the NLP domain has grown continuously, covering a wide variety of different tasks, e.g., speech recognition and question answering. While each task holds its own challenges, some problems appear to be related to almost all natural language. The most common of these problems is the ambiguity of natural language text. Ambiguity describes the property of text to change its meaning based on context, speech, and presuppositions. In the sentence "I saw a kid with a telescope." for example, it is unclear whether the kid has the telescope or if it is seen through a telescope. In contrast to humans who can try to resolve these problems based on experience or phonetics, machines are often limited to one type of input and thus struggle with concepts that are subconsciously used in everyday talks. Liddy, (1998) and Feldman, (1999) propose the following levels of language analysis, which are commonly used to evaluate the meaning of natural language text:

¹Following the slides of the "Introduction to Text Mining" course by Henning Wachsmuth at Paderborn University (<https://en.cs.uni-paderborn.de/de/css/teaching/courses/text-mining-w19>).

- **Phonetics or Phonology:** Physical and linguistic sound of speech.
- **Morphology:** The smallest part of a word and its meaning, e.g., word stems, prefixes, and suffixes.
- **Syntax:** Grammatical relationships of words/sentences, e.g., nouns, verbs, relative clauses.
- **Semantics:** Lexical and contextual meaning of words/sentences.
- **Discourse:** Structural meaning of larger text units, e.g., type of text, document structure.
- **Pragmatics:** External knowledge about the world.

Looking at the methods applied in practice, NLP often follows a pipeline approach that involves pre-processing the raw input text. This phase is usually used to extract information related to the levels of language analysis presented previously. The extracted information, often referred to as features, is then used, alone or in conjunction with the raw text, to create an algorithmic model capable of solving a specific task.

2.1.1 Text Classification, Summarization and Translation

In our work, we will face two different NLP tasks. First, we will classify text in a binary setting to predict whether an input text consisting of a conclusion and a set of premises is sufficient following the definition of Wachsmuth et al., (2017a). Second, we train a model to generate conclusions based on a set of premises. To do this, we will reuse methods from another task in NLP, i.e., text summarization. Text summarization can be divided into two types of approaches. This is extractive summarization and abstractive summarization. The former creates a summary of an input text by copying and fusing parts of the input text, while the latter creates a summary using newly generated content. Since conclusion generation is a reasoning task that requires abstraction instead of picking and fusing parts of the premises (extraction), our focus lies on approaches that successfully tackle abstractive summarization. However, we will investigate both approaches in this work. In addition, we will refer to the NLP task known as machine translation. Machine translation aims to create an approach that is capable of translating from one language to another. Famous examples which employ these techniques are

Google Translator² and more recently DeepL³. While translation is not our goal, many of the past years' major improvements were initially developed to solve this problem and only later adapted and applied towards the tasks in scope. Therefore, it is important to understand that while both text summarization and translation are text generation tasks, their input-output relation differs substantially, e.g., the length of a translated text is usually close to the input text. In contrast, the length of summarization is much shorter than the corresponding input text.

²<https://translate.google.com>

³<https://www.deepl.com/translator>

2.2 Neural Networks in Natural Language Processing

This section will provide an overview of models and techniques typically used in Natural Language Processing and are important for our experiments. As natural language is mostly sequential, meaning that the order of words in a text is important, we will explain Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986; Werbos, 1990) as a special form of Neural Networks that can process this special type of data. We will also explain an adaptation of RNNs called Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). Based on these, we will introduce the task of Sequence-to-Sequence learning, which is in our case to generate a conclusion based on a set of premises. Finally, we will explain the architectural structures that are key to the success of today's language models, such as BERT and GPT-2.

2.2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986; Werbos, 1990) are a special form of Neural Networks that were developed to process sequential, often temporal data. Unlike feed-forward Neural Networks, RNNs do not process the entire input data simultaneously but process it one bit (often a word) at a time using the same procedure. However, the procedure itself is not only influenced by the current input but also by the previously generated output and/or intermediate state. Depending on the task, RNNs allow for different structures (Figure 2.1).

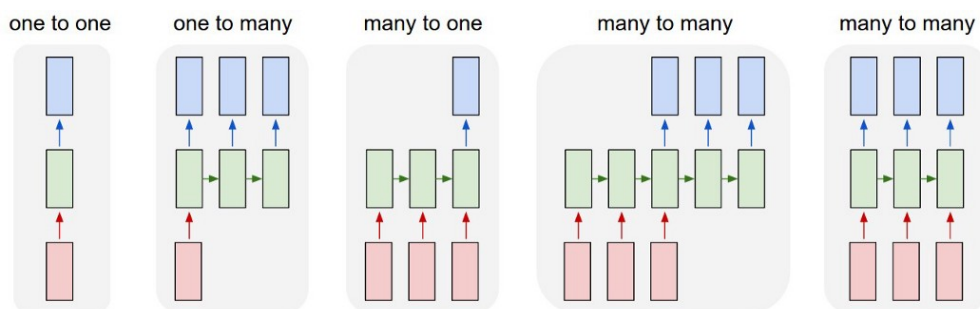


Figure 2.1.: RNN tasks depending on the number of input and output features. Taken from a blogpost of Andrej Karpathy⁴.

One of the most used forms of RNNs are Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) Neural Networks. While keeping the original idea, the authors change the cells' internal structure to adjust for long time relationships

⁴<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

in the data and deal with the vanishing gradient problem that vanilla RNNs were suffering from.

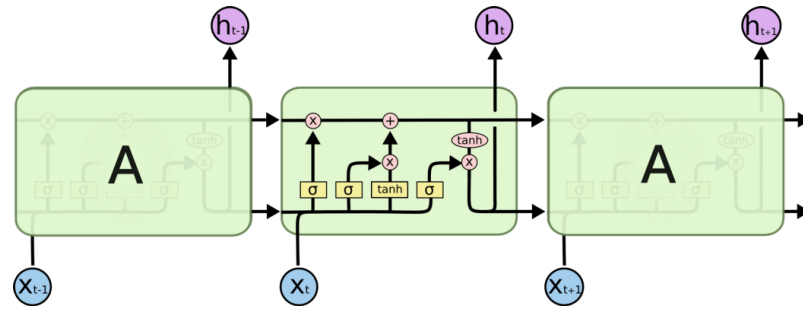


Figure 2.2.: Internal structure of an LSTM unit. Taken from a blogpost of Christopher Olah⁵.

The internal structure of an LSTM unit holds a gating system that consists of a cell c that acts as a memory component and three gates that regulate the amount of information that flows through the network. First, the input gate i controls the amount of information that flows into the cell. Second, the forget gate f controls how much of the cell's information is kept. Third, the output gate controls how much of the information in the cell is used to compute the output activation of the LSTM unit. Figure 2.2 shows the internal structure of an LSTM unit.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1}) \quad (2.1)$$

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1}) \quad (2.2)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1}) \quad (2.3)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1}) \quad (2.4)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (2.5)$$

$$h_t = o_t \circ \sigma_h(c_t) \quad (2.6)$$

Where i_t , f_t and o_t are the activation's of the input, output and forget gate at time t . Correspondingly x_t is the input at time t and h_{t-1} the hidden state of the previous LSTM unit. As shown in Equations 2.1 - 2.6, all gates are similar in structure, while the cell depends on input and forget gate as well as on its own gate-like structure \tilde{c} . The current LSTM unit output h_t depends on the output gate and the cell state. σ_g is a sigmoid function while σ_c and σ_h are tanh functions. The output of all gates depends on their weight matrices $W \in \mathbb{R}^{h \times d}$ and $U \in \mathbb{R}^{h \times h}$, which are learned during the training of the network, where h is the number of hidden units and d the number of input features. Note that we have omitted the bias term in Equations 2.1 - 2.6 for better readability.

⁵<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

2.2.2 Sequence-to-Sequence

The idea to use Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986; Werbos, 1990) for sequence to sequence (Seq2Seq) learning, was first introduced by Sutskever et al., (2014). The authors used the Long short-term memory (LSTM) idea of Hochreiter and Schmidhuber, (1997), which is a special form of an RNN that is particularly useful to account for long term relationships in texts. The general structure includes an encoder that sequentially reads in the input text and creates a context vector that is then used by a decoder to create text, token by token until a special end-of-sentence token is produced (Figure 2.3).

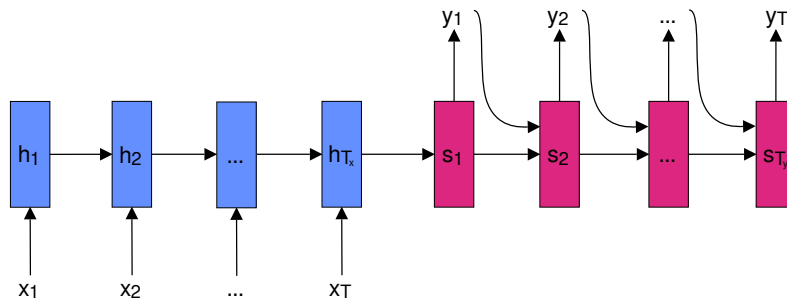


Figure 2.3.: Seq2Seq model that has an encoder (blue) and a decoder (red). The encoder sequentially reads an input (x_1, \dots, x_{T_x}) and the decoder sequentially produces an output (y_1, \dots, y_{T_y}) .

Let $\mathbf{x} = (x_1, \dots, x_{T_x})$ be a sequence of input vectors where x_t represents the word of an input sentence \mathbf{x} at time t . Furthermore, let $\mathbf{y} = (y_1, \dots, y_{T_y})$ be a sequence of output vectors where y_t represents the word of an output sentence \mathbf{y} at time t . The task of an encoder is to create a representation c based on \mathbf{x} that serves as an input to the corresponding decoder, which will then generate an output \mathbf{y} . Following the ideas of Bahdanau et al., (2014) the output of the encoder depends on two functions. First f (Equation 2.7), which generates the hidden state at a specific point in time h_t based on the vector representation of the input word at this time x_t and the previous hidden state h_{t-1} . Second a function q that takes all hidden states and creates the representation c based on these states (Equation 2.8).

$$h_t = f(x_t, h_{t-1}) \quad (2.7)$$

$$c = q(\{h_1, \dots, h_{T_x}\}) \quad (2.8)$$

In encoder-decoder architectures q is often considered to take the last hidden state h_{T_x} , thus setting $c = h_{T_x}$. After c is generated the decoder has to generate \mathbf{y} based on it. As the decoder itself is also a RNN it generates the output \mathbf{y} sequentially such that y_t is a conditional probability function g (Equation 2.9) that depends on the

previously predicted words y_{t-1} , the current hidden state s_t and the context vector generated by the encoder c . Broadly speaking g calculates the probability for each word to be chosen at a specific point in time based on previously chosen words and the context of the input. These probabilities are then combined to define a probability over an entire output sentence y (Equation 2.10).

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c) \quad (2.9)$$

$$p(\mathbf{y}) = \prod_{t=1}^T p(y_t | \{y_1, \dots, y_{t-1}\}, c) \quad (2.10)$$

2.2.3 Attention

However, the previously described encoder-decoder architecture has some disadvantages. As c is the same at every time-step t , it contains a general representation of the input context that covers information about the whole input text. In contrast the authors of Bahdanau et al., (2014) show that using a distinct context for each time-step greatly improves the performance at the task of translation. This idea that allows an output word to learn which parts of the input text are relevant for the current step of generation, is referred to as attention. Technically, the previously described function g does now depend on c_i that is the context vector c at time i (Equation 2.11).

$$p(y_i | y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \quad (2.11)$$

This new context vector is calculated as the weighted sum of all hidden states in the encoder. The weighting is calculated using a softmax function (Equation 2.12) applied to the output of a feedforward neural network $score(s_{i-1}, h_j)$ which takes into account the hidden state of the encoder at time h_j and the previous hidden state of the decoder s_{i-1} (Equation 2.13).

$$c_i = \sum_{j=1}^{T_x} \frac{\exp(score(s_{i-1}, h_j))}{\sum_{k=1}^{T_x} \exp(score(s_{i-1}, h_k))} h_j \quad (2.12)$$

$$score(s_{i-1}, h_j) = W_3^T \tanh(W_1 h_j + W_2 s_{i-1}) \quad (2.13)$$

Where $W_1 \in \mathbb{R}^{n \times n}$, $W_2 \in \mathbb{R}^{n \times m}$ and $W_3 \in \mathbb{R}^n$ are weight matrices and m and n are the number of hidden units in the encoder and decoder respectively. This specific kind of attention is often referred to as additive attention.

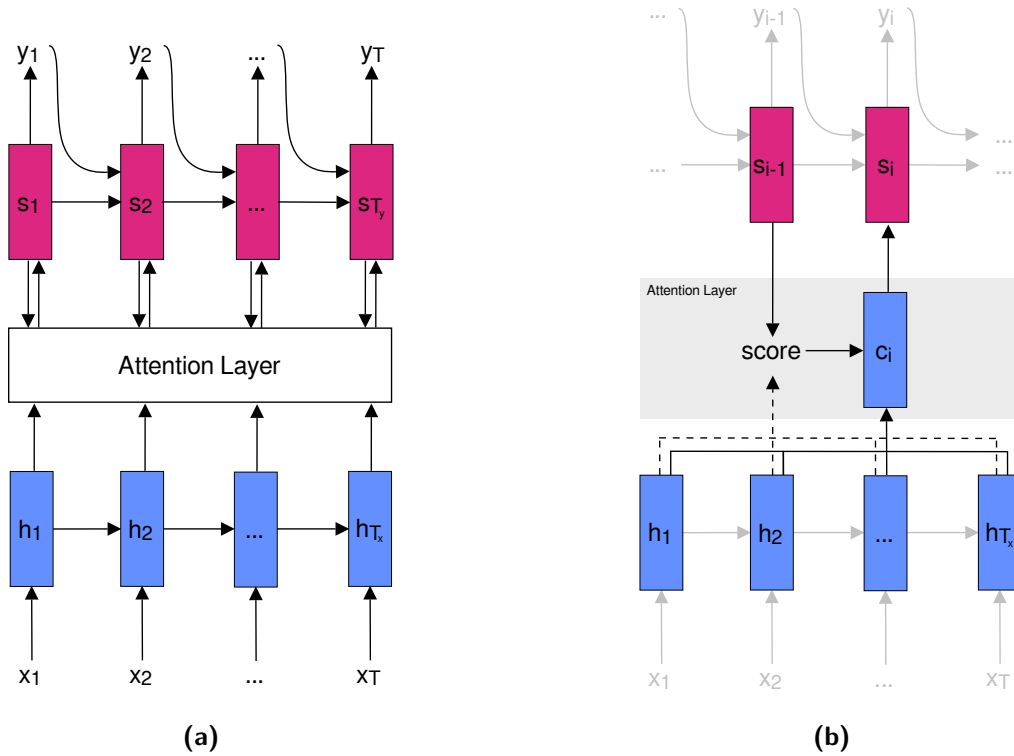


Figure 2.4.: (a) Seq2Seq structure using an attention layer and (b) Internal attention layer overview.

Generalizing Attention

While the general idea of attention stays the same, there exist several adaptations that differ in the way how the weighting is computed (Cheng et al., 2016; Graves et al., 2014; Luong et al., 2015; Vaswani et al., 2017). In order to better understand the differences of these, we will use a generalized annotation for attention that is consistent to the one used by Devlin et al., (2018). The way we described attention above there are three different inputs, which are necessary to compute the final weighting, that are independent of the architecture (in our case an RNN). In order to describe these inputs, let Q be a set queries, K a set of keys and V a set of values. Then we can rewrite the previous formulas as shown in Equations 2.14 and 2.15.

$$Attention(Q, K, V) = c_i = \sum_{j=1}^{T_x} \frac{\exp(\text{score}(Q_{i-1}, K_j))}{\sum_{k=1}^{T_x} \exp(\text{score}(Q_{i-1}, K_k))} V_j \quad (2.14)$$

$$\text{score}(Q_{i-1}, K_j) = W_3^T \tanh(W_1 K_j + W_2 Q_{i-1}) \quad (2.15)$$

Attention vs. Self-Attention

One of the adaptations of the attention mechanism is known as inter-attention or self-attention (Cheng et al., 2016). In contrast to the attention mechanism presented before, self-attention solely relies on the input of the encoder to create the attention scoring. Figure 2.5a and 2.5b show the different parts of attention in their generalized form applied to the RNN structure of Graves et al., (2014) and the corresponding self-attention adaptation.

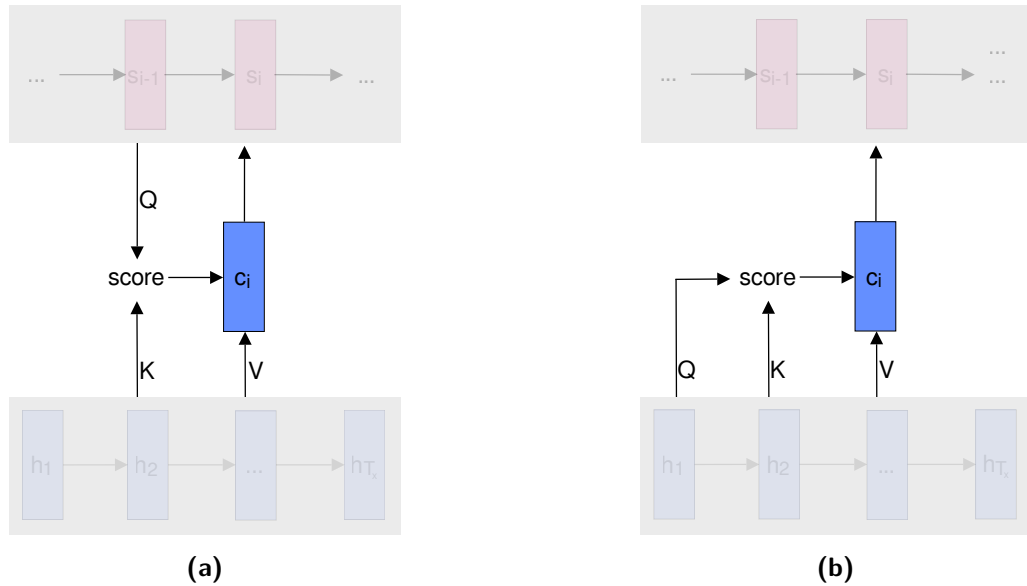


Figure 2.5.: Generalized attention mechanism (a) as represented in Figure 2.4b and the corresponding self-attention mechanism (b).

2.2.4 Transformers

Unlike the previous architectures that are typically RNNs or LSTMs, which use the attention idea, Vaswani et al., (2017) introduced a new architecture entirely built on the self-attention mechanism. The architecture itself consists of a multilayer encoder and a multilayer decoder. Each layer holds a multi-head attention module followed by a feed-forward neural network (Figure 2.6). Multi-head attention modules in the encoder rely on normal self-attention. In contrast, the decoder holds two multi-head attention modules, one who takes the decoder's output as keys K and values V and one which generates the corresponding input query Q . The latter is called a masked multi-head attention module that performs self-attention on previous predictions (i.e., generated tokens) and masks all future positions in the sequence, allowing parallelizing training since the ground truth targets are known. However, during inference, the masked multi-head attention module works sequentially, as masking is not possible due to the unknown target. Thus, the final multi-head attention module

accounts for the input and monitors what it has already generated, similar to RNNs or LSTMs.

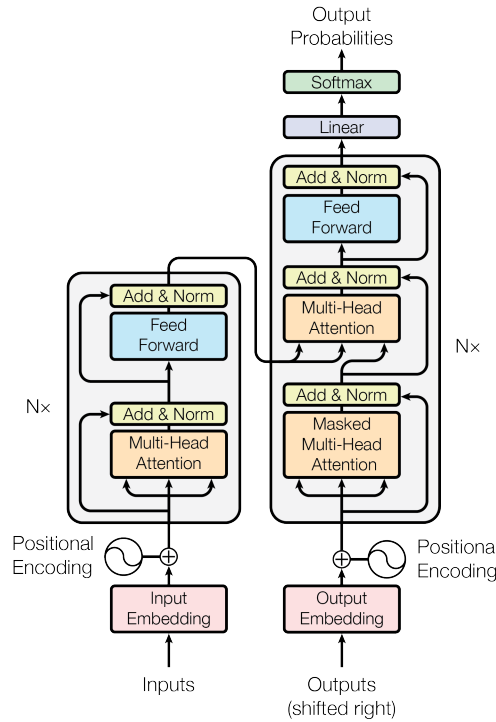


Figure 2.6.: Encoder (left) and decoder (right) of the Transformer model architecture from Vaswani et al., (2017).

In contrast to RNNs and LSTMs, the input text is no more sequentially read but all at the same time, thus to keep positional information, the authors introduce positional embeddings which are added to the original input embeddings of the text (Equations 2.16 and 2.17). A positional embedding is considered a sinusoidal function based on a token position in a text pos and the embedding dimension i .

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (2.16)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (2.17)$$

The model can thus learn the position of all words based on the pattern applied to the word embeddings. The authors use scaled dot-product attention (Formula 2.19 and Figure 2.7a), which takes a set of values V , a set of keys K and a set of queries Q . Scaled Dot-Product attention is a dot-product attention (Equation 2.18) scaled based on the dimensionality of queries and keys d_k in order to cope for vanishing gradients caused by a large value of d_k . Dot-Product attention instead of additive attention allows all calculations to work on matrices that can be highly optimized.

$$\text{score}(Q_{i-1}, K_j) = Q_{i-1}^T K_j \quad (2.18)$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.19)$$

The key component introduced by the authors is the multi-head attention mechanism (Figure 2.7b). Before running scaled dot-product attention, all queries, keys and values are mapped into lower dimensional spaces by using the weight matrices $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ correspondingly (Equation 2.20). Where d_v and d_k are the number of dimensions in the values and keys and d_{model} the number of dimensions the model is supposed to output. The Scaled Dot-Product Attention mechanism is then computed h times in parallel and the results are concatenated and fed through another linear layer using the weight matrix $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ to recreate the original dimensionality (Equation 2.21).

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.20)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.21)$$

Each Scaled Dot-Product Attention mechanism, computed in parallel, is called a head that can attend to different parts of the original input. The authors use an output dimensionality of 512 and 8 heads in their base model, resulting in a dimensionality of $512/8=64$ for each head.

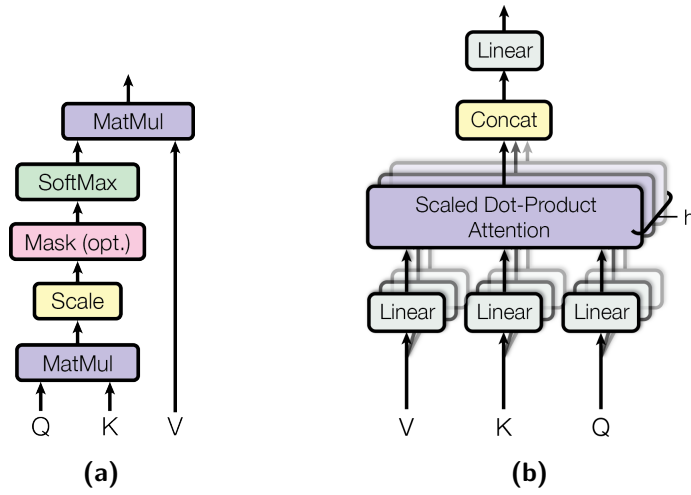


Figure 2.7.: (a) Scaled Dot-Product attention and (b) Multi-head self-attention from Vaswani et al., (2017).

2.3 Language Modeling, Text Classification and Text Summarization

This section will introduce the task of Language Modeling (LM) that is widely used for model pre-training. Pre-trained models are supposed to contain a general understanding of language syntax and semantics. This knowledge can be used for a wide variety of downstream tasks, e.g., text classification or question answering. Subsequent, we will explain the models which we will use in our experiments namely GPT (Radford et al., 2018), GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2018) and BART (Lewis et al., 2019). Finally, we will discuss the major differences in model pre-training and introduce the terms Autoregression and Autoencoding.

2.3.1 Language Modeling

Previously, we have introduced the idea of predicting an output word based on previously generated output words and a context vector which, generated by an encoder, represents the original input sentence (Equations 2.9 and 2.10). This idea is a special case of Language Modeling, which describes the general idea to predict the probability of an output word w_n (Equation 2.23) or a sequence W of words (Equation 2.22) based on a set of input words.

$$p(W) = p(w_1, w_2, \dots, w_n) \quad (2.22)$$

$$p(w_n | w_1, w_2, \dots, w_{n-1}) \quad (2.23)$$

Using chain rule on the conditional probabilities the probability of a sequence can be written as:

$$p(W) = \prod_{i=1}^n p(w_i | w_1, w_2, \dots, w_{i-1}) \quad (2.24)$$

However, the Equations 2.22, 2.23 and 2.24 only describe unidirectional models. In our context, this means that only words that appear before the word which we are trying to predict (on the left side) are used to compute its probability. In contrast, bidirectional models use both words before and after the target word to estimate this probability (Equation 2.25). While this idea is less useful for tasks where the next word is unknown, i.e., Text Generation; Language Modeling, which is usually used for model pre-training to obtain general Natural Language Understanding (NLU), is shown to benefit from leveraging the entire context surrounding a word (Devlin et al., 2018).

$$p(w_n | w_1, w_2, \dots, w_{n-1}, w_{n+1}, w_{n+2} \dots, w_N) \quad (2.25)$$

Unlike the work introduced in Chapter 2.2 which is mostly focused on the task of machine translation, LM simply takes an input word or sequence and predicts the next word. This condition allows for tackling a common problem in NLP. While the Natural Language Processing domain includes a wide range of different tasks, the amount of data effectively labeled for each task is often insufficient to obtain Natural Language Understanding. To be more precise: Solving natural language-related problems usually requires background knowledge about language in general, e.g., about words that convey a similar meaning or the order that words do/can appear in. In contrast to the amount of labeled data, unlabeled data (written text) is available in large amounts.

To cope with this problem, a lot of work (Mikolov et al., 2013; Pennington et al., 2014) focuses on transferring general natural language knowledge, obtained from unlabeled written text, into a format that is easy to use when approaching tasks where the data itself is not big enough to infer the required NLU. Most prominent, word-vectors, which represent each word as a fixed-size vector, helped improve the performance on a wide range of different domains and tasks. However, most of the past work focused on representations of single words, leaving the order of words and their meaning in different contexts out of scope.

Improving Language Understanding by Generative Pre-Training (Radford et al., 2018)

Radford et al., (2018) approached LM using an approach known as generative pre-training (GPT). Instead of creating a vector representation for each word, the authors provide a pre-trained model. GPT allows for discriminative fine-tuning towards the task at hand by adaptation of its architecture. It is based on a slightly modified version of the transformer (Vaswani et al., 2017) decoder architecture explained in Section 2.2.4. Essentially allowing the positional embeddings to be learned instead of having to rely on sinusoidal functions. In contrast to the previously discussed work, the main contribution is not the architecture but the way its pre-training and fine-tuning works. The proposed model is pre-trained, using the LM objectives, on 7000 books of different domains from the BooksCorpus (Zhu et al., 2015) containing approximately one billion tokens. To fine-tune the model towards different tasks instead of predicting the probability of words, the final layer is replaced by a single linear layer that projects the output to the required dimensionality. Thus, the architecture and all weights (except for the final layer) are fine-tuned for the task and its domain, while only the newly added layer is trained from scratch. Based on the task at hand, the input text is augmented through special tokens, which allow the model to learn structural relationships in the data, e.g., questions and answers.

While the model allows for arbitrary special tokens, these usually signify the start or end of input and delimiters between the start and end.

Language Models are Unsupervised Multitask Learners (Radford et al., 2019)

Based on their initial GPT idea, the authors also released a more recent version named GPT-2 (Radford et al., 2019) making minor adjustments to the architecture, but using a new dataset and drastically scaling the number of parameters the model has to learn. Instead of using the BooksCorpus as for GPT, the authors create a new corpus named WebText, including over 8 million documents and 40GB of text. The crawled web text introduces a greater number of domains and tasks that naturally appear in natural language, e.g., translation or summarization. The authors show the advantages of their approach, beating 7 SOTA results on LM datasets from different domains. The authors show that although the model was only trained on the newly proposed dataset and has never seen the task-specific target (zero-shot learning), GPT-2 has learned multiple tasks during pre-training, which can be accessed through natural language keywords, e.g., adding the TL;DR: token to the end of a sequence lets the model output a summary of that sequence. The results show that GPT-2 without fine-tuning yields a reasonable performance. However, it could not beat models that were fine-tuned or are specialized in a specific task. Thus to obtain the best results for a particular task and/or domain, GPT-2 should be fine-tuned as described in Radford et al., (2019). As GPT and GPT-2 mostly consist of the same architecture and GPT is the predecessor of GPT-2 in the following, we will only refer to GPT-2 for readability reasons.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2018)

Bidirectional Encoder Representations from Transformers (BERT) is another pre-trained language model similar to GPT-2 (Radford et al., 2019) that allows for an easy and efficient adaptation towards different NLP tasks through fine-tuning. In contrast to GPT-2, BERT is based on the encoder of the transformer (Vaswani et al., 2017) architecture, meaning that it is based on self-attention without masking future input. Thus, the most significant difference between BERT and GPT-2 is that the former is bidirectional while the latter is unidirectional, which we will explain later in this section. In addition, BERT is trained on the BooksCorpus (Zhu et al., 2015) and the English version of Wikipedia, which totals approximately 3.5 billion tokens. The authors also introduce Next Sentence Prediction (NSP) as an additional pre-training task, which they argue improves the performance on several

downstream tasks that require an analysis of relationships between two sentences. NSP is treated as a classification task where the model has to classify if a given sentence is the next sentence after the current one. To do this, the input is altered using "[CLS]" as a special token for classification at the beginning of input and "[SEP]" as a separator token to mark the input boundaries. In contrast to GPT-2, these tokens are used during fine-tuning and in pre-training, i.e., NSP, and can be used for transfer learning. BERT achieved SOTA results on eleven downstream tasks, e.g., question answering. Figure 2.8 shows an example input and how BERT computes the embedding used as an input to its transformer architecture. The *Final Embeddings* is based on *Position Embeddings*, representing the temporal structure in a text, *Segment Embeddings* which are used to signal to which part of an input the text belongs, and *Token Embeddings*, which cover the semantic meaning of input tokens.

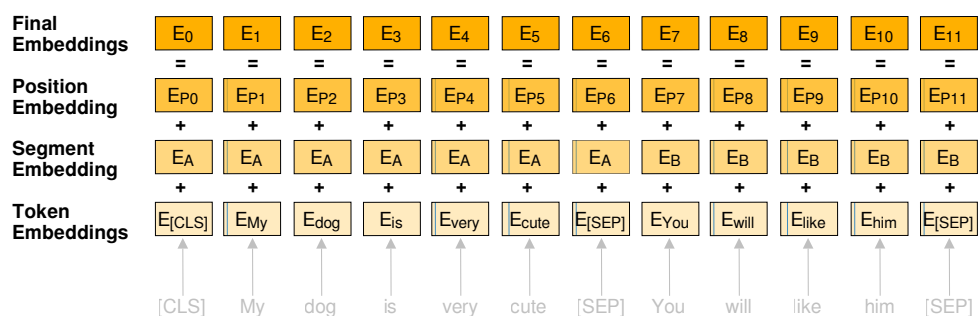


Figure 2.8.: Final Embeddings used as an input to BERTs transformer architecture are created as a sum of Token, Segment and Position Embeddings. Adapted from Devlin et al., (2018).

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension (Lewis et al., 2019)

The Bidirectional and Auto-Regressive Transformer (BART) is a Seq2Seq model developed for natural language text generation. It combines both BERT (Devlin et al., 2018) and GPT-2 (Radford et al., 2019) into a single architecture, with the former being the encoder and the latter being the decoder, respectively. Figure 2.9 shows an example of the pre-training strategies of GPT-2, BERT, and BART. This combination is very similar to the original transformer architecture (Vaswani et al., 2017), with minor changes in the architecture as introduced in (Devlin et al., 2018; Radford et al., 2018, 2019).

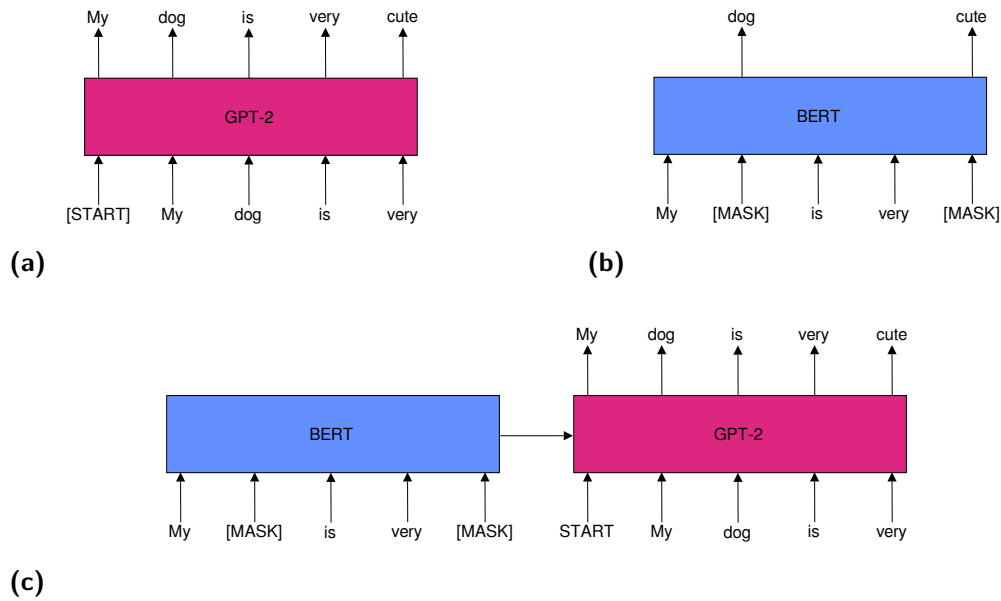


Figure 2.9.: Example input and output during pre-training (Language Modeling): (a) GPT-2, (b) BERT and (c) BART. Adapted from Lewis et al., (2019).

The authors show that this architecture allows for a new LM training procedure in which input text can be corrupted arbitrarily (as in BERT), which improves the generational performance of GPT-2. The different input transformations are the following:

- **Token Masking:** As introduced in BERT, some tokens are randomly masked (replaced) by a masking token.
- **Token Deletion:** Some tokens are randomly deleted from the input text.
- **Text Infilling:** Same as *Token Masking*, but instead of masking single tokens, multiple tokens that form a sequence are masked at the same time and replaced by a single masking token.
- **Sentence Permutation:** All sentences in the document are shuffled to create a different sentence order.
- **Document Rotation:** Randomly selecting a single token and rotate the input text such that it begins with this token.

BART achieves SOTA results in seven text generation tasks and performs exceptionally well in abstractive summarization tasks, making it an excellent choice for our work.

Autoregression, Autoencoding and Sequence-to-Sequence

There are three different categories that the models based on the transformer (Vaswani et al., 2017) architecture fall into. These models differ in the way they are pre-trained using the Language Modeling objective. First, Autoregressive models are trained on the LM tasks and correspond to the transformer architecture decoder. Autoregressive means that the model is unidirectional. Thus, during pre-training, it only looks at the left (previous) part of an input to predict the current word, while the rest of the input is masked out as it is the case for GPT-2 (Radford et al., 2019). Second, Autoencoding is linked to the transformer architecture encoder and describes its bidirectional pre-training objective. Unlike Autoregressive models, Autoencoders consider both the left (previous) and right (after) input to predict the current word as it is the case for BERT (Devlin et al., 2018). Figure 2.10 shows how the difference between Autoregressive and Autoencoding models based on their architecture. Instead of predicting the next word, Autoencoding models try to fix a corrupted version of the input by predicting the missing (masked) word. Finally, Sequence-to-Sequence models use both the encoder and the transformer architecture decoder, i.e., Lewis et al., 2019. During the encoder phase, the entire input is considered (bidirectional), while in the decoder phase, only the left (unidirectional) part, together with the representation created by the encoder, is used.

Both Autoregressive and Seq2Seq models are usually used for text generation tasks, while Autoencoders are used for downstream tasks, e.g., classification. Note, however, that all three can be fine-tuned to a wide range of different tasks.

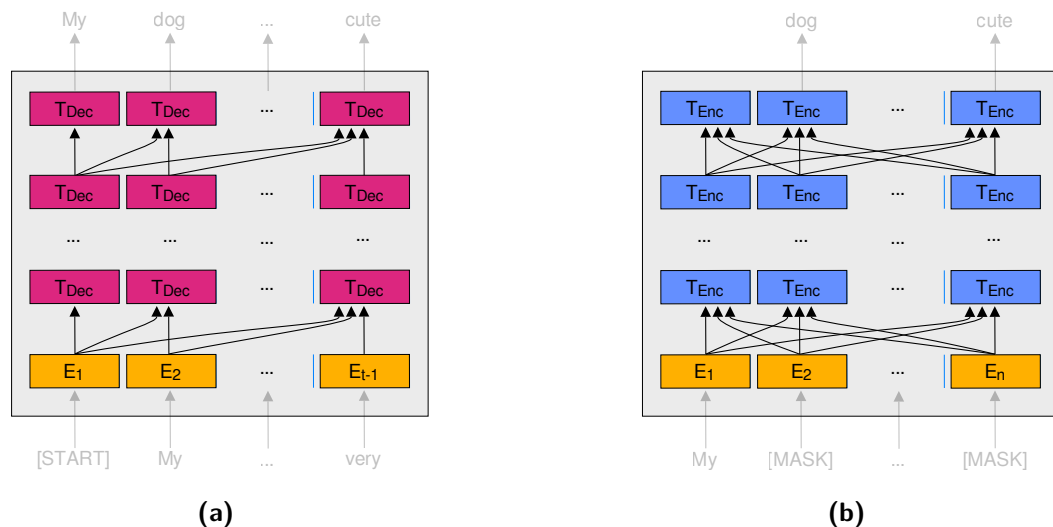


Figure 2.10.: (a) Autoregressive model architecture as in GPT-2 (b) Autoencoding model architecture as in BERT. Adapted from Devlin et al., (2018).

Metrics

To measure the generated conclusions' quality, it is important to choose a metric that measures the properties we want to be present in the generated text. To do this, we will first compare the nature of our task with already existing NLP tasks that require text generation and discuss which of the metrics used in these contexts might be useful for us.

Compared to the task of translation, in our case, the generation of conclusions is characterized by a single reference conclusion. For the second major task in the field of text generation, that is to summarize, we have to differentiate between extractive and abstractive text summarization assessment. In extractive summarization, the information from the text input is copied and fused into a summary. The resulted summary, therefore, contains many words that can directly be taken from the input. On the contrary, when creating a conclusion, the text is not directly copied from the premises. Finally, abstractive summarization requires the output to convey the meaning of the input text but can express the meaning by words different from those used in the input text. This task is at least in evaluation similar to ours, even if our task does not need a summary to compare to but a logical conclusion that can be inferred but not found in the input text.

Looking at these well-established tasks, we conclude that our task's evaluation is most similar to the evaluation of abstractive summarization, considering that we are looking to cover the meaning of the ground truth conclusion in our generated conclusion. In the following, we will briefly introduce the metrics we will use to evaluate our conclusion generation approach and explain why we included them.

Bilingual Evaluation Understudy (Papineni et al., 2002)

The Bilingual Evaluation Understudy (BLEU) is one of the most used metrics in text generation approaches. Initially, it was used to evaluate the quality of translations created by a machine to a set of human references. BLEU measures how many words (and/or n-grams) that appear in the candidate also appear in the set of references (Equation 2.26). To avoid counting matches that appear in more than one reference multiple times, it uses a clipped count.

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-gram} \in C} Count_{clip}(n\text{-gram})}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-gram}' \in C'} Count(n\text{-gram}')} \quad (2.26)$$

Thus BLEU is a precision based metric. It also involves a brevity penalty term BT to avoid candidates c to be much shorter than the references r (Equation 2.27) and

averages multiple n-gram sizes using uniform weights w_n . Equation 2.28 shows the complete BLEU formula. In all of our experiments we use a BLEU implementation⁶ which uses n-grams of size one ($N = 1$) and two ($N = 2$).

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \quad (2.27)$$

$$BLEU = BP * exp\left(\sum_{n=1}^N w_n * log(p_n)\right) \quad (2.28)$$

We will consider BLEU as our evaluation metric to measure the precision of our generated conclusions as it is an easy to interpret metric (range 0-1) that is well established in the literature and correlates well with human judgement. However, it is important to keep in mind that the task of conclusion generation is highly abstractive and BLEU does not account for the use of synonyms but requires the n-grams of the generated text to match those in the reference texts exactly.

Recall-Oriented Understudy for Gisting Evaluation (Lin, 2004)

The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is one of the most frequently used metrics for evaluating the quality of text summarization approaches. Unlike BLEU, it is recall oriented, i.e., it measures how many of the n-grams of the reference texts appear in the generated text (Equation 2.29). There are several different adaptations of ROUGE, which are used depending on the task at hand. The most common adaptations consider the longest common sequence of n-grams (ROUGE-L, ROUGE-W) or try to keep the order of words in the generated text (ROUGE-S, ROUGE-SU). Unlike BLEU, ROUGE prefers longer generated texts and does not apply any penalty to cope with this problem. However, this is not a problem for approaches that allow a controlled maximum number of n-grams, such as BART (Lewis et al., 2019). Different n-gram sizes are not averaged, but each is simply considered an independent measurement. Like BLEU, one of ROUGE's main problems is the absence of taking into account synonyms and, in general, texts that convey the same meaning but use different words/grams to express it. Note that the Google version of ROUGE that we use⁷ also uses stemming and text normalization.

$$ROUGE - N = \frac{\sum_{S \in ReferenceSummaries} \sum_{n \in S} Count_{match}(gram_n)}{\sum_{S \in ReferenceSummaries} \sum_{n \in S} Count(gram_n)} \quad (2.29)$$

⁶<https://github.com/mjpost/sacrebleu>

⁷<https://github.com/google-research/google-research/tree/master/rouge>

We will use ROUGE-1 and ROUGE-2, as commonly used in the literature, as our evaluation metric to measure the recall of our approach to conclusion generation because of its interpretability (range 0-1) and its correlation with human judgment.

Metric for Evaluation of Translation with Explicit ORDERing (Banerjee and Lavie, 2005)

The Metric for Evaluation of Translation with Explicit ORDERing (METEOR) is commonly used in machine translation and was developed to solve some of BLEU's problems. Unlike BLEU and ROUGE, METEOR takes recall and precision into account. However, it weights precision much higher (9x) than recall (Equation 2.30). In general, METEOR can be thought of as a penalized F1 score that favors precision over recall (Equation 2.32). In addition, METEOR uses stemming and synonym resolution to allow matches that do not cover exactly the same words/gram. METEOR uses a penalty term (Equation 2.31) to consider words/grams that occur only in the references and not in the candidate (and cannot be resolved as a synonym) and vice versa. Recall is defined as the ratio of unigrams in the references that can be mapped to a unigram in the candidate, and precision as the ratio of unigrams in the candidate that can be mapped to a unigram in the references. For all our experiments, we use the METEOR implementation that is available as part of the Natural Language Toolkit⁸ (NLTK).

$$F_{mean} = \frac{10PR}{R + 9P} \quad (2.30)$$

$$Penalty = 0.5 * \left(\frac{chunks}{unigrams_matches} \right)^3 \quad (2.31)$$

$$METEOR = F_{mean} * (1 - Penalty) \quad (2.32)$$

We will use METEOR as a metric, primarily to account for the use of synonyms, which is necessary due to the abstract nature of our task, but also to evaluate precision and recall in a single score.

⁸https://www.nltk.org/_modules/nltk/translate/meteor_score.html

BERTScore (Zhang et al., 2019)

BERTScore is one of the most up-to-date metrics that has been developed for evaluating the quality of a generated text. It is based on BERT’s contextual embeddings, which consider words in relation to their surrounding words/text, and is pre-trained on large amounts of data. BERTScore takes a reference x and a candidate \hat{x} and creates a contextual embedding for each word in both sentences. Afterward, the pairwise cosine similarities between all words in the candidate and the reference are calculated. The maximum similarity value for each word is then weighted by importance using the inverse document frequency (Equation 2.33) and combined into a single precision/recall and F1 score (Equations 2.34, 2.35 and 2.36).

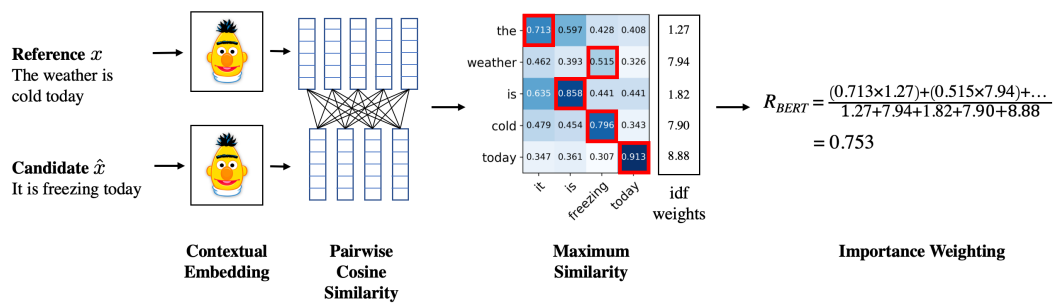


Figure 2.11.: Overview of BERTScore approach from Zhang et al., (2019)

BERTScore can assess reference-candidate pairs that express the same meaning but use different wording in contrast to BLEU, ROUGE, and METEOR. However, it is much less interpretable, as the exact influence of individual words and the creation of embeddings and the resulting similarity assessment is no longer a transparent process. Note that the Equations 2.34 and 2.35 are based on pre-normalized vectors (unit vectors) and therefore do not contain the denominator of the cosine similarity formula ($\frac{\mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\|\mathbf{x}_i\| \|\hat{\mathbf{x}}_j\|}$).

$$\text{idf}(w) = -\log \frac{1}{M} \sum_{i=1}^M \mathbb{I}[w \in x^{(i)}] \quad (2.33)$$

$$R_{BERT} = \frac{\sum_{x_i \in x} \text{idf}(x_i) \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{x_i \in x} \text{idf}(x_i)} \quad (2.34)$$

$$P_{BERT} = \frac{\sum_{\hat{x}_j \in \hat{x}} \text{idf}(\hat{x}_j) \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{\hat{x}_j \in \hat{x}} \text{idf}(\hat{x}_j)} \quad (2.35)$$

$$F_{BERT} = 2 \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \quad (2.36)$$

We will use BERTScore to evaluate our approaches, as this is the only metric that can fully account for the use of different ways to represent the same meaning.

2.4 Argumentation, Argument Mining, and Argument Quality

2.4.1 Argumentation

Even though argumentation is part of our everyday life, its structure and motivations often evolve naturally without clear rules to follow. But what is argumentation? Why do we argue? And what is a good argument? The answer to these questions are still not fully explored (Freeley and Steinberg, 2013) and may never be solved completely. Van Eemeren et al., (2004) define argumentation as "a verbal, social, and rational activity aimed at convincing a reasonable critic of the acceptability of a standpoint by putting forward a constellation of propositions justifying or refuting the proposition expressed in the standpoint." A more simplistic definition could be "The usage of arguments to persuade, agree, deliberate, or similar." ⁹. While holistic definitions of argumentation may be useful in some cases, they lack the precision and completeness necessary in others.

Bentahar et al., (2010) offer a more fine-grained look into argumentation by categorizing the past decades' research. The authors divide argumentation based on Monological, Dialogical and Rhetorical models which differ in structure, foundation, and linkage: (1) Monological models focus on the structure within single arguments, meaning they split an argument into components, often referred to as argumentative units, and analyze the relationships (linkage) between them. While several models represent these components and their relationships in different ways (Farley and K. Freeman, 1995; Reed and Walton, 2003; Toulmin, 2003), the main components of a single argument are some kind of premises and conclusions. Usually the /conclusion is supported by one or more premises (Walton et al., 2008) and states a stance on a controversial issue (Freeley and Steinberg, 2013). In contrast, premises shall provide reasons to prove conclusions (foundation). However, premises often remain implicit (Toulmin, 2003). That is, the author of an argument does not explicitly state all premises but leaves out those that he believes to be true in common sense. (2) In contrast, Dialogical models aim at the relationships between arguments (linkage) and how they are used to reason in an argumentation. They describe structures that appear in conversations between two or more participants that are mostly governed by rules that limit participants' possibilities to persuade others of their view. Dialogical models are based on the idea that most arguments are defeasible (Walton, 2005) by nature (foundation). (3) Rhetorical models are based on the audience's

⁹Following the slides of the "Computational Argumentation" course by Henning Wachsmuth at Paderborn University (<https://en.cs.uni-paderborn.de/de/css/teaching/courses/computational-argumentation-s19>).

perception of arguments (foundation) and describe the structure of arguments in terms of patterns, schemes, or strategies. Their main target is not to find the truth but to discover how arguments can be connected (linkage) to persuade the audience.

The categorization of Bentahar et al., (2010) shows that argumentation differs based on the circumstances it is used in as well as on different granularity levels. The levels of granularity sorted from low to high are (1) argumentative units, (2) arguments, (3) (Monological) argumentation; and (4) (Dialogical) debates Wachsmuth et al., 2017a. Our work will focus on the level of arguments and the argumentative units they consist of. However, as our source dataset (Chapter 3) contains essays, we will also briefly provide an overview of (Monological) argumentation models in the following. (Monological) argumentation models are often represented as graphs or graph-like structures. A graph usually contains the argumentative units as nodes connected by labeled or unlabeled edges that express the relations between nodes. Beardsley, (1950) and Thomas, (1981) define a total of five different types of structures that can be found in arguments, later further discussed by J. B. Freeman, (2011). Following the explanations and graphs of J. B. Freeman, (2011) and Stab and Gurevych, (2017a), Figure 2.12 shows the different types of argument structures.

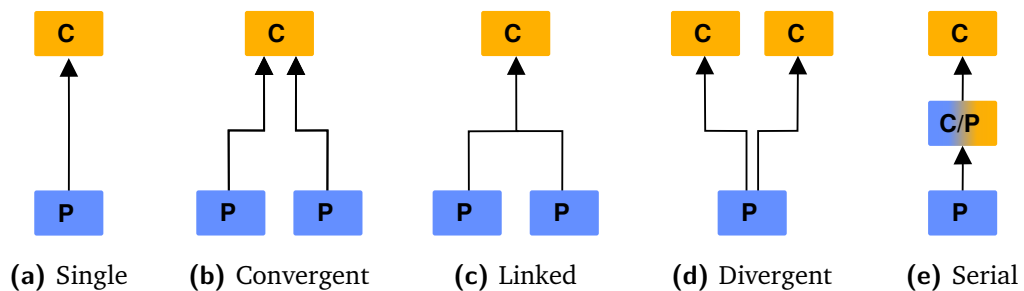


Figure 2.12.: The five different types of argument structures.

- **Single:** A conclusion supported by a single premise.
- **Convergent:** A conclusion supported by multiple independent premises.
- **Linked:** A conclusion supported by multiple premises that depend on each other.
- **Divergent:** Multiple conclusions supported by the same premise.
- **Serial:** A conclusion supported by a premise that simultaneously is the conclusion on another premise.

While Convergent and Linked arguments are hard to distinguish (Stab and Gurevych, 2017a), we still think that typing common structures within arguments is useful to

understand argument models in general. Toulmin, (2003) for example, focus on the use of unit roles, differentiating between facts, warrants, and backings as types of supportive premises, rebuttals as attacking premises, a qualifier which signals how strong the conclusion is and the conclusion itself. Facts are specific contextual information, while warrants are rules that support the conclusion considering the context given by the facts and backings that support the warrants. In contrast, J. B. Freeman, (2011) focus on a dialectical view of argumentation, where oppositions and propositions are fighting each other to establish the main conclusion's truth. Closest to our work is the model of Walton et al., (2008), which models argumentation as a conclusion supported/attacked by minor and major premises that belong to a certain type of argument, e.g., an argument from values. Minor premises provide specific information, while major premises link multiple premises together to generalize a rule. The concept is similar to facts and warrants of Toulmin, (2003) respectively. While these complex structures are understandable for humans, they are tough for computers to access and especially problematic to use as an input to machine learning algorithms. In our work, we will use arguments annotated similar to the approach of Walton et al., (2008) and transform them to fit our approaches. The data and our transformation procedure will be discussed in Chapter 3.

But what defines good argumentation in general? Wachsmuth et al., (2017a) survey existing theories and combine them into a single taxonomy of 15 dimensions (Figure 2.13).

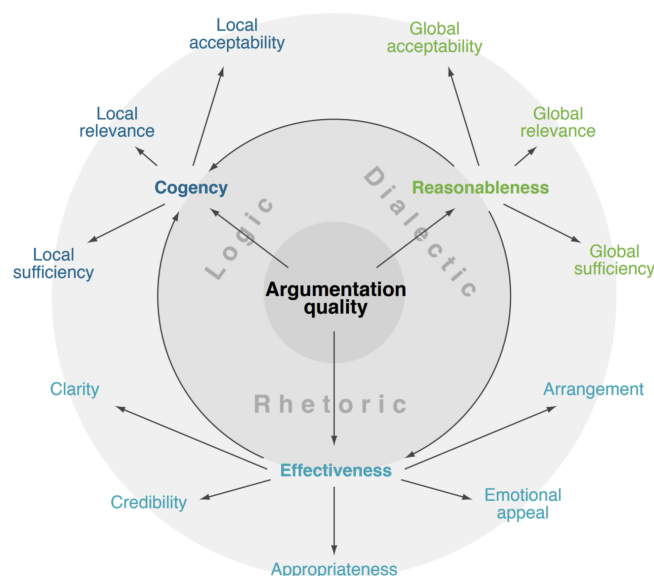


Figure 2.13.: Argument quality dimensions from Wachsmuth et al., (2017a).

The 15 dimensions are split into 3 head-categories and 12 sub-categories which hold a parent-child relationship.

- **Cogency** describes the logical aspect of an argument, previously referred to by Monological models (Bentahar et al., 2010). According to Wachsmuth et al., (2017a): "An argument is cogent if it has acceptable premises that are relevant to its conclusion and that are sufficient to draw the conclusion." Thus it has to fulfill the requirements of all three of its sub-categories: *Local Acceptability* as a measure of the truthfulness of the premises, *Local Relevance* as a measure of usefulness from premises to the conclusion; and *Local Sufficiency* to measure if the premises give enough support to rationally draw the conclusion.
- **Reasonableness** describes the dialectical aspect of argumentation, that is it "contributes to the issue's resolution in a sufficient way that is acceptable to the target audience" (Wachsmuth et al., 2017a). Its sub-categories are the same as in Cogency but redefined on the level of argumentation instead of a single argument. Thus *Global Acceptability* is a measure of the truthfulness of all arguments in the argumentation that also considers how arguments are presented. *Global Relevance* is a measure of the usefulness of the argumentation to resolve the target issue, and *Global Sufficiency*, is a measure of robustness to counter-arguments.
- **Effectiveness** describes the rhetorical aspect of an argumentation. As discussed previously, it tries to persuade the target audience to believe the author's stance on the issue. The sub-categories of Effectiveness are defined as: *Credibility*, as a measure of credence of the author and the way he argues, *Emotional Appeal* to create emotions which help to persuade, *Clarity* as a measure of correctness of the language used and to avoid ambiguities, unnecessary complexity as well as deviation from the issue, *Appropriateness* to measure if the language used fits the issue; and *Arrangement* to measure if the structure of the argumentation is correct.

Considering these quality dimensions, a "good" argumentation appears to be of a complex nature and requires many different considerations. To gain further insight into the various aspects of argumentation, we will next explain the field of *Computational Argumentation*. To avoid confusion of terms related to argument quality, we will follow the ideas and terms proposed by Wachsmuth et al., (2017a) throughout this work. In particular, we will use the definition of *Local Sufficiency* as it is the assessment task of this thesis.

2.4.2 Computational Argumentation

Today, with the recent advent of so-called fake news as well as filter bubbles, it has become more and more complex to form an opinion in a self-determined manner. *Computational Argumentation* is at the core of a wide range of applications that can help solving this problem e.g. fact-checking (Popat et al., 2017; Samadi et al., 2016) or argument search engines (Stab et al., 2018; Wachsmuth et al., 2017b). *Computational Argumentation* describes the analysis and synthesis of natural language argumentation based on data that is processed by a machine¹⁰. In contrast to the traditional argumentation research, *Computational Argumentation* usually processes larger amounts of data. While some resources exist on the web, e.g., idebate.org and debate.org, that provide data that is directly related to argumentation, most of the available text is either not argumentative or only partially argumentative. Thus to obtain large amounts of argumentative data, it is necessary to filter the available text for argumentative units. In addition, to evaluate the quality of argumentation, it is often necessary to understand its structure, i.e., find premises and conclusions and their relationships and the type of these relationships (support vs. attack). This process of data acquisition is called *Argument Mining* which is often necessary as a pre-step to perform other tasks, e.g., *Argument Quality Assessment* and *Argument Generation*.

Argument Mining

As discussed previously, arguments are often modeled as graphs containing conclusions and premises connected by an attack or support relation. Following the definitions of Wachsmuth et al., (2017a), to measure *Local Sufficiency*, it is necessary to extract both premises and conclusions and their relationships. While these are already annotated in the Argument Annotated Essays (AAE) corpora (Stab and Gurevych, 2014, 2017a) (see Chapter 3), to get more data or data from other domains, automatic annotation is often required. The approach to extract arguments and their structure from natural language text is often modeled as a pipeline consisting of multiple steps. A typical pipeline could look as follows: First, the argumentative units are extracted, that is, the text that fulfills an (until now) unknown argumentative function. Second, the argumentative units are classified into argumentative components, e.g., conclusions or premises. Third, the relations between these components are identified. Fourth, the argument structure is modeled using components and relations. Finally, the relationships are classified, e.g., support

¹⁰Following the slides of the "Computational Argumentation" course by Henning Wachsmuth at Paderborn University (<https://en.cs.uni-paderborn.de/de/css/teaching/courses/computational-argumentation-s19>).

or attack. Stab and Gurevych, (2017a) for example use an Integer Linear Programming (ILP) approach on the AAE-v2 corpus which contains 751 major conclusions, 1506 conclusions, 3832 premises. Their evaluation suggests that the extraction of argumentative and non-argumentative units and the extraction of components work well (F1-scores of .867 and .826). In comparison, the classification performance for relations and relation stance classification is much lower than the human baseline (F1-scores of .751 and .680), showing that both finding relationships and especially attacking relationships is a complex task (F1-scores of .585 and .413). In contrast, Wachsmuth et al., (2016) leverage the structure of an essay to obtain premises and conclusions that relate to each other and build a classifier to classify each sentence in an essay into one of four classes (thesis, conclusion, premise, none). However, in contrast to the approach of Stab and Gurevych, (2017a), the authors define each sentence to belong to exactly one of the four classes. They thus do not account for multiple argumentative discourse units within sentences. Compared to the approach of Stab and Gurevych, (2017a) the results of Wachsmuth et al., (2016) are slightly better with F1-scores of .745 and .726 correspondingly. Besides that, Al Khatib et al., (2016) show that cross-domain *Argument Mining* is an even more challenging task and that the domain of essays is dissimilar from other domains, e.g., online debates, which makes it difficult to generalize the performance of both approaches.

In this work, we will perform the assessment of an Arguments quality dimension, i.e., *Local Sufficiency* and exploit its definition in such a way that we redefine it to an Argument generation task, which is: An argument is locally sufficient if the conclusion can be generated based on its premises. As the code of Stab and Gurevych, (2017a) is not publicly available, we decided to use the approach of Wachsmuth et al., (2016) in our work to extend the amount of input data for our generation approach. However, as we found no improvement in a first test of conclusion generation using the approach of Wachsmuth et al., (2016) on the ICLE-v2 dataset (see Chapter 3), we will use only the annotated AAE-v2 dataset and do not mine argumentative discourse units in this work.

Argument Generation

The task of argument generation has gained traction in recent years. In contrast to our work, most of the approaches focus on generating entire arguments, e.g., premises and conclusions. Reisert et al., (2015) approach the task of argument generation utilizing rules to create arguments that follow the Toulmin model, thus extracting warrants, backing, and facts based on a conclusion as input. Similarly, Sato et al., (2015) tackle the same task without relying on a complex argumentative model using a neural network ranking approach. Le et al., (2018) focus on creating

an argumentative dialogue agent that can discuss topics with users. The authors use a siamese LSTM to obtain relevant responses to the input, which are then combined to generate the output using a Seq2Seq model. Wachsmuth et al., (2018) and El Baff et al., (2019) generate arguments that follow rhetorical strategies based on a combination of clustering, Language Modeling, and regression. Instead of focusing on planning based argument generation, Schiller et al., (2020) focus on aspect-based argument generation using the Controllable Language Model (CTRL) (Keskar et al., 2019) architecture. Arguments generation can thus be controlled based on topic, stance, and aspect. Similarly, Park et al., (2019) use a Seq2Seq to generate arguments from multiple perspectives; however, instead of controlling the aspects, the authors use latent mechanisms as an extra input to the decoder to create a more diverse output. Hua and Wang, (2018) and Hua et al., (2019) generate counter-arguments using an adapted version of the attention approach by (Bahdanau et al., 2014) and later an LSTM based planning approach, incorporating external knowledge from Wikipedia and news media. Hidey and McKeown, (2019) approach the task of counter-argument generation on the level of conclusions obtaining data from the fixed-that-for-you (FTFY) Reddit sub-forum. Closest to our approach, Wang and Ling, (2016) generate abstractive summaries using opinionated text from the iDebate portal and RottenTomatoes. The authors use an LSTM with an attention-based encoder and importance based sub-sampling to allow the encoder to learn which parts of the input text are essential to creating a summary. Considering both datasets, the iDebate dataset is closest to the data in our work. It contains a central conclusion that is supported by multiple premises. In addition, Alshomary et al., (2020b) tackled the problem of conclusion target inference using a Seq2Seq-based approach. The authors try to infer a conclusion target from a set of potential premise targets. To do this, they use a combination of a ranking based approach and a triplet neural network that embeds premises and conclusions in an embedding space to pick the conclusion target. For comparison premise targets, annotations were used as an additional input to the LSTM proposed by Wang and Ling, (2016) to create a conclusion. During training and test, premises and conclusions of the iDebate dataset were used. The authors showed that premise targets annotations can further improve the Seq2Seq approach by on their task.

In our work, as an alternative to the LSTM-based approach, a transformer-based approach will be used. In our case, we will consider BART (Lewis et al., 2019) due to its extensive pre-training, which can introduce not only knowledge about language which is useful in creating more fluent output, but also a latent representation of common knowledge.

Argument Quality Assessment

To conduct the planned work, it is necessary to define a few terms related to argument quality. Therefore the ideas proposed by Wachsmuth et al., (2017a) are used to define the terminology for quality dimensions. In particular, the definitions of *Local Sufficiency* will be used. *Local Sufficiency* belongs to Cogency, meaning the way it is trying to persuade is of logical nature. The applicability of these quality dimensions was tested on 304 arguments, taken from the UKPConvArgRank dataset (Habernal and Gurevych, 2016b), annotated by three annotators on a scale of 1-3. The authors evaluate the annotators' agreement using all annotators' full and majority agreement, which shows that the dimension is reasonably well defined to be assessed (at least for humans). Stab and Gurevych, (2017b) annotated 402 essays from the AAE-v2 corpus for the *Local Sufficiency* criterion as either sufficient or insufficient. During the evaluation of the annotation process, the authors found that 33.8% of all annotated arguments were insufficiently supported. To automatically assess the *Local Sufficiency* of an argument, an SVM on lexical, syntactic, and length-based features and a CNN with word vectors were used. The best results were obtained by a CNN with a macro F1-score of .827 and an Accuracy of .843. In addition, Wachsmuth et al., (2016) used various argument discourse units and structural features to improve the assessment of four argument quality dimensions. The results show that argumentative features obtained from *Argument Mining* can improve the performance in assessing the quality of arguments, especially if these are related to structure.

In our work, we will use the work of Stab and Gurevych, (2017b) as our main baseline as it is, to the best of our knowledge, the only work that assesses *Local Sufficiency*. In addition, we will also try to assess *Local Sufficiency* using argumentative features as proposed by Wachsmuth et al., (2016). In particular, the argumentative feature we are aiming for in our work is the generated conclusion.

Data for Local Sufficiency Assessment and Conclusion Generation

This chapter will describe and discuss the datasets used in this work: (1) We will describe existing datasets that fulfill the previously discussed criteria at least partially. (2) We will combine datasets to create our own dataset, which completely meets our requirements.

The datasets discussed in this chapter were chosen based on our approaches' requirements and are based on the general steps of our pipeline. Thus, we need data that fits our domain and is annotated towards *Local Sufficiency*. In addition, as we aim to explore the potential of argumentative features as well as to follow the *Local Sufficiency* definitions of Wachsmuth et al., (2017a) more strictly, we need data that is already segmented and classified, i.e., claims and premises, together with their relationship.

3.1 Existing Corpora

3.1.1 Persuasive Essays

International Corpus of Learner English v2 (Granger et al., 2009)

The International Corpus of Learner English v2 (ICLE-v2) corpus was published in 2009 as a collaborative result of several universities. It contains 6805 essays written by students from 16 countries who learn English and belong to higher intermediate or advanced learners. All essays were written as a response towards a given prompt, which belongs to one of the many topics covered in the corpus. We chose this corpus due to its overlap in the topic (student essays) with the other corpora to extend the number of argument annotated essays to improve conclusion generation. Since there are no annotations of conclusions and premises available for this corpus, and the automated mining of them using the approach of Wachsmuth et al., (2016) did not improve conclusion generation results in a first test, we decided against using this corpus. However, it could still be a valuable resource for future research in this

domain as *Argument Mining* approaches improve or manual annotations become available.

Argument Annotated Essays v2 (Stab and Gurevych, 2017a):

The Argument Annotated Essays v2 (AAE-v2) corpus was published in 2017 in the paper "Parsing Argumentation Structures in Persuasive Essays," using the same annotation scheme as its predecessor the AAE-v1 (Stab and Gurevych, 2014) corpus. The corpus contains annotations for 402 essays written by students taken from essayforum.com. In general, student essays usually consist of an introduction followed by one or more arguments/paragraphs and ends with a conclusion. A conclusion represents each argument/paragraph's central component, while a premise provides the reasons for the argument/paragraph. An essay can contain several main conclusions, usually located in its introduction and conclusion. Since the annotation scheme used does not explicitly model the relations between conclusions and major conclusions, it is not entirely clear whether a conclusion's stance can be translated as a support/attacking relation to all major conclusions. However, the authors implicitly suggest that the stance of all major conclusions in an essay are the same. Thus, it is reasonable to assume that a conclusion's relationship to all major conclusions in an essay is the same. In total, 751 major conclusions, 1506 conclusions, and 3832 premises were found. Premises and conclusions are connected by 219 attack and 3613 support relations.

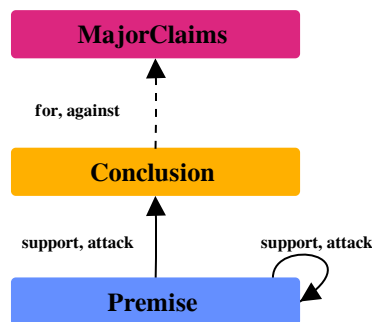


Figure 3.1.: Argument annotation scheme of the AAE-v2: Argument components are connected by support/attack relations (paragraph level) or a for/against relation (essay level).

The argumentation structure of each essay is modeled as a tree structure (Figure 3.1), where the first level (root) is a major conclusion, which represents the stance of an essay on its topic. Note that the stance towards the topic is not part of the annotated dataset. The second level of the tree is a conclusion that supports or attacks the corresponding major conclusion. Each other level of the tree contains premises that either support or attack a conclusion or, at deeper levels of the tree,

another premise. Thus both conclusions and premises have exactly one outgoing relation and no or several incoming relations. To distinguish the inner relations of arguments/paragraphs from relations, which cross these boundaries, relations from conclusions to major conclusions are labeled as for/against instead of support/attack.

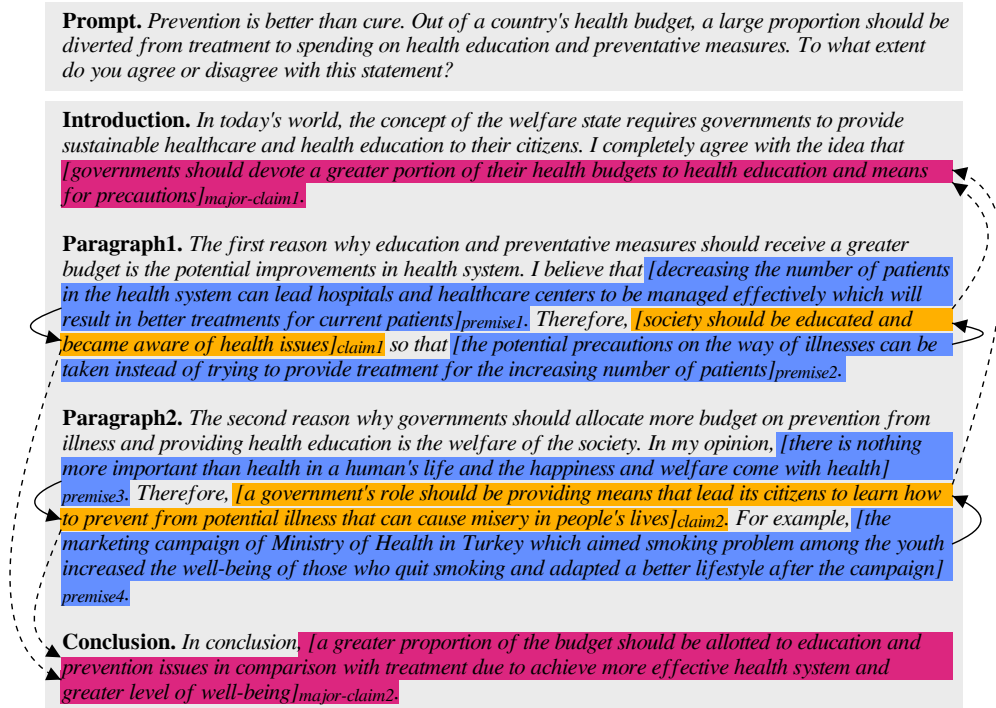


Figure 3.2.: Example annotations of Essay171: Major conclusions (pink) and conclusions (orange) are connected by for/against relations (dotted-arrows), while conclusions and premises (blue) are connected by support/attack relations (solid-arrows). Relation labels are omitted as they are not important for our approach.

Figure 3.2, shows an example essay from the AAE-v2 dataset and its annotations. The shown essay consists of an introduction, two arguments/paragraphs, and a conclusion. Both arguments/paragraphs contain two premises supporting a single conclusion. Each conclusion contains a stance towards the major conclusions in the introduction and in the conclusion. The resulting tree structure is shown in Figure 3.3.

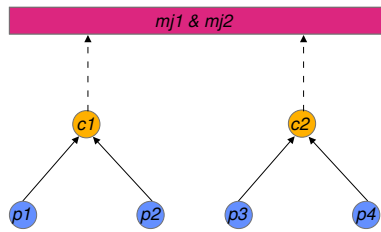


Figure 3.3.: Argumentation structure of Essay171: Major conclusions (pink) and conclusions (orange) are connected by for/against relations (dotted-arrows), while conclusions and premises (blue) are connected by support/attack relations (solid-arrows). Note, that for/against relations are not specific to a single major conclusion.

We chose the AAE-v2 corpus because it is the only corpus that contains argument component annotations for essays which are also annotated for *Local Sufficiency*.

3.1.2 Local Sufficiency Assessment

Insufficiently Supported Arguments in Argumentative Essays (Stab and Gurevych, 2017b):

The authors of the paper "Recognizing Insufficiently Supported Arguments in Argumentative Essays" used the 402 essays of the AAE-v2 corpus as a starting point to create binary *Local Sufficiency* annotations for each argument/paragraph of an essay. To define *Local Sufficiency*, the definitions of Johnson and Blair, (2006) were used. Stab and Gurevych, (2017b) define *Local Sufficiency* as:

"An argument complies with the sufficiency criterion if its premises provide enough evidence for accepting or rejecting the claim."

In total, the authors labeled 1029 arguments/paragraphs, of which 681 (66.2%) were considered sufficient and 348 (33.8%) were considered to be insufficient. On average, each argument/paragraph has a length of 4.5 sentences and contains 94.6 tokens. Note that all annotations are on the level of arguments/paragraphs and thus do not strictly follow the definition of Wachsmuth et al., (2017a), since some (4.3%) of the arguments/paragraphs contain several conclusions supported by premises. However, the authors argue that this abstraction has practical advantages, as it prevents possible error propagation in the identification of argumentative components and relationships.

Sufficient.	<p>Paragraph1. <i>The first reason why education and preventative measures should receive a greater budget is the potential improvements in health system. I believe that decreasing the number of patients in the health system can lead hospitals and healthcare centers to be managed effectively which will result in better treatments for current patients. Therefore, society should be educated and became aware of health issues so that the potential precautions on the way of illnesses can be taken instead of trying to provide treatment for the increasing number of patients.</i></p>
Insufficient.	<p>Paragraph2. <i>The second reason why governments should allocate more budget on prevention from illness and providing health education is the welfare of the society. In my opinion, there is nothing more important than health in a human's life and the happiness and welfare come with health. Therefore, a government's role should be providing means that lead its citizens to learn how to prevent from potential illness that can cause misery in people's live. For example, the marketing campaign of Ministry of Health in Turkey which aimed smoking problem among the youth increased the well-being of those who quit smoking and adapted a better lifestyle after the campaign.</i></p>

Figure 3.4.: Example *Local Sufficiency* annotations of Essay171: Paragraph 1 is considered sufficient and paragraph 2 is considered insufficient.

Figure 3.4 shows an exemplary annotation of the arguments/paragraphs of Essay171 (Figure 3.2). In this case, one of the arguments/paragraphs is marked as insufficient, which means that its premise does not provide sufficient support for accepting the conclusion. In contrast, the other is marked as sufficient, which means that its premises provide sufficient support for accepting the conclusion.

3.2 Corpus Transformation and Creation

3.2.1 Corpus Transformation

Since none of the previously discussed corpora is annotated for *Local Sufficiency* and argumentative units simultaneously, we have transformed the sufficiency corpus annotations of Stab and Gurevych, (2017b) accordingly with information from the AAE-v2 corpus of Stab and Gurevych, (2017a). However, arguments/paragraphs sometimes contain argumentative structures for which it is not trivial to assign a unique *Local Sufficiency* label to them. To deal with this problem, we will discuss the different types of argumentative structures we found in the datasets and how we resolved these in the following. Figure 3.5 gives an overview of the rules we have used to decide whether the argument/paragraph is kept or removed. The rules are sorted in order of application. Note that we have omitted major conclusions as we are only interested in *Local Sufficiency* labels on the argument/paragraph level. Furthermore, we do not distinguish between supporting and attacking relations. Thus a conclusion can include supporting and attacking relations in its set of premises at the same time.

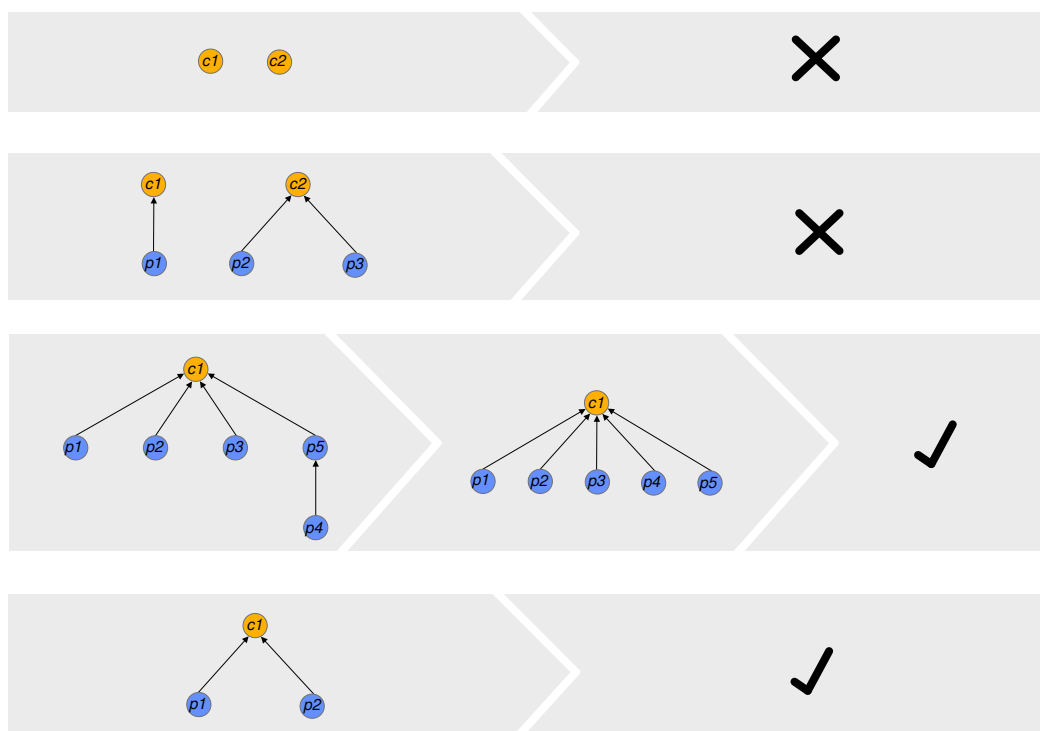


Figure 3.5.: Rules used to transform the AAE-v2 (Stab and Gurevych, 2017a) and insufficiency (Stab and Gurevych, 2017b) corpora into a single corpus containing conclusions and premises supporting these conclusions as well as a sufficiency label.

The most trivial structure of an argument/paragraph contains only one conclusion and one or more premises, which support this conclusion. In these cases the corresponding *Local Sufficiency* label of the argument can simply be used. Figure 3.6 shows an example essay containing a single conclusion supported by two premises. As this is the exact structure we aim for, the essay would be accepted for our dataset.

Besides, [nowadays technology is entering into our society really quick]*claim1* and [scientists develop robots, which help people cope with their problems or just invent coffee machines, engines with integrated computers and etc]*premise1*. For example, [before centeries, there were not washing machines, which clean your clothes, while a person do another job]*premise2*.

Figure 3.6.: Second paragraph of Essay335 and its argumentative structure as represented in the AAE-v2 (Stab and Gurevych, 2017a).

Second, some arguments/paragraphs contain only a single conclusion followed by one or more premises, but these premises could have incoming relations from other premises. If this is the case, we simply append all premises, which appear in the argumentation tree below the second level, directly to the conclusion. Thus the final tree structure is a simple conclusion followed by premises. Although we lose some structural information at this point, we assume that these premises are still important for drawing the conclusion and thus decided to keep them. Figure 3.7 shows an example essay containing a conclusion supported by five premises. One of these premises (Premise5) is supported by another premise, which is then attached directly to the conclusion, while the relationship to the parent premise is removed. After solving this problem, we end up with a single conclusion supported by five premises and thus will be accepted.

Secondly, [it is crucial to keep one's identity]*claim1* for [they need a connection back to their country as well as teach their children their value of origin]*premise1*. For instance, [children immigrated to a new country will face social troubles in school with new friends]*premise2*. [In this new environment, parent should find friends coming from their same country so that they can socialize in a very familiar manner as feeling being home]*premise3*. [Fail to create this familiarity makes them felt isolated, in the extreme can lead to social disorder like autism]*premise4*. Hence, it is clear that [keeping the cultural traditions in the destination countries is tremendous important]*premise5*.

Figure 3.7.: Second paragraph of Essay002 and its argumentative structure as represented in the AAE-v2 (Stab and Gurevych, 2017a).

Third, we remove all arguments/paragraphs containing more than one conclusion. The reason for this is that it is not clear whether the *Local Sufficiency* label is attached to one of the conclusions or to both. Figure 3.8 shows an example essay containing two conclusions, one of which is supported by a single premise and the other by two premises. It is therefore unclear whether the corresponding *Local Sufficiency* label, applies to both conclusions or only to one of them.

On the other hand, [the significance of competition is that how to become more excellence to gain the victory]_{premise1}. Hence it is always said that [competition makes the society more effective]_{claim1}. However, [when we consider about the question that how to win the game, we always find that we need the cooperation]_{premise2}. The greater our goal is, the more competition we need. [Take Olympic games which is a form of competition for instance, it is hard to imagine how an athlete could win the game without the training of his or her coach, and the help of other professional staffs such as the people who take care of his diet, and those who are in charge of the medical care]_{premise3}. The winner is the athlete but the success belongs to the whole team. Therefore [without the cooperation, there would be no victory of competition]_{claim2}.

Figure 3.8.: Second paragraph of Essay001 and its argumentative structure as represented in the AAE-v2 (Stab and Gurevych, 2017a).

Finally, some arguments/paragraphs contain only one conclusion, without any premises associated with the conclusion. If this appears, we cannot assign a *Local Sufficiency* label, as our approaches need at least one premise as input to generate a conclusion as output. Figure 3.9 shows an example essay that falls into this category. Interestingly, the paragraph is originally labeled as sufficient. This paragraph also violates our second rule because it contains several conclusions and cannot be solved by rule three.

Firstly, [connecting people by email is easy and fast]_{claim1}. In addition, [World Wide Web offers humanity to access to information, which they want to know for less than 10 seconds]_{claim2}. These are two of the benefits, why IT is useful.

Figure 3.9.: First paragraph of Essay335 and its argumentative structure as represented in the AAE-v2 (Stab and Gurevych, 2017a).

In total, we removed 47 (4.6%) paragraphs/arguments from the original corpus so that 982 conclusions and their premises are included in the end. The final corpus contains 647 (65.9%) sufficiently and 335 (34.1%) insufficiently supported conclusions. Each document in our corpus has exactly one conclusion and an average of 3.6 premises, resulting in an average of 81.5 characters per document. Compared to the original sufficiency data set, the class distribution has changed by 0.3% and the number of sentences by 0.1, while the average number of tokens has been reduced by 13.1.

Approaches and Implementation

In this chapter, we will discuss the experimental setup of our work. (1) We will explain our approach to directly assess *Local Sufficiency* and explain our *Model Setup* plus the *Data Pre-processing* and the *Training Procedure* we have used to train our models. (2) Similarly, we will describe how we obtained our conclusion generation baselines and our final fine-tuned models, which we will use for the indirect *Local Sufficiency* assessment approach. (3) Finally, we will explain how we used the conclusions generated as an argumentative feature to augment the *Local Sufficiency* assessment approach established at the beginning of this chapter.

4.1 Direct Local Sufficiency Assessment

This section will explain our approach to directly assess the *Local Sufficiency* of an argument. That is, we use the conclusion and its corresponding premises as an input to a BERT (Devlin et al., 2018) model to output a binary *Local Sufficiency* score that indicates whether the source argument is sufficient or insufficient.

Model Setup

Our data transformation procedure (see Chapter 3) led to the removal of 4.6% of the data, which could affect the models' performance, e.g., by removing difficult instances. Similarly, removing the non-argumentative text from arguments keeping only conclusions and premises could also influence our results. To rule out these problems, we train and evaluate our models and the CNN of Stab and Gurevych, (2017b) on the original dataset, our new version where we removed some instances, and the final version without non-argumentative text. For each of the three settings, we created a baseline using the original code used in Stab and Gurevych, (2017b) which we received from the authors to compare our approach against.

We chose BERT as a transformer-based model to predict the *Local Sufficiency* of an argument for our approaches. BERT's classification procedure is quite simple in our case and mostly follows the ideas proposed by the original authors (Devlin et al., 2018). First, remove the Language Modeling head of the original model, that is, the layers that predict the input's masked words. Second, add a linear layer on top of

BERT that projects the final layer embeddings of the "[CLS]" token to the desired output dimensionality, which in our case is one, as the *Local Sufficiency* annotations are binary. We build upon the Huggingface transformers implementation¹ of BERT for all our experiments and adjust it according as previously discussed.

Data Pre-processing

Considering that a single instance of input data consists of a set of premises and a conclusion, there is no structural relationship in the order in which they appear. However, BERT (Devlin et al., 2018) is pre-trained on natural language text and thus requires its input to be natural language text as well. To cope with this problem, we transformed all premises and the conclusion to appear like regular sentences and ordered them based on their original order in the argument, beginning with the conclusion. Thus we obtain a single natural language text sequence. To transform the conclusion and its premises into regular sentences, we uppercased the first letter and added a period at the end of each of them. Our input data is a single sentence conclusion and a joined sequence of single sentence premises. This structure allows two different types of input for BERT, based on its input structure discussed in Chapter 2 and especially Figure 2.8: First, join the conclusion and the sequence of premises to form a single sequence input and set input type tokens, which are responsible for the Segment Embeddings, to be the same for every input. Second, treat the conclusion and the sequence of premises as two different inputs, that is, joining them together to form a single sequence, but separate them using a "[SEP]" token and adjust the input type tokens accordingly. For the CNN approach of Stab and Gurevych, (2017b) we only considered the single sequence approach as it is difficult to tell the model which part of the input is a premise or a conclusion without changing the architecture.

Training Procedure

As we use a large pre-trained Language Model, i.e., BERT (Devlin et al., 2018), that takes a long time to train, we could not assess the *Local Sufficiency* of an argument following the original 20 times 5-fold cross-validation setting. Instead, we only used the first two of the 20 5-fold cross-validation setups proposed by Stab and Gurevych, (2017b) yielding a total of 10 different test folds. Every split ensures that conclusion and premise tuples from one essay are not split between training, validation, and test data. This avoids possible data leakage that could artificially improve the final evaluation scores. To ensure our changes to the cross-validation setup were also

¹<https://github.com/huggingface/transformers>

not affecting the results, we again repeated the CNN’s training and evaluation by Stab and Gurevych, (2017b) accordingly. For each cross-validation fold, we use 70% training, 10% validation, and 20% test data.

During our work, we found it difficult to successfully optimize BERT using Mean-Squared-Error (MSE) or Cross-Entropy loss functions, as both of them did not seem to align with our target metric (macro F1-score) very well. We found the work of Puthiya Parambath et al., (2014) and Eban et al., (2017) to be useful in this regard, as well as their application and discussion in practice^{2,3}. The authors propose to directly optimize machine learning models on the F1-score. Instead of interpreting a single binary value, we allow the model to output probabilities. If the model predicts 0.2 as a label and the ground truth is 1, the loss is calculated as 0.2 true positive and 0.8 false negative.

Similar to Stab and Gurevych, (2017b), we also allow our model to adjust hyperparameters between folds. To do this, we use the hyperparameter optimization framework optuna⁴ to optimize the dropout percentage of the final layer, batch size, and learning rate. Hyperparameter optimization frameworks automate and optimize the hyperparameter selection, defining a search space that is efficiently explored based on the model’s feedback at the end of the training procedure. In our case, this is the macro F1-score of the validation data obtained at the end of the training. We run 10 trials for each fold to find a dropout rate between 0.0 and 0.5, a batch size between 2 and 32, and a learning rate between 1e-6 and 5e-5. The boundaries are chosen based on sequence classification tasks that BERT was already fine-tuned on (e.g., SQUAD, MNLI). We fixed the number of epochs to 3 per fold because we could not find any improvements afterward and used a cosine learning rate scheduler without warm-up to reduce the learning rate during training as the model proceeds to see more examples. Finally, we selected the epoch for each trial out of the three, which performed best on the validation data.

4.2 Conclusion Generation

This section will explain our approach to generate conclusions. That is, we use the premises of a conclusion as an input to a BART (Lewis et al., 2019) model to output a conclusion that is close/matches the ground truth conclusion.

²<https://www.kaggle.com/rejpalcz/best-loss-function-for-f1-score-metric>

³<https://towardsdatascience.com/the-unknown-benefits-of-using-a-soft-f1-loss-in-classification-systems-753902c0105d>

⁴<https://optuna.org/>

Model Setup

In contrast to our *Local Sufficiency* assessment approach, we will use BART (Lewis et al., 2019) to generate conclusions. This is because BERT (Devlin et al., 2018) is not particularly useful for text generation tasks due to its pre-training procedure. In contrast, BART's performance on abstractive summarization could be useful for the task of conclusion generation. As explained in Chapter 2 BART is a sequence-to-sequence architecture consisting of BERT as an encoder and GPT-2 as its decoder. To the best of our knowledge, conclusion generation is a novel task previously only explored by (Wang and Ling, 2016). Thus, in addition to pre-trained BART, we investigate the transfer learning opportunities from tasks that BART was already fine-tuned on and seemed related to our task. In particular, we found BART pre-trained on the CNN-DailyMail (Nallapati et al., 2016) and the XSum (Narayan et al., 2018) dataset to be promising. Considering extractive (CNN-DailyMail) and abstractive (XSum) summarization tasks could improve results on the generation and provide valuable information to the conclusion generation task. We also consider three different versions of our dataset for conclusion generation: (1) the full dataset, (2) only the instances in our dataset that are labeled as *Local Sufficient*, (3) only the instances that are labeled as not *Local Sufficient*. These scenarios help us to rule out potential problems in generating conclusions, as conclusions from instances that are labeled as not *Local Sufficient*, could be much harder or even impossible to generate based on their premises and thus introduce a lot of noise to the data, which may affect the model effectiveness.

Data Pre-processing

Our model's input is a set of premises joined into a single sequence as in our BERT for *Local Sufficiency* assessment approach. Similarly, we also used the conclusion in its regular sentence setting as previously described.

Training Procedure

To generate conclusions that we can use as an input for our *Local Sufficiency* assessment approach and avoid overfitting, we must generate a conclusion for every conclusion premise tuple in our dataset from the test set. Thus to create a full set of conclusions, we must train multiple models to combine all test set predictions. Consequently, to keep our results consistent, we followed the same two 5-fold cross-validation setup as previously described for our *Local Sufficiency* assessment

approach. Each 5-fold cross-validation iteration yields a full set of generated conclusions for our dataset, of which we pick one (the first) for our downstream task. However, note that our automatic evaluation is still averaged over both 5-fold cross-validation iterations, thus being a total of 10 different settings.

We use the Cross-Entropy loss for fine-tuning BART, as is the usual loss function used in text generation.

Otherwise, we follow the same hyperparameter optimization procedure as before but with different parameters and ranges to tune. As the BART model is a lot bigger in terms of parameters, we adjusted the batch size between 4 and 8 and the learning rate between $5e-6$ and $5e-5$. As we did not change the model architecture, there was no dropout value to tune anymore. Considering that the search space is much smaller now, we also adjusted the number of trials to 5 per fold. We fixed the number of epochs to 3 and used cosine learning rate scheduling to reduce the learning rate the further the training advances. As our batch size was much smaller this time, we also used 50 warm-up steps to stabilize the training. Warm-up steps linearly increase the learning rate for some batches, in the beginning, i.e., 50 until we reach our initial learning rate from which we decay using cosine learning rate scheduling. This procedure can help stabilize the training with small batch sizes as picking a "bad" batch in the beginning, does not influence the model too much, which could lead to worse results in the end. To obtain the generated conclusion for our test set, we use a beam size of 4, max length of 70 tokens (derived based on our longest ground truth sequence +20%). Instead of generating a single conclusion greedily word by word, beam search explores multiple possible words at each level, i.e., 4 that are the most likely, and creates conclusions based on these. Afterward, the conclusion with the highest overall probability is chosen as the output. Finally, we considered the epoch out of the three, which performs best on the validation data.

4.3 Indirect Local Sufficiency Assessment

This section will explain our approach to indirectly assess the *Local Sufficiency* of an argument. That is, we use the conclusion and its corresponding premises as well as the generated conclusion as an input to a BERT (Devlin et al., 2018) model to output a binary *Local Sufficiency* score that indicates whether the source argument is sufficient or insufficient.

The indirect *Local Sufficiency* assessment approach uses the same model as its direct counterpart. In addition, the procedure in which we obtain hyperparameters stays the same, as well as the cross-validation setup. Thus we will skip repeating the explanation at this point and only address the difference in *Data Pre-processing*, which explains how we used generated conclusions to augment the *Local Sufficiency* assessment and what baselines we compare our approach to.

Data Pre-processing

After generating conclusions, we use the *Local Sufficiency* assessment approach previously described in this chapter in combination with our generated conclusions. Thus we change the input data in the following ways:

1. Use only the premises as input.
2. Use only the ground truth conclusion as input.
3. Use only the generated conclusion as input.
4. Use both conclusions as input.
5. Use the generated conclusion together with the premises as input.
6. Use the generated conclusion, the ground truth conclusion, and the premises as input.

Note that we used the single sequence setup, as we could find no improvements (see Chapter 5) in adjusting the input type tokens in our direct *Local Sufficiency* assessment approach.

Experiments and Evaluation

This chapter will evaluate the approaches described in Chapter 4. (1) We will evaluate our approach to directly assess *Local Sufficiency* and analyze the impact of our data set changes on model performance. (2) We will evaluate our approach to conclusion generation both automatically and in a manual annotation study. (3) We will evaluate our indirect *Local Sufficiency* assessment approach.

This chapter’s results are calculated based on 10 models (two times 5-fold cross-validation). Therefore, for each measurement, we report the average metrics and the corresponding standard deviation. To ensure the statistical significance of our approaches, we use the Wilcoxon signed-rank test, since the number of observations in our case is small (10) and the difference between the measurements does not follow a normal distribution, as is required for other statistical tests, such as the paired student t-test. Note that all results discussed in this chapter, unless stated differently, are significant concerning the Wilcoxon signed-rank test with a p-value of 0.05.

5.1 Direct Local Sufficiency Assessment

Based on the approaches described in chapter 4.1, we compare our approach with the human baseline and the former SOTA CNN of Stab and Gurevych, (2017b). Note, however, that the human baseline created by Stab and Gurevych, (2017b) is based on a subset of 433 arguments annotated by three annotators. The human baseline scores are based on pairwise comparisons of the three annotators. As previously described, we have modified the data set to better fit the *Local Sufficiency* definition of Wachsmuth et al., (2017a). To avoid that changing the data affects our evaluation values, we investigate three different settings in our experiments: (1) the original data set without changes, which we refer to as *Full*, (2) a subset of the original data set where we have removed 4.6% of the data referred to as *Sub*; and (3) the subset of (2), but without non-argumentative text (only conclusions and premises), which is referred to as *C&P*.

For evaluation purposes, we consider the macro F1 score to be our main metric, as it considers the balance of recall and precision and the imbalance of the data set. However, we also report recall and precision, and overall accuracy independently

in Table 5.1. We also perform a more fine-grained assessment of our approaches for each of the datasets. That is, we divide the evaluation for each dataset into sufficient and insufficient instances and compare F1, recall, and precision scores in a class-specific manner. Table 5.2, Table 5.3, and Table 5.4 show the results for the *Full*, *Sub*, and *C&P* dataset. Finally, we examine how the changes we made to the data (Chapter 3) affect our results and the reasons for the changes.

		Accuracy	Macro F1	Macro Prec.	Macro Rec.
Full	Human [†]	.911 ± .022	.883 ± .029	.873 ± .042	.903 ± .020
	CNN	.846 ± .022	.831 ± .023	.830 ± .021	.832 ± .028
	BERT	.868 ± .017	.854 ± .020**	.856 ± .046	.860 ± .066
Sub	Human [‡]	.912 ± .021	.886 ± .027	.876 ± .041	.906 ± .017
	CNN	.836 ± .024	.820 ± .023	.820 ± .021	.820 ± .029
	BERT	.874 ± .026	.854 ± .037**	.865 ± .047	.860 ± .061
C&P	CNN	.805 ± .040	.778 ± .037	.807 ± .049	.752 ± .031
	BERT	.837 ± .034	.811 ± .041*	.831 ± .049	.800 ± .056
	BERT _[SEP]	.829 ± .023	.801 ± .030	.822 ± .043	.792 ± .056

Table 5.1.: Results of the direct assessment approach based on two 5-fold cross-validations compared to a human upper bound. ** and * mark significance over the CNN baseline of Stab and Gurevych, (2017a) with a p-value of 0.01 and 0.05 respectively. ‡ and † are obtained on a subset of 432 and 410 arguments respectively.

Table 5.1 shows that our approach significantly outperforms the previous SOTA of Stab and Gurevych, (2017b) on all of our three datasets and metrics. Specifically, we outperform the previous SOTA on the *Full* dataset on average by a macro F1 score of $+.023$. Compared to the human baseline on the same dataset, our model performs slightly worse with an average difference of $-.029$. Thus, our model is approximately at a level of 96.3% of human performance. Considering the *Sub* dataset, we find the same pattern with our model being a solid step in between the CNN of Stab and Gurevych, (2017b) and the human baseline, with macro F1 score differences of $+.034$ and $-.032$ respectively. Overall, we could not find a significant difference between the performances on the *Full* and the *Sub* dataset, even though the average macro F1 score of the CNN is slightly lower ($-.011$). Based on these results we conclude, that the removal of 4.6% of the data during dataset transformation (Chapter 3.2) does not impact the learnability of the dataset. Finally, the evaluation of the *C&P* dataset in terms of macro F1 score, holds the same relationships between our BERT model and the reference CNN, as on the *Full* and *Sub* datasets, with our model outperforming the CNN by a macro F1 score of $+.045$. However, the overall scores are significantly lower compared to the other datasets for our best model as well as the CNN with a reduction in macro F1 score of $-.043$ and $-.042$ respectively compared to the *Sub* dataset. We will discuss possible reasons for this effect later in this Section. As discussed in Chapter 4.1 we also tried to separate the two inputs (a conclusion and its premises) using a separator token in between as well as adjusting

the token type ids correspondingly. However, considering the macro F1 score, we found its performance to be significantly worse compared to the single sequence BERT model ($-.022$). We think that this is due to the similarity between conclusions and premises. Both are statements, but the former needs support to be seen as true, while the latter is usually seen as true without any support. Supposedly, the models are not able to learn the difference between them correctly. Contrasting the macro F1 scores with accuracy’s, which favor sufficient arguments due to the dataset imbalance (65.9% vs. 34.1%), the results are analogous. Thus showing that in both cases (balanced and imbalanced), our model outperforms the previous SOTA CNN. Analysing precision and recall, our model as well as the CNN of Stab and Gurevych, (2017b) are balanced on the *Full* and *Sub* dataset, with our model scoring .856; .860 and .865; .860 respectively, beating the CNN on both dataset as also indicated by the macro F1 score. Thus, our model weighs the retrieval of sufficient and insufficient arguments and the correctness of each class’s retrieved arguments equally. In contrast to the automated approaches, on average, humans have slightly higher recall than precision ($+.030$), thus slightly favoring finding all the sufficient and insufficient arguments over confusing both classes. However, we could not find statistical significance for this result, which may be caused by the small number of annotators. While our BERT model outperforms the CNN and BERT_[SEP] on the *C&P* dataset, their overall scores drop as indicated by the macro F1 score. Interestingly, for both models, the recall drops much more than the precision ($-.068$ and $-.060$ vs. $-.024$ and $-.034$), but still not to the point of significance.

		F1	Precision	Recall
Suff.	Human†	.827 ± .043	.787 ± .099	.884 ± .049
	CNN	.882 ± .019	.892 ± .033	.875 ± .041
	BERT	.899 ± .015*	.915 ± .037	.886 ± .047
Insuff.	Human†	.940 ± .035	.959 ± .023	.923 ± .038
	CNN	.775 ± .034	.768 ± .046	.788 ± .075
	BERT	.810 ± .029**	.797 ± .055	.833 ± .086

Table 5.2.: Results of the direct assessment approach based on two 5-fold cross-validations compared to a human upper bound on the *Full* dataset. ** marks significance over the CNN baseline of Stab and Gurevych, (2017b) with a p-value of 0.01. † is obtained on a subset of 432 arguments.

On the *Full* dataset (Table 5.2), in terms of F1 score, our model yields minor improvements in recognizing sufficient arguments (.007) but major improvements for those arguments that are labeled as insufficient (.035). Similar to the CNN of Stab and Gurevych, (2017b), our model is better at recognizing sufficient arguments (.899) than recognizing insufficient arguments (.810). This result is especially interesting in the context that humans hold an inverted behavior (.827 vs. .940). In total, although our model outperforms the human baseline on sufficient arguments by an F1 score of .072 overall it is slightly less accurate ($-.023$) as humans outperform our model by an F1 score of .130. Considering precision and recall for both classes on average

we found both automated approaches to favor precision over recall for sufficient arguments while the classification on insufficient arguments is the other way around. However, the statistical test shows no significance in this observation. Humans, in comparison, show a completely inverted behavior thus favoring recall for sufficient and precision for insufficient arguments.

		F1	Precision	Recall
Suff.	Human \ddagger	.831 \pm .041	.792 \pm .098	.887 \pm .045
	CNN	.874 \pm .022	.881 \pm .035	.870 \pm .043
	BERT	.896 \pm .015**	.906 \pm .035	.906 \pm .042
Insuff.	Human \ddagger	.941 \pm .014	.960 \pm .021	.924 \pm .037
	CNN	.761 \pm .036	.759 \pm .045	.770 \pm .075
	BERT	.815 \pm .038	.824 \pm .058	.815 \pm .080

Table 5.3.: Results of the direct assessment approach based on two 5-fold cross-validations compared to a human upper bound on the *Sub* dataset. ** and * mark significance over the CNN baseline of Stab and Gurevych, (2017b) with a p-value of 0.01 and 0.05 respectively. \ddagger is obtained on a subset of 410 arguments.

We also find the same behavior for the *Sub* dataset, supporting our hypothesis that deleting 4.6% of the data does not affect the general performance.

		F1	Precision	Recall
Suff.	CNN	.874 \pm .022	.809 \pm .039	.919 \pm .052
	BERT	.881 \pm .024	.849 \pm .034	.917 \pm .033
	BERT _[SEP]	.875 \pm .017	.845 \pm .031	.909 \pm .034
Insuff.	CNN	.673 \pm .035	.806 \pm .082	.584 \pm .059
	BERT	.740 \pm .058**	.813 \pm .064	.683 \pm .079
	BERT _[SEP]	.728 \pm .045	.798 \pm .056	.675 \pm .078

Table 5.4.: Results of the direct assessment approach based on two 5-fold cross-validations compared to a human upper bound on the *Full* dataset. ** marks significance over the CNN baseline of Stab and Gurevych, (2017b) with a p-value of 0.01.

Finally, on the *C&P* dataset the class-specific F1 scores, mostly affect the capability of the models to recognize insufficient arguments, decreasing it by $-.088$ for the CNN $-.075$ for our best BERT model compared to the *Sub* dataset. In contrast, the performance of sufficient arguments does not change significantly. We also find that there is no significant drop in precision between sufficient and insufficient arguments, as was the case for the other datasets, although on average their performance is still lower for our model ($-.36$). This is especially the case for the CNN of Stab and Gurevych, (2017b) which on the *Sub* dataset has a significantly higher precision on sufficient arguments compared to insufficient arguments. On the *C&P* in contrast, we could not find this significant difference anymore. Also, both our model as well as the reference CNN significantly drop precision performance on the sufficient instances while changing the dataset without losing significant performance on insufficient arguments. In terms of recall, we found the inverted behavior thus both our model

as well as the reference CNN perform significantly worse on insufficient arguments, induced by a significant drop in performance for this class transitioning to the *C&P* dataset.

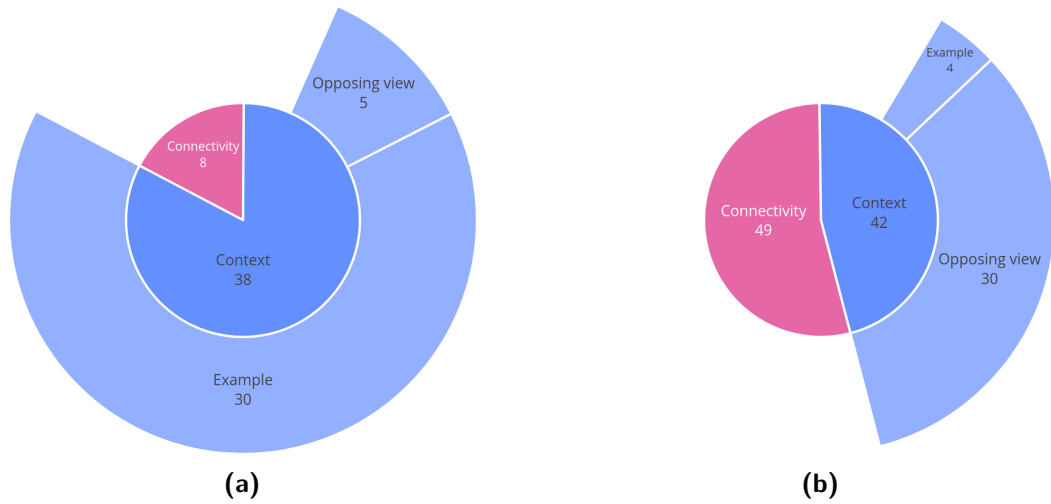


Figure 5.1.: (a) Number of arguments that were classified incorrectly before removing non-argumentative text and correctly after the removal and (b) the number of arguments that were classified correctly before removing non-argumentative text and incorrectly after the removal.

As we have found a significant drop in performance between the *Sub* and *C&P*, we investigated the corpus creation procedure to better understand potential problems during dataset conversion. As we trained two 5-fold cross-validations, we have two predictions for each argument. To find the most impactful cases, we specifically looked at those arguments that were predicted to be sufficient by both models before removing non-argumentative text and predicted as insufficient by both models afterward and vice versa. Figure 5.1 shows how removing non-argumentative text from the arguments changed the models predictions. In a manual investigation, we found 137 cases in which our models consistently changed their predictions, with 46 of them changing from incorrect to correct and 91 from correct to incorrect. Based on the non-argumentative text which we removed, we clustered these arguments into *Connectivity* and *Context* classes. *Connectivity* describes arguments where only text is removed that indicates typical argumentative structures, which are single words, e.g., "First," "Second," "Thus"; or short sequences, e.g., "Another argument is." *Context* in contrast includes all arguments where either Context in the literal sense is removed or markers that indicate the quality or stance of an argument. As we found the latter to be common cases, we further divided the *Context* class into *Opposing view* and *Example* classes. *Opposing view* describes *Context* that indicates that one or more of the premises convey a different stance on the issue than the rest of the premises (attacking relationship), important sequences we found are, for example, "One may argue ... but ...", "... however ..." or "I disagree with ... because ...". On the other hand, *Example* describes arguments where the text was removed

that indicates that one or more premises are examples, usually introduced as "For example" or "For instance." Comparing the change in predictions, we found that dropping *Connectivity* is more likely to decrease classification performance (8 vs. 49) while dropping *Context* can be equally beneficial as hurting to our models. However, a deeper investigation shows that removing *Example* instances, often increases the performance (30 vs. 4) while removing *Opposing view* leads to a decreases (30 vs. 5).

5.2 Conclusion Generation

In addition to our direct assessment approach, we also study the task of conclusion generation and its potential to enhance the *Local Sufficiency* assessment. This section will evaluate our conclusion generation approaches both in an automated way using the metrics discussed in Chapter 2 and based on a manual annotation study involving five annotators. Based on our experimental setting (see Chapter 4), we use the pre-trained BART (Lewis et al., 2019) model as an initial starting point. In the following, we will refer to the pre-trained BART model as *BART-large*. In addition, we also explore the potential of transferring knowledge from text summarization, i.e., news article summarization. Thus, we consider two versions of *BART-large* that were pre-trained on news summarization datasets. We refer to these models as *BART-CNN* and *BART-XSum*, with the former being trained for extractive and the latter for abstractive text summarization. In our evaluation, we differentiate three versions of our *C&P* dataset to ensure that our conclusion generation model can learn its task successfully: (1) The full dataset referred to as *Full*, (2) only sufficient arguments from the dataset referred to as *Suff.* and; (3) only insufficient arguments from the dataset referred to as *Insuff.*

As for the direct approach, all of the results in this chapter are calculated based on 10 models (two times 5-fold cross-validation). Therefore, for each measurement, we report the average metrics. The full table with standard deviations can be found in the appendix for readability reasons. To ensure our approaches' statistical significance, we use the Wilcoxon signed-rank test for the same reasons as mentioned previously. The results discussed in this chapter, unless stated differently, are significant concerning the Wilcoxon signed-rank test with a p-value of 0.05.

5.2.1 Automatic Evaluation

As all the models we use in this section are already pre-trained, we divided the evaluation into two steps. First, we will evaluate the initial performance of the models on our dataset without any training. This is to create baselines and explore which pre-trained model is the best starting point for the fine-tuning process. Second, we fine-tune the pre-trained model on our dataset and discuss the results concerning the baselines we have created.

To evaluate our conclusion generation approaches, we use several different text generation metrics discussed in Chapter 2. We use BLEU-1 (B1) and BLEU-2 (B2) to measure exact word precision, ROUGE-1 (R1) and ROUGE-2 (R2) to measure exact word recall, and METEOR (M) as a combination of them. As measuring exact word overlap, does not account for potential matches in meaning, we also use BertScore to obtain equivalent precision (BS-P), recall (BS-R), and combinational (BS-F1)

evaluation scores. The complete tables with standard deviations can be found in the appendix.

		B1	B2	R1	R2	M	BS-P	BS-R	BS-F1	Len.
—	BART-large	11.5	1.71	15.9	2.74	15.8	7.22	24.1	15.6	52.8
	BART-CNN	13.7	1.99	18.4	3.15	16.8	13.6	28.0	20.7 [†]	43.8
	BART-XSum	19.7	2.44	20.0	3.32	14.2	19.9	26.8	23.4^{†‡}	25.9
Full	BART-large	26.5	4.06	21.5	4.44	13.4	32.9	28.4	30.6 [‡]	17.2
	BART-CNN	24.7	3.63	20.2	4.04	12.8	30.8	27.9	29.4	17.9
	BART-XSum	27.9	4.21	21.1	4.23	12.4	34.5	28.0	31.2[‡]	15.8
Suff.	BART-large	25.5	3.84	20.7	4.09	12.9	32.2	28.2	30.2 [‡]	17.6
	BART-CNN	24.2	3.44	20.5	3.92	13.2	29.6	27.9	28.8	18.6
	BART-XSum	25.9	3.86	21.0	4.19	13.1	32.7	28.6	30.7[‡]	17.2

Table 5.5.: Results of the conclusion generation approach based on two five-fold cross validation **with** and **without** fine-tuning. † and ‡ mark significance over the *BART-large* and *BART-CNN* models.

Considering Table 5.5, without any fine-tuning on our dataset, the *BART-XSum* performs best with an BS-F1 score of +2.7 and +7.8 compared to the *BART-CNN* and *BART-large* respectively. However, its BERT based recall score (BS-R) score is much higher (+6.9) compared to the corresponding precision score (BS-P), thus generated conclusions favor containment of words from the ground truth conclusion over the inclusion of irrelevant words. The same behavior can also be found for *BART-CNN* and *BART-large*, with +14.4 and + 16.9 respectively. Even though *BART-XSum* performs best on the BS-F1 score, *BART-CNN* has higher BS-R (+1.2) and thus lower BS-P (−6.3). Considering that *BART-CNN* is trained on extractive summarization that uses sentences from the input and fuses them into a summary, this is expected as extractive summarization usually creates longer summaries compared to abstractive summarization and thus contains more irrelevant words because conclusions are often short sentences. As *BART-large* just generates text that the model believes follows after the input text, it has the lowest BS-P score (7.22). Note that we have limited the number of generated tokens to 70 (based on ground truth conclusions) to avoid artificial performance decreases due to generated text that is too long. In general, as none of the models was trained on our dataset, the average length of the summaries varies heavily (52.8 vs. 43.8 vs. 25.9) while the average ground truth conclusion length is 19.3. In general, longer conclusions potentially increase recall oriented metrics, while shorter conclusions increase precision-oriented metrics. However, considering that conclusions generated by *BART-XSum* are on average −17.9 tokens shorter compared to those generated by *BART-CNN* and −26.9 tokens shorter compared to those generated by *BART-large* and that the increase of BS-P is much higher compared to the decrease in BS-R, the task of abstractive summarization does still seem to be closest to our task of conclusion generation. In terms of the traditional metrics that require exact token matches, we found similar behavior with *BART-XSum* being the best model for B1, B2, R1, and R2, even outperforming

BART-CNN on the recall metrics R1 and R2. However, R1 is a bit lower than BS-R on average (-8.2 , -9.6 , and -6.8). We also found that B2 and R2, which both assess the overlap of longer sequences (in this case 2), are very low for all three models with a maximum of 2.44 and 3.32 for the *BART-XSum* respectively. Finally, the M score favors *BART-CNN* over *BART-XSum* by a score of $+2.6$, although the former beating the latter on B1 and R1, which is probably caused by either the synonym resolving or word ordering assessment of M.

		B1	B2	R1	R2	M	BS-P	BS-R	BS-F1	Len.
BART-large	Suff.	11.9	1.79	16.0	2.75	15.5	8.62	24.0	16.2[†]	50.3
	Insuff.	10.7	1.59	15.8	2.72	16.3	4.49	24.4	14.3	57.6
BART-CNN	Suff.	14.4	2.06	18.9	3.20	16.8	14.9	28.1	21.5[†]	41.7
	Insuff.	12.5	1.89	17.5	3.01	17.0	11.1	27.7	19.3	47.7
BART-XSum	Suff.	19.9	2.55	20.0	3.32	14.3	19.7	26.4	23.1	26.2
	Insuff.	19.4	2.20	19.8	3.29	14.0	20.4	27.5	24.0	25.3

Table 5.6.: Results of the conclusion generation approach based on two five-fold cross validation **without** fine-tuning split by sufficient and insufficient arguments. [†] marks significance improvement over the the other class.

Table 5.6 shows the performance of all models without fine-tuning split into sufficient and insufficient instances. *BART-large* and *BART-CNN* both perform better on sufficient arguments with BS-F1 scores of $+1.9$ and $+2.2$. In contrast *BART-XSum* performs better on insufficient arguments ($+0.9$) on average but without significance. Considering metrics that require exact matching of words (B1, B2, R1 and R2), we find that for all of our models the performance on sufficient instances is slightly better compared to insufficient arguments. In contrast analysing BERTScore precision and recall, we find that in terms of BS-P *BART-large* and *BART-CNN* perform significantly better on sufficient arguments ($+4.13$ and $+3.8$) while there is no significant difference in BS-R for all of our models. The higher BS-F1 score of *BART-large* and *BART-CNN* is thus the result of better precision on sufficient arguments. Finally, the results indicate that it is slightly easier to generate conclusions of sufficient arguments, which is in so far expected as some conclusions of insufficient arguments may suffer from not correctly adhering to the topic at hand. However, the results of the *BART-XSum* model which is the best model without finetuning, suggest that once a model reaches a certain performance this difference becomes negligible.

To investigate both hypothesis, we finetuned each of the three models on the full dataset, as well as on only sufficient arguments (see Table 5.5). *BART-XSum* and *BART-large* outperform *BART-CNN* on both datasets with an BS-F1 score of $+1.8$ and $+1.2$ on the *Full* dataset and $+1.9$ and $+1.4$ on the *Suff.* dataset. Overall, based on the *Full* dataset, the finetuned models outperform the pre-trained model baselines by $+15.0$ for *BART-large*, $+8.7$ for *BART-CNN* and $+7.3$ for *BART-XSum* and likewise for the *Suff.* dataset. This shows that the task of conclusion generation

can be learned to at least a certain degree. We also found that after finetuning, all models perform roughly the same, that is *BART-large* which was previously the weakest model gained the most performance, *BART-CNN* previously second weakest the second most performance and *BART-XSum* previously the best gained the least performance. Consequently, even though abstractive summarization seems to be the closest task compared to conclusion generation, our dataset does not profit from transferring knowledge between both tasks. However, as *BART-CNN* has significantly lower BS-F1 scores compared to *BART-large* and *BART-XSum*, extractive summarization seems to be the least related task to conclusion generation and if used as a starting point decreases its overall performance compared to the initial pre-trained *BART-large*. Although, finetuning increased the performance on all of our metrics, longer sequence metrics which measure exact word matches (B2 and R2) are still very low. Most of the improvements we see are precision based metrics, thus focused on avoiding irrelevant words in the generated conclusion. This is to a degree related to the much shorter length of generated conclusions. However, as the recall oriented metrics also increase in performance, the model also successfully generates relevant text. After finetuning we could find no significant difference between training the model on the *Full* dataset and training it on the *Suff.* dataset, although on average the metrics of the full dataset are slightly higher with an BS-F1 score of +0.4, +0.6 and +0.5 respectively.

		B1	B2	R1	R2	M	BS-P	BS-R	BS-F1	Len.
BART-large	Suff.	26.6	4.01	21.4	4.32	13.3	32.5	27.9	30.3	17.3
	Insuff.	26.1	4.01	21.5	4.63	13.5	33.5	29.2	31.4†	17.1
BART-CNN	Suff.	24.8	3.65	20.0	3.99	12.6	30.8	27.5	29.2	17.9
	Insuff.	24.5	3.59	20.6	4.07	13.1	31.0	28.6	29.8	18.1
BART-XSum	Suff.	28.4	4.24	21.2	4.24	12.3	34.6	27.6	31.1	15.8
	Insuff.	27.0	4.13	20.9	4.16	12.6	34.2	28.7	31.5	15.9

Table 5.7.: Results of the conclusion generation approach trained on the *Full* dataset based on two five-fold cross validation **with** fine-tuning split by sufficient and insufficient arguments. † marks significance improvement over the the other class.

Evaluating the performance on sufficient and insufficient arguments from models trained on the *Full* dataset (Table 5.7), we found that the average performance of generated conclusions for insufficient arguments is better compared to their sufficient counterparts, with BS-F1 differences of +1.1, +0.6 and +0.4 for the *BART-large*, *BART-CNN* and *BART-XSum* model respectively. However only the *BART-large* based model performs significantly better on insufficient arguments.

		B1	B2	R1	R2	M	BS-P	BS-R	BS-F1	Len.
BART-large	Suff.	25.5	3.81	20.6	3.97	12.9	31.8	27.8	29.9	17.7
	Insuff.	25.4	3.91	21.0	4.27	13.0	32.9	28.9	31.0	17.3
BART-CNN	Suff.	24.8	3.59	20.8	3.99	13.3	30.1	27.9	29.1†	18.5
	Insuff.	23.1	3.16	20.0	3.68	13.0	28.6	27.8	28.7	18.9
BART-XSum	Suff.	26.3	4.01	21.2	4.14	13.2	32.7	28.5	30.6	17.3
	Insuff.	25.1	3.57	20.6	4.23	13.0	32.7	28.8	30.8	17.1

Table 5.8.: Results of the conclusion generation approach trained on the *Suff.* dataset based on two five-fold cross validation **with** fine-tuning split by sufficient and insufficient arguments. † marks significance improvement over the the other class.

In contrast, analysing the performance on sufficient and insufficient arguments from models trained on the *Suff.* dataset (Table 5.7), we found that only the *BART-CNN* model performs significantly better on sufficient arguments with an BS-F1 difference of +0.4. For *BART-large* and *BART-XSum*, we found no significant difference in the performance on sufficient and insufficient arguments. Thus training a model on only sufficient arguments does not change the model to improve its performance both on sufficient and insufficient arguments. This observation is interesting as it helps us to draw conclusions about the dataset of Stab and Gurevych, (2017a): First, the quality of the data seems to be high in so far as both sufficient and insufficient arguments contain premises that fit the topic of the conclusion. Second, it shows that the difference between both classes seems to be small, as removing insufficient arguments from training does not bias the models to favor sufficient arguments.

5.2.2 Manual Evaluation

The metrics that we have used to automatically evaluate our conclusion generation approach are not perfectly suitable for conclusion generation. They either expect a single correct result from a pool of very similar results (as in summarization) or multiple references (as in translation). The task of conclusion generation, in contrast, allows for multiple dissimilar correct conclusions, e.g., having a different target created from a single set of premises. Since our dataset has only a single reference conclusion, we decided to conduct a manual annotation study to assess our models' general conclusion generation ability compared to the human ground truth and to find the best model for our indirect *Local Sufficiency* assessment approach. In the appendix we have included ten sets of premises together with conclusions generated by our models and the human reference.

Premises:

- Students have tendency to shun difficulties.
- They might avoid tough courses, but instead attend mainly mickey mouse courses and enjoy themselves too much.
- This might result in limited choices in the future.

Please rank the following claims based on their fit to the premises above:

	1 (Most likely)	2	3	4 (Least likely)
Students have tendency to shun difficulties and enjoy themselves too much.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Schools should be responsible for their future and protect them.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Students should be aware of the consequences of their choices.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Studying at an online university might be a waste of time and money.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Zurück Weiter Seite 14 von 51

Figure 5.2.: Example ranking task from our annotation study: annotators are asked to rank the four conclusions (lower part) and their likelihood to be drawn, based on the premises (upper part) on a scale from 1 (most likely) to 4 (least likely)

We designed the study as a ranking task where we chose 100 arguments from our dataset at random such that 50 of the arguments are labeled as sufficient and the other 50 as insufficient. For each argument, we used the generated conclusions

from the first 5-fold cross-validation for each of our finetuned models (*BART-large*, *BART-CNN*, and *BART-XSum*) together with the ground truth conclusion written by a human. Note that we have used the models trained on the *Full* dataset, as we found no difference between these models' performance and the model trained on the *Suff.* dataset in our automatic evaluation. We presented the four conclusions with their premises to five annotators with different academic backgrounds (economics, computer science, health/medicine). We asked each annotator to solve the following task for each of the 100 arguments:

"Rank the conclusions that you believe are most likely/least likely to be made based on a set of premises."

Figure 5.2 shows an example annotation task for a single argument taken from the study created with Google Forms¹. In addition, we also provided the following annotation guidelines:

- You must give exactly one answer per row and exactly one answer per column for each form you will see.
- You should assume that all the premises given are true. Therefore, you should apply the beliefs of the authors to create the ranking.
- Sometimes, the conclusions can refer to different topics. If this is the case, you are invited to deduce the most probable/most unlikely topic yourself.
- If you come across very similar conclusions, you are supposed to choose the one that is better expressed.
- Finally, the conclusions and premises you will see in this study are all taken from student essays and may therefore contain language errors that you can simply ignore. However, if you do not understand the premises or conclusions, you can skip the question.

To evaluate the inter-annotator agreement, we used the rank correlation coefficient Kendall's τ , which is a non-parametric significance test between two features on an at least ordinal scale. Table 5.9 (a) shows the pairwise Inter-Annotator Agreement (IAA) and (b) Inter-Rank Agreement (IRA). If one of the annotators decided to skip an argument, we did not consider it in our evaluation. Using this procedure, the maximum number of skipped arguments of two annotators we found was 10, leaving 90 arguments for evaluation, which should still be a representative quantity

¹<https://www.google.com/forms/about/>

for agreement calculation. We also tested all of our agreement scores using a one-sample Student’s t-test to ensure a significant difference from zero.

(a) IAA Full							(b) IRA Full					
	1	2	3	4	5	Avg.		1	2	3	4	Avg.
1	–	.20	.28	.22	.27	.24	1	–	.06	.25	.48	.26
2	.20	–	.14	.23	.23	.20	2	.06	–	.09	.39	.18
3	.28	.14	–	.25	.39	.27	3	.25	.09	–	.18	.18
4	.22	.23	.25	–	.32	.26	4	.48	.39	.18	–	.35
5	.27	.23	.39	.32	–	.30	Avg.	.26	.18	.18	.35	.24
Avg.	.24	.20	.27	.26	.30	.25						

Table 5.9.: Pairwise Kendall’s τ (a) of the Inter-Annotator Agreement (IAA) between the five annotators of our conclusion generation study and (b) of the Inter-Rank Agreement (IRA) based on the IAA agreement scores.

Considering the IAA scores from Table 5.9a, we found an average agreement score between all annotators of .25 which we consider as slight agreement. We found the highest agreement between Annotator-3 and Annotator-5 with a τ value of .39, considered as moderate agreement. In addition, Annotator-5 is also the annotator with the highest average agreement score (.30). As some of our annotators reported difficulties ranking conclusions that are very similar in their semantic meaning or the fit to the premises, especially for Rank-2 and Rank-3, but also for other ranks if multiple conclusion do not fit the premises, we also evaluated the Inter-Rank Agreement (IRA), which is the agreement between annotators in distinguishing the different ranks from one another. The results in Table 5.9b, show an average agreement score between all ranks of .24 which is very similar to the IAA score. Thus there is only slight agreement in overall rank discrimination. In contrast, we found moderate agreement discriminating Rank-1 and Rank-2 from Rank-4 with Kendall’s τ scores of .48 and .39 respectively, which shows that annotators are able to agree on the worst conclusion given. Consequently Rank-4 also has the highest average agreement (.35) of all ranks. Consistent with the statement of our annotators, the agreement between Rank-2 and Rank-3 is very low (.09). Interestingly the agreement on the best (Rank-1) and second best (Rank-2) conclusion holds an even lower agreement score (.06) while Rank-1 and Rank-3 can be differentiated to some degree (.25). This also explains the moderate agreement of Rank-2 and Rank-4 which is the results of a higher confusion between Rank-1 and Rank-2 (.06) compared to Rank-3 and Rank-4 (.18). In summary our annotators best agree on which is the worst conclusion while agreement on the best conclusion is mostly split between two of the remaining conclusions. Therefore, in order to choose the best model for the indirect *Local Sufficiency* assessment, we will consider the IRA as an additional factor.

(a) IAA Suff.							(b) IAA Insuff.						
	1	2	3	4	5	Avg.		1	2	3	4	5	Avg.
1	–	.27	.36	.24	.30	.29	1	–	.14	.20	.20	.25	.20
2	.27	–	.21	.25	.32	.26	2	.14	–	.06	.22	.14	.14
3	.36	.21	–	.35	.47	.35	3	.20	.06	–	.16	.32	.18
4	.24	.25	.35	–	.37	.30	4	.20	.22	.16	–	.26	.21
5	.30	.32	.47	.37	–	.37	5	.25	.14	.32	.26	–	.24
Avg.	.29	.26	.35	.30	.37	.31	Avg.	.20	.14	.18	.21	.24	.19

Table 5.10.: Pairwise Kendall’s τ of the Inter-Annotator Agreement (IAA) between the five annotators of our conclusion generation study, split into (a) sufficient and (b) insufficient arguments.

In addition to the IAA Full as well as the IRA, we analysed IAA split into arguments sufficient and insufficient (Table 5.10). We found that annotators agree more often on sufficient arguments compared to their insufficient counter parts with average Kendall’s τ scores of .31 and .19 respectively. Although, we can not fully explain this behavior, we hypothesize that sufficient arguments either hold premises that support choosing the premise order e.g. by narrowing down the conclusion target and thus limiting the space of possible suitable conclusions which helps to differentiate the ranks; or the generated conclusions are more diverse between the models. The best agreement between pairs of annotators are equally distributed between both classes with Annotator-3 and Annotator-5 agreeing the most with moderate scores for sufficient (.47) and for insufficient (.32) arguments.

To compare the conclusion generation models to the human baseline and find the best model to use for our indirect *Local Sufficiency* assessment approach, we used the majority rank of our five annotators for each argument and the corresponding models. In case we found ties between our most frequent ranks, we resolved this issue as follows: In our case, due to the number of annotators, a tie is only possible between two ranks with two votes each. Thus, to resolve this issue, we use the remaining annotation to decide between the tied ranks. If the remaining annotation is closer to the pair of better ranks, we choose this rank and vice versa for pairs of worse ranks. If the remaining annotation is in between the tied ranks, we simply select the best among them. Figure 5.3 shows the majority based ranking split by our three conclusion generation model in terms of their total count. For each of the models, we have 100 ranks, which correspond to the 100 conclusions generated by the models/written by the human. Our findings show that *BART-large* performs best in terms of receiving the most (32) Rank-1 votes in total while having a similar number of Rank-4 votes (28) as *BART-CNN* (28) and the Human baseline (29). In contrast to the *BART-large* model, the Human baseline and the *BART-CNN* model have a higher number of Rank-4 conclusions compared to Rank-1 conclusions (+5). Considering Rank-2 and Rank-3, the Human Baseline seems to be slightly better compared to *BART-CNN* as the former has marginally less (–2) Rank-3 conclusions.

BART-XSum is more challenging to evaluate. It is similar to *BART-large* in having a positive difference of Rank-1 and Rank-4 conclusions (+3), but it also has the most Rank-3 conclusions (37) by far. The overall average ranks also reflects the model ranking with values of 2.32, 2.5, 2.53, and 2.55 for *BART-large*, Human baseline, *BART-CNN*, and *BART-XSum*, respectively. In summary, the most interesting finding of our study is that although our annotators have a moderate agreement over ranking the conclusions, the human baseline’s performance is very close to all of our models’ performance and overall slightly lower compared to our best model. Consequently, we argue that our conclusion generation models create relevant and convincing conclusions for a set of premises. However, as the conclusions for a single set of premises often differ in their target, we hypothesize given a set of premises, it is possible to draw multiple conclusions with different targets that fulfill the *Local Sufficiency* quality criterion. Our study also shows that, given a set of premises, it is unclear which conclusion target is correct. As the *BART-large* model performs best in our manual annotation study and is on par with other models in our automatic evaluation, we will use it for our indirect *Local Sufficiency* assessment approaches.

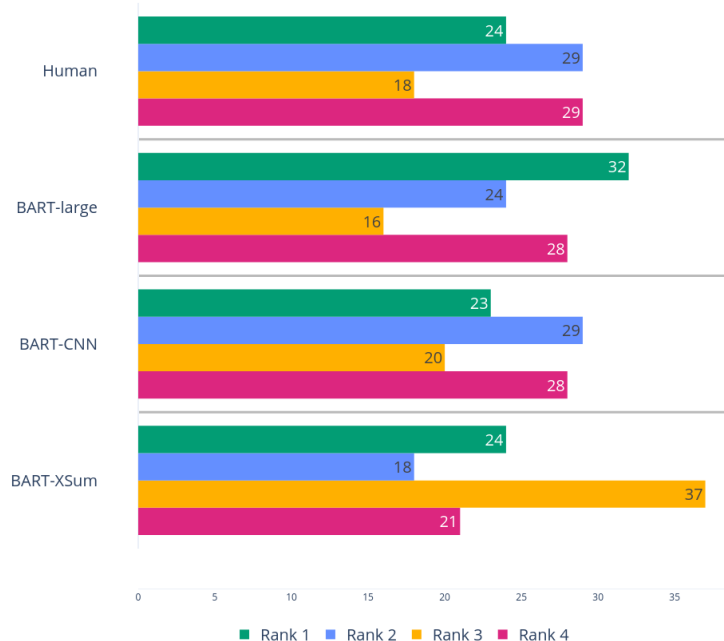


Figure 5.3.: Majority based ranking of our three conclusion generation models and the human baseline based on 100 randomly chosen arguments from our *Full* dataset. Horizontal bars show the number of times an argument of the corresponding model was ranked to one of the four ranks.

Finally, we also investigate the performance of our models split into sufficient and insufficient arguments. Figure 5.4 shows the absolute difference in majority ranks between both classes. Similar to our automatic evaluation results, we could not find a clear difference between both classes. Thus the quality of generated conclusions seems to be independent of the *Local Sufficiency*. Considering *BART-large* as the best overall model, we found that although the number of conclusions on Rank-4, which are generated of premises from an insufficient argument, is much higher (10) compared to their sufficient counterparts, for the second-lowest rank, the relationship is the other way around (6). However, the average overall rank for *BART-large* is slightly better for sufficient compared to insufficient conclusions (2.26 vs. 2.54). For the other models and the Human baseline, we found no significant difference in their overall ranking.

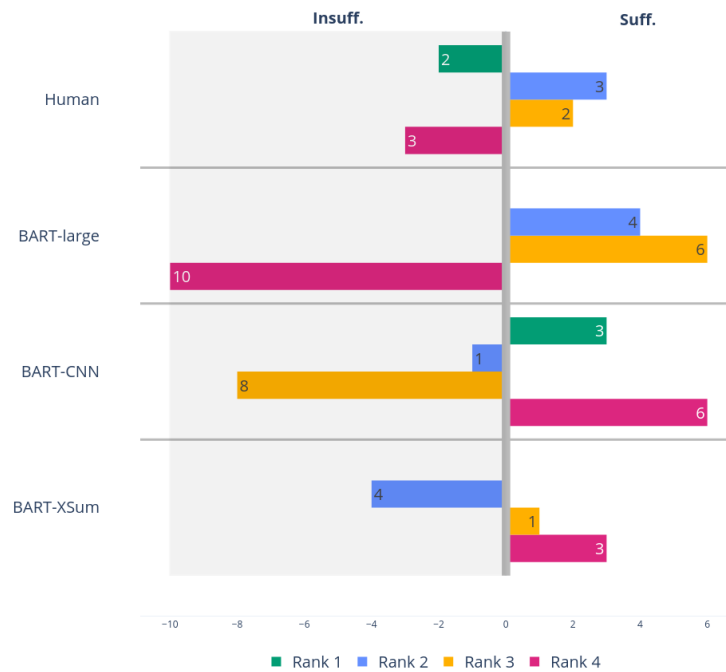


Figure 5.4.: Difference of the majority based ranking of our three conclusion generation models and the human baseline based on 100 randomly chosen arguments from our *Full* dataset, split into sufficient and insufficient arguments. Horizontal bars show the absolute difference in the number of times arguments of the two classes were ranked to one of the four ranks.

5.3 Indirect Local Sufficiency Assessment

Analogous to the direct local sufficiency assessment in Section 5.1, we use the BERT (Devlin et al., 2018) model to assess the *Local Sufficiency* of an argument in our indirect approach. However, in contrast to the former, we use the previously generated conclusions of the BART-large model as an additional input to the BERT model. Table 5.11 shows the corresponding metrics for the used approaches as well as some additional experiments we have run to evaluate our hypothesis of the imbalance in importance between conclusions and premises in *Local Sufficiency* assessment. Note that all the results here are on the *C&P* dataset we created for this task. We refer to the *Local Sufficiency* assessment using only the premises as *P*, only the ground truth conclusions as *C*, only the generated conclusions as *GC*, both the ground truth conclusions together with the generated conclusions as *C+GC*, our standard setup from Section 5.1 using the ground truth conclusions and premises as *C+P*, the generated conclusions together with the ground truth premises as *GC+P* and all three components together as *C+GC+P*. To join the components to a single sequence, we use the same procedure described in Chapter 4.

	Accuracy	Macro F1	Macro Prec.	Macro Rec.
P	.817 ± .028	.788 ± .071	.806 ± .052	.780 ± .061
C	.641 ± .036	.553 ± .063	.582 ± .048	.567 ± .144
GC	.632 ± .025	.532 ± .043	.560 ± .038	.544 ± .106
C+GC	.659 ± .026	.571 ± .036	.762 ± .030	.396 ± .092
C+P	.837 ± .034	.811 ± .041	.881 ± .024	.740 ± .058
GC+P	.832 ± .030	.799 ± .038	.831 ± .038	.785 ± .046
C+GC+P	.826 ± .043	.796 ± .051	.820 ± .056	.783 ± .057

Table 5.11.: Results of the indirect assessment approaches based on two five-fold cross-validations compared a human upper bound and the results from the direct *Local Sufficiency* assessment approach from Section 5.3.

Considering the results in Table 5.11, we found that although the direct BERT approach (*C+P*) is still the best overall model with a macro F1-score of .811, it does not significantly differ from the *GC+P* model with a macro F1-score of .799. Thus replacing the original conclusion with our generated conclusion only slightly affect the *Local Sufficiency* assessment. This result matches our findings in conclusion generation evaluation that there is little to no difference between our generated conclusions and conclusions written by humans. This is also supported by our evaluation for the *C* and *GC* models where the former performs insignificantly better than the latter. Consequently, adding the generated conclusion as an additional input (*C+GC+P*) does not increase the performance of our approach but slightly decreases it by a macro F1-score of $-.015$ and $-.003$ compared to the *C+P* and *GC+P*

models respectively. Contrary the *C+GC* model shows that there is at least some performance gained combining both premises with an increase in macro F1-score of $+0.018$ compared to the *C* model. However, this effect is most likely negligible as using the premises already accounts for it. We think that this decrease is the result of confusing the model, adding another conclusion. This is because the added conclusion either has the same target as the original conclusion and thus likely conveys the same semantic meaning, which, given our hypothesis, would mean it is sufficient, or it has a different target, which would mean it is insufficient. However, as already hypothesized during the valuation of our conclusion generation approaches and supported by the results of Alshomary et al., (2020b), identifying the target of premises is a very hard task that, given our study, even humans disagree on. This raises the question of how important the conclusion is in assessing the *Local Sufficiency* of an argument compared to the premises. To answer this question, we trained two BERT models, one that only uses the premises of an argument to predict the *Local Sufficiency* and one that uses only the conclusion. The evaluation shows that the premises' macro F1-score is only -0.023 lower than our best model, which uses both the conclusion and its premises. In contrast, the conclusion alone only reaches a score of $.553$, which is better than a majority vote but far lower than the premises. Thus premises are much more important in assessing the *Local Sufficiency* of an argument than the conclusion. However, we do not think that this observation is highly representative but rather a consequence of the domain of student essays and the quality of arguments written in this domain. Looking at the dataset, we could not find cases where conclusions are completely unfit to the premises in terms of matching targets, which shows that the general argument quality is rather high. Consequently, we believe that assessing the *Local Sufficiency* of arguments, especially if used in real-world applications, requires not only a bigger dataset in general but also a wider range of domains or at least a broader range of different quality arguments to really learn the task of *Local Sufficiency* assessment. However, considering that the *C+P* model outperforms the *P* model, we still see that this property is at least to some degree present in the dataset of Stab and Gurevych, (2017b). Finally, analyzing accuracy, macro precision, and macro recall, we see no changes to our findings in Section 5.1 which is a slightly better performance in precision compared to recall.

Conclusion

In this work, we assessed the *Local Sufficiency* of arguments, measuring whether the premises given in an argument are sufficient and together make it rational to draw the proposed conclusion (Wachsmuth et al., 2017a). To do this, we studied two approaches. First, the direct assessment of *Local Sufficiency* using the BERT (Devlin et al., 2018) model to predict whether an argument is sufficient or insufficient (binary) and second the indirect assessment of *Local Sufficiency* in which we first train BART (Lewis et al., 2019) models to generate a conclusion given a set of premises, and in a second step uses the generated as an argumentative feature, to augment the direct assessment approach further.

We proposed a news state-of-the-art *Local Sufficiency* assessment model, that outperform the previous SOTA CNN of Stab and Gurevych, (2017b) and achieves 96.7% of human performance on the AAE-v2 dataset (Stab and Gurevych, 2017a) annotated for *Local Sufficiency* by Stab and Gurevych, (2017b). Showing that large scale pre-trained *Language Models* can successfully improve the performance in fine-grained *Argument Quality Assessment* as already shown for holistic *Argument Quality Assessment* (Gretz et al., 2019b; Toledo et al., 2019). Additionally, we found that following the more strict view of the *Local Sufficiency* definition of (Wachsmuth et al., 2017a), that is, removing non-argumentative text from arguments based on the annotations of Stab and Gurevych, (2017a) decreases the performance of *Local Sufficiency* assessment models in general. Our investigation revealed that removing textual markers that indicate that a premise is an example improves the *Local Sufficiency* assessment. In contrast, the removal of connectivity between claims and premises, which often provides information about the number of premises, as well as removing textual markers introducing an opposing view, decreases the *Local Sufficiency* assessment performance. Using our adapted version of the AAE-v2 dataset, we have had a first look into NLP tasks, and their relationship to the task of conclusion generation, in which we found that abstractive summarization compared to extractive summarization and Language Modeling is closest, although finetuning on our dataset revealed that transferring knowledge from abstractive summarization to conclusion generation did not work in our case. We also showed in a manual annotation study that the conclusions generated by our models are on par or even slightly better compared to the one written by a human. This result is interesting as our conclusions' target is not always the same as the target of the conclusion written by the human. Thus we

hypothesize that given a set of premises, it is unclear or very hard to come up with a single ground truth target because the premises can be sufficient to support multiple conclusions with different targets. This hypothesis could also explain some of the problems in the work of Alshomary et al., (2020a), who tried to infer the target given a set of premises. Finally, we showed that our generated conclusion can be used as a replacement of the original conclusion without significantly affecting the *Local Sufficiency* assessment, which does support our previous finding that there is no significant difference in the quality of the generated conclusions and conclusions written by a human. However, as the conclusions are quite similar and we did not find a significant difference in conclusions both generated and written by a human in our study, adding the generated conclusion as an argumentative feature to assess the *Local Sufficiency* of an argument did not improve our results any further. In addition, as contrasting the generated and the ground truth conclusion to assess the *Local Sufficiency* performed much worse compared to all approaches using premises, we found that the importance of premises is much higher compared to the importance of conclusions in *Local Sufficiency* assessment. This result is of high importance as it shows that the general quality of arguments in the AAE-v2 dataset is very high in so far as all sets of premises fit the corresponding conclusion, at least to a certain degree. Thus, we argue that the task of *Local Sufficiency* assessment needs a bigger and more diverse dataset.

Considering our initial research questions, we showed the enormous potential of adapting large scale pre-trained Language Models to the task of *Local Sufficiency* assessment and thus promote further usage towards other dimensions of *Argument Quality Assessment*. Furthermore, we showed that LMs can successfully adapt to the task of conclusion generation to human-level performance. Finally, we did not find a proper way to incorporate text generation, i.e., conclusions generation, as an argumentative feature to improve *Local Sufficiency* assessment. Instead, we gained a lot of valuable insights into the task of *Local Sufficiency* assessment and the strengths and weaknesses of the currently only available dataset. We hope that our contributions to *Computational Argumentation* are useful not only in the theoretical nature but especially help to bridge the gap to real-world applications by showcasing the potential of SOTA NLP approaches and pointing out possible limitations that need to be overcome in the future.

6.1 Future Outlook

As we have touched on different parts of *Computational Argumentation*, including *Argument Quality Assessment*, *Argument Mining*, and also *Argument Generation*, there are some ideas that we hope to investigate or to be investigated by others in the future. We think that most important for future success is the availability of more diverse and consequently bigger datasets not only in the domain of *Local Sufficiency* assessment but also in related *Argument Quality Assessment* dimensions as well as in *Argument Mining*. However, we are well aware that the acquisition of such datasets often requires experts who are limited in numbers and expensive compared to crowd-based solutions. Nevertheless, we think it is an interesting topic to work on as it is the fundamental key for advancement both in theory and practice. We believe the goal of *Computational Argumentation* is social in nature. Thus results should help people in their everyday life. Considering that *Computational Argumentation* is a quite new field with limited research, we believe that it is important to unify concepts to work on common ground as, for example, the work by Wachsmuth et al., (2017a). While the former is a more general observation that we made throughout our time in the field, considering this thesis specifically, we still think that the task of generation holds some potential in shaping good arguments. As a starting point, we believe it would be interesting to study why removing non-argumentative parts of the text affects its logical quality. In a first test, we found that training a model to fuse premises and conclusions into a natural language text argument, which can be thought of as *Context* or *Connectivity* creation, does increase the *Local Sufficiency* assessment performance although not matching the performance before removal. However, learning to generate these parts can be useful in applications such as the IBM Debater¹ or at any point where arguments generated by a machine are presented to a human. Finally, it is still questionable if LMs in their current state can reason or make logical conclusions. Thus, investigating these models' real capabilities seems an interesting and important task if we want to create high-quality content.

¹<https://www.research.ibm.com/artificial-intelligence/project-debater/>

Appendix

A.1 Experiments and Evaluation: Conclusion Generation

A.1.1 Automatic Evaluation

Table 5.5

		B1	B2	R1	R2	M
—	BART-large	11.5 ± .334	1.71 ± .112	15.9 ± .412	2.74 ± .211	15.8 ± .552
	BART-CNN	13.7 ± .192	1.99 ± .157	18.4 ± .270	3.15 ± .216	16.8 ± .770
	BART-XSum	19.7 ± .779	2.44 ± .336	20.0 ± .937	3.32 ± .505	14.2 ± .991
Full	BART-large	26.5 ± 1.08	4.06 ± .524	21.5 ± 1.23	4.44 ± .613	13.4 ± 1.27
	BART-CNN	24.7 ± 2.32	3.63 ± .751	20.2 ± .667	4.23 ± .484	12.8 ± .952
	BART-XSum	27.9 ± 1.34	4.21 ± .410	21.1 ± .903	4.23 ± .490	12.4 ± .866
Suff.	BART-large	25.5 ± 1.63	3.84 ± .381	20.7 ± .891	4.09 ± .491	12.9 ± .890
	BART-CNN	24.2 ± 1.47	3.44 ± .473	20.5 ± 1.02	3.92 ± .454	13.2 ± .599
	BART-XSum	25.9 ± .975	3.86 ± .429	21.0 ± 1.00	4.19 ± .585	13.1 ± 1.04

Table A.1.: Results of the conclusion generation approach based on two five-fold cross validation **with** and **without** fine-tuning.

		BS-P	BS-R	BS-F1	Len.
—	BART-large	7.22 ± .789	24.1 ± .725	15.6 ± .551	52.8 ± 20.4
	BART-CNN	13.6 ± .664	28.0 ± .547	20.7 ± .341†	43.8 ± 11.1
	BART-XSum	19.9 ± 1.20	26.8 ± .760	23.4 ± .893 † ‡	25.9 ± 5.91
Full	BART-large	32.9 ± 1.03	28.4 ± .996	30.6 ± .609‡	17, 2 ± 3.71
	BART-CNN	30.8 ± 2.84	27.9 ± 1.01	29.4 ± 1.26	17.9 ± 4.38
	BART-XSum	34.5 ± 1.38	28.0 ± .841	31.2 ± .709‡	15.8 ± 3.66
Suff.	BART-large	32.2 ± 1.75	28.2 ± .711	30.2 ± 1.05‡	17.0 ± 3.55
	BART-CNN	29.6 ± 2.14	27.9 ± .875	28.8 ± 1.26	18.7 ± 4.22
	BART-XSum	32.7 ± 1.22	28.6 ± 1.35	30.7 ± 1.13‡	16.0 ± 3.86

Table A.2.: Results of the conclusion generation approach based on two five-fold cross validation **with** and **without** fine-tuning. † and ‡ mark significance over the *BART-large* and *BART-CNN* models.

Table 5.6

		B1	B2	R1	R2	M
BART-large	Suff.	11.9 ± .382	1.79 ± .191	16.0 ± .456	2.75 ± .323	15.5 ± .730
	Insuff.	10.7 ± .732	1.59 ± .220	15.8 ± 1.20	2.72 ± .429	16.3 ± 1.15
BART-CNN	Suff.	14.4 ± .494	2.06 ± .291	18.9 ± .553	3.20 ± .430	16.8 ± 1.08
	Insuff.	12.5 ± .599	1.89 ± .321	17.5 ± .821	3.01 ± .450	17.0 ± 1.09
BART-XSum	Suff.	19.9 ± .758	2.55 ± .260	20.0 ± .620	3.30 ± .410	14.3 ± .852
	Insuff.	19.4 ± 1.23	2.20 ± .727	19.8 ± 1.78	3.29 ± .959	14.0 ± 1.74

Table A.3.: Results of the conclusion generation approach based on two five-fold cross validation **without** fine-tuning split by sufficient and insufficient arguments.

		BS-P	BS-R	BS-F1	Len.
BART-large	Suff.	8.62 ± .814	24.0 ± .659	16.2 ± .566†	50.3 ± 20.9
	Insuff.	4.49 ± 1.77	24.4 ± 1.17	14.3 ± 1.16	57.7 ± 18.5
BART-CNN	Suff.	14.9 ± .937	28.1 ± .601	21.5 ± .505†	41.7 ± 10.5
	Insuff.	11.1 ± .988	27.7 ± 1.06	19.3 ± .727	47.7 ± 11.2
BART-XSum	Suff.	19.7 ± 1.43	26.4 ± .739	23.1 ± .903	26.2 ± 5.77
	Insuff.	20.4 ± 1.32	27.5 ± 1.59	24.0 ± 1.38	25.3 ± 6.12

Table A.4.: Results of the conclusion generation approach based on two five-fold cross validation **without** fine-tuning split by sufficient and insufficient arguments. † marks significance improvement over the the other class.

Table 5.7

		B1	B2	R1	R2	M
BART-large	Suff.	26.6 ± 1.27	4.01 ± .671	21.4 ± 1.20	4.32 ± .749	13.3 ± 1.50
	Insuff.	26.1 ± 1.91	4.01 ± .986	21.5 ± 2.11	4.63 ± .973	13.5 ± 1.81
BART-CNN	Suff.	24.8 ± 2.13	3.65 ± .789	20.0 ± .979	3.99 ± .612	12.6 ± 1.42
	Insuff.	24.5 ± 3.13	3.59 ± .962	20.6 ± 1.49	4.07 ± .984	13.1 ± .952
BART-XSum	Suff.	28.4 ± 1.39	4.24 ± .693	21.2 ± .951	4.24 ± .715	12.3 ± 1.13
	Insuff.	27.0 ± 1.96	4.13 ± .755	20.9 ± 2.07	4.16 ± .963	12.6 ± 1.85

Table A.5.: Results of the conclusion generation approach trained on the Full dataset based on two five-fold cross validation **with** fine-tuning split by sufficient and insufficient arguments.

		BS-P	BS-R	BS-F1	Len.
BART-large	Suff.	32.5 ± .954	27.9 ± 1.08	30.3 ± .429	17.3 ± 3.79
	Insuff.	33.5 ± 1.61	29.2 ± 1.48	31.4 ± 1.30†	17.1 ± 3.56
BART-CNN	Suff.	30.8 ± 2.79	27.5 ± 1.09	29.2 ± 1.15	17.9 ± 4.29
	Insuff.	31.0 ± 3.23	28.6 ± 2.08	29.8 ± 2.15	18.1 ± 4.54
BART-XSum	Suff.	34.6 ± 1.80	27.6 ± 1.18	31.1 ± 1.04	15.8 ± 3.63
	Insuff.	34.2 ± 1.10	28.7 ± 1.39	31.5 ± .964	15.9 ± 3.71

Table A.6.: Results of the conclusion generation approach trained on the Full dataset based on two five-fold cross validation **with** fine-tuning split by sufficient and insufficient arguments. † marks significance improvement over the the other class.

Table 5.8

		B1	B2	R1	R2	M
BART-large	Suff.	25.5 ± 1.46	3.81 ± .320	20.6 ± .907	3.97 ± .487	12.9 ± 1.20
	Insuff.	25.4 ± 2.21	3.91 ± .776	21.0 ± 1.72	4.27 ± .821	13.0 ± .837
BART-CNN	Suff.	24.8 ± 1.58	3.59 ± .425	20.8 ± 1.22	3.99 ± .574	13.3 ± .832
	Insuff.	23.1 ± 1.86	3.16 ± .759	20.0 ± 1.24	3.68 ± .622	13.0 ± .762
BART-XSum	Suff.	26.3 ± 1.37	4.01 ± .516	21.2 ± 1.36	4.14 ± .843	13.2 ± 1.50
	Insuff.	25.1 ± 1.17	3.57 ± .739	20.6 ± 1.23	4.23 ± .802	13.0 ± 1.05

Table A.7.: Results of the conclusion generation approach trained on the Suff. dataset based on two five-fold cross validation **with** fine-tuning split by sufficient and insufficient arguments.

		BS-P	BS-R	BS-F1	Len.
BART-large	Suff.	31.8 ± 1.40	27.8 ± .996	29.9 ± .806	17.7 ± 4.31
	Insuff.	32.9 ± 2.82	28.9 ± 1.59	31.0 ± 2.13	17.3 ± 4.11
BART-CNN	Suff.	30.1 ± 2.03	27.9 ± .701	29.1 ± 1.01†	18.5 ± 4.16
	Insuff.	28.6 ± 2.48	27.8 ± 1.67	28.7 ± 1.84	18.9 ± 4.75
BART-XSum	Suff.	32.7 ± 1.70	28.5 ± 1.46	30.6 ± 1.43	17.3 ± 3.89
	Insuff.	32.7 ± .850	28.8 ± 1.79	30.8 ± 1.17	17.1 ± 3.97

Table A.8.: Results of the conclusion generation approach trained on the Suff. dataset based on two five-fold cross validation **with** fine-tuning split by sufficient and insufficient arguments. † marks significance improvement over the the other class.

A.1.2 Manual Evaluation

LS	Premises	Human	BART-large	BART-CNN	BART-XSum
Yes	<ul style="list-style-type: none"> When the students study with their friends, they should have schedule to use their time in an effective manner Group-studying causes some kind of internal obligation to manage the time, inasmuch as we always care about other people and we do not want to waste their time 	<p>Time management is another brilliant benefit of group-studying</p>	<p>Group-studying is a good way for students to manage their time</p>	<p>Spending too much time on social activities is not the best way to manage the time</p>	<p>Group-studying should not be allowed</p>
Yes	<ul style="list-style-type: none"> This kind of gender biased approach is detrimental for the development of the society Women have proved their mettle in every field Women like margaret thatcher, indira gandhi, mari curie etc., are symbols of women's intelligence and skills 	<p>Some male chauvinists think that women are incapable in particular fields, like maths, mechanical engineering, defense etc</p>	<p>It is unfair for the society to judge women based on their looks and not their abilities</p>	<p>Women have proved their mettle in every field, especially in science and medicine</p>	<p>There is no place for women to be treated as second class citizens</p>
No	<ul style="list-style-type: none"> With savings you can afford some luxurious goods and fulfill your dreams I made a month-long backpack trip through southeast asia last summer with my girlfriend Without our savings we could not have pursued our long-time dream 	<p>Working hard and saving money for later on investments is a great way of getting a satisfying feeling and to reward yourself for something you have accomplished</p>	<p>Saving is the best way to achieve your goals</p>	<p>It is very important to save money</p>	<p>Without savings you would not be able to pursue your dream</p>
No	<ul style="list-style-type: none"> If one can efficiently acquire the information, then, there might come along more opportunities for him to achieve his goals There are various advertisements here and there, whether on the leaflets or on the big screen showing things of all sorts in commercial zones If one is seeking a job right in the city or planning somewhere to travel, or just wants to buy a product in a dependable store, he might refer to what he has seen or heard from the advertisements to make a relatively wise choice 	<p>Living in big cities means more exposure to stimulation, like all kinds of information</p>	<p>There are many sources of information that one can rely on</p>	<p>Advertisers provide useful information to the public, particularly to the lower income group</p>	<p>Advertisers can help to improve the quality of life for the individual</p>

Table A.9.: Conclusions generated by our four models together with the conclusion and premises written by a human and the corresponding Local Sufficiency (LS) label.

LS	Premises	Human	BART-large	BART-CNN	BART-XSum
Yes	<ul style="list-style-type: none"> Evaluating teachers made by students could improve teachers' performance Evaluation made by students is similar to feedback from customer that business owners need to know in order to improve their product Students might complain about their teachers spoke too fast that students could not catch up on information Teachers could improve themselves by speaking slower to make sure that students could get all information that they want to give 	Evaluating from students could improve teachers' performance	Students could improve teachers' performance by making them listen to them	Students could improve teachers' performance	Students' feedback could improve the performance of teachers
No	<ul style="list-style-type: none"> Guns will create a violent society if the trend continues In american, young adults and even juveniles can get access to guns, which leads to the tragedies of school gun shooting Some terrorists are able to possess more advanced weapons than the police, which makes citizens always live in danger 	Most violent crimes are related to the abuse of guns, especially in some countries where guns are available for people	Some terrorists are able to possess more advanced weapons than the police	Guns have become an indispensable part of terrorist activities, such as terrorism	The more guns we have, the more violent we become
Yes	<ul style="list-style-type: none"> Without proper training and knowledge, it would be impossible for them to perform their best in their careers Not only that, when they are in a university, they will have the opportunity to share their knowledge as well as to network for business associates and partners This can benefit them in future when they intend to open up their own companies or hospitals 	It is certainly true that university education is of paramount importance for certain professions such as doctor, engineer and lawyer	Without proper training and knowledge, it would be impossible for them to perform their best in their careers	They will have the chance to study in a university which is a good thing for them	Universities provide young people with the opportunity to develop their skills and knowledge
No	<ul style="list-style-type: none"> We need to consider that if the government expands the use scope of that languages, it can be really expensive Educational programmes will need to be given so as to keep young people learning and using them The local government has recently struggled with their costly investments on language teachers, facilities and appropriate materials 	Saving regional languages could be seen as a waste of money	The government should be aware of the cost of language education	The government should invest in the promotion of minority languages	Investing in the education system for minority languages will need to be considered

Table A.10.: Conclusions generated by our four models together with the conclusion and premises written by a human and the corresponding Local Sufficiency (LS) label.

LS	Premises	Human	BART-large	BART-CNN	BARTXSum
Yes	<ul style="list-style-type: none"> Trading are less active in poorer countries may also be caused by incomplete transportation infrastructure Residents in rural areas still depend on their own farm for food production Richer nations can provide financial service to poorer countries to encourage trading The former can also give consultation on setting up a complete and efficient transportation infrastructure to support trading 	In the area of trade, no one can deny that poorer countries are lagging behind richer countries due to shortage of money	The former can provide financial support to poorer countries to encourage trading	Trading in poorer countries is less active than in rich countries	Richer nations can provide financial service to poorer countries to encourage trading
No	<ul style="list-style-type: none"> With rapid development of the internet, people are able to enjoy quick electronic communication via internet As lots of chatting apps available online such as "wechat", people tend to send instant messages free of charge by using their phones rather than face to face communication 	Mobile phones have shortened the distance of communication	Communication has become more important than face-to-face contact	The development of mobile phones has also led to the increase in the speed of communication	Technology has changed the way people communicate with each other

Table A.11.: Conclusions generated by our four models together with the conclusion and premises written by a human and the corresponding Local Sufficiency (LS) label.

Bibliography

- Ajjour, Yamen, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein (2017). „Unit segmentation of argumentative texts“. In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 118–128 (cit. on p. 1).
- Al Khatib, Khalid, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein (2016). „Cross-domain mining of argumentative text through distant supervision“. In: *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1395–1404 (cit. on p. 33).
- Alshomary, Milad, Syed Shahbaz, Potthast Martin, and Wachsmuth Henning (2020a). „Target Inference in Argument Conclusion Generation“. In: (cit. on p. 72).
- Alshomary, Milad, Shahbaz Syed, Martin Potthast, and Henning Wachsmuth (2020b). „Target Inference in Argument Conclusion Generation“. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4334–4345 (cit. on pp. 34, 69).
- Aristotle (2007). *A Theory of Civic Dis-course*(George A. Kennedy, Translator). ClarendonAristotle series. Oxford University Press. (cit. on p. 1).
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). „Neural machine translation by jointly learning to align and translate“. In: *arXiv preprint arXiv:1409.0473* (cit. on pp. 12, 13, 34).
- Banerjee, Satanjeev and Alon Lavie (2005). „METEOR: An automatic metric for MT evaluation with improved correlation with human judgments“. In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72 (cit. on p. 26).
- Beardsley, Monroe C (1950). „Practical logic“. In: (cit. on p. 29).
- Bentahar, Jamal, Bernard Moulin, and Micheline Bélanger (2010). „A taxonomy of argumentation models used for knowledge representation“. In: *Artificial Intelligence Review* 33.3, pp. 211–259 (cit. on pp. 28, 29, 31).
- Cheng, Jianpeng, Li Dong, and Mirella Lapata (2016). „Long short-term memory-networks for machine reading“. In: *arXiv preprint arXiv:1601.06733* (cit. on pp. 14, 15).
- Chowdhary, KR (2020). „Natural language processing“. In: *Fundamentals of Artificial Intelligence*. Springer, pp. 603–649 (cit. on p. 7).
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). „Bert: Pre-training of deep bidirectional transformers for language understanding“. In: *arXiv preprint arXiv:1810.04805* (cit. on pp. 3, 14, 18, 20, 21, 23, 45, 46, 48, 50, 68, 71).

- Eban, Elad, Mariano Schain, Alan Mackey, et al. (2017). „Scalable learning of non-decomposable objectives“. In: *Artificial intelligence and statistics*. PMLR, pp. 832–840 (cit. on p. 47).
- El Baff, Roxanne, Henning Wachsmuth, Khalid Al Khatib, Manfred Stede, and Benno Stein (2019). „Computational argumentation synthesis as a language modeling task“. In: *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 54–64 (cit. on p. 34).
- Farley, Arthur M and Kathleen Freeman (1995). „Burden of proof in legal argumentation“. In: *Proceedings of the 5th international conference on Artificial intelligence and law*, pp. 156–164 (cit. on p. 28).
- Feldman, Susan (1999). „NLP meets the Jabberwocky: Natural language processing in information retrieval“. In: *ONLINE-WESTON THEN WILTON- 23*, pp. 62–73 (cit. on p. 7).
- Freeley, Austin J and David L Steinberg (2013). *Argumentation and debate*. Cengage Learning (cit. on p. 28).
- Freeman, James B (2011). *Argument Structure:: Representation and Theory*. Vol. 18. Springer Science & Business Media (cit. on pp. 29, 30).
- Gleize, Martin, Eyal Shnarch, Leshem Choshen, et al. (2019). „Are You Convinced? Choosing the More Convincing Evidence with a Siamese Network“. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 967–976 (cit. on p. 2).
- Granger, Sylviane, Estelle Dagneaux, Fanny Meunier, and Magali Paquot (2009). *International corpus of learner English* (cit. on p. 37).
- Graves, Alex, Greg Wayne, and Ivo Danihelka (2014). „Neural turing machines“. In: *arXiv preprint arXiv:1410.5401* (cit. on pp. 14, 15).
- Gretz, Shai, Roni Friedman, Edo Cohen-Karlik, et al. (2019a). „A Large-scale Dataset for Argument Quality Ranking: Construction and Analysis“. In: *arXiv preprint arXiv:1911.11408* (cit. on p. 1).
- (2019b). „A Large-scale Dataset for Argument Quality Ranking: Construction and Analysis“. In: *arXiv preprint arXiv:1911.11408* (cit. on pp. 2, 71).
- Habernal, Ivan and Iryna Gurevych (2016a). „Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM“. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1589–1599 (cit. on p. 2).
- (2016b). „Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM“. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1589–1599 (cit. on p. 35).
- Hidey, Christopher and Kathleen McKeown (2019). „Fixed that for you: Generating contrastive claims with semantic edits“. 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)*, pp. 1756–1767 (cit. on p. 34).
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). „Long short-term memory“. In: *Neural computation* 9.8, pp. 1735–1780 (cit. on pp. 10, 12).

- Hua, Xinyu, Zhe Hu, and Lu Wang (2019). „Argument generation with retrieval, planning, and realization“. In: *arXiv preprint arXiv:1906.03717* (cit. on p. 34).
- Hua, Xinyu and Lu Wang (2018). „Neural argument generation augmented with externally retrieved evidence“. In: *arXiv preprint arXiv:1805.10254* (cit. on p. 34).
- Johnson, Ralph Henry and J Anthony Blair (2006). *Logical self-defense*. Idea (cit. on p. 40).
- Keskar, Nitish Shirish, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher (2019). „Ctrl: A conditional transformer language model for controllable generation“. In: *arXiv preprint arXiv:1909.05858* (cit. on p. 34).
- Le, Dieu Thu, Cam-Tu Nguyen, and Kim Anh Nguyen (2018). „Dave the debater: a retrieval-based and generative argumentative dialogue agent“. In: *Proceedings of the 5th Workshop on Argument Mining*, pp. 121–130 (cit. on p. 33).
- Lewis, Mike, Yinhan Liu, Naman Goyal, et al. (2019). „Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension“. In: *arXiv preprint arXiv:1910.13461* (cit. on pp. 3, 18, 21–23, 25, 34, 47, 48, 57, 71).
- Liddy, Elizabeth D (1998). „Enhanced text retrieval using natural language processing“. In: *Bulletin of the American Society for Information Science and Technology* 24.4, pp. 14–16 (cit. on p. 7).
- Lin, Chin-Yew (2004). „ROUGE: A Package for Automatic Evaluation of Summaries“. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81 (cit. on p. 25).
- Luong, Minh-Thang, Hieu Pham, and Christopher D Manning (2015). „Effective approaches to attention-based neural machine translation“. In: *arXiv preprint arXiv:1508.04025* (cit. on p. 14).
- Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). „Efficient estimation of word representations in vector space“. In: *arXiv preprint arXiv:1301.3781* (cit. on p. 19).
- Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. (2016). „Abstractive text summarization using sequence-to-sequence rnns and beyond“. In: *arXiv preprint arXiv:1602.06023* (cit. on pp. 3, 48).
- Narayan, Shashi, Shay B. Cohen, and Mirella Lapata (2018). „Don’t Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization“. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium (cit. on pp. 3, 48).
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). „BLEU: a method for automatic evaluation of machine translation“. In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 311–318 (cit. on p. 24).
- Park, ChaeHun, Wonsuk Yang, and Jong C Park (2019). „Generating Sentential Arguments from Diverse Perspectives on Controversial Topic“. In: *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pp. 56–65 (cit. on p. 34).
- Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). „Glove: Global vectors for word representation“. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543 (cit. on p. 19).

- Popat, Kashyap, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum (2017). „Where the truth lies: Explaining the credibility of emerging claims on the web and social media“. In: *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1003–1012 (cit. on p. 32).
- Potash, Peter and Anna Rumshisky (2017). „Towards Debate Automation: a Recurrent Model for Predicting Debate Winners“. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2465–2475 (cit. on p. 2).
- Puthiya Parambath, Shameem, Nicolas Usunier, and Yves Grandvalet (2014). „Optimizing F-measures by cost-sensitive classification“. In: *Advances in Neural Information Processing Systems 27*, pp. 2123–2131 (cit. on p. 47).
- Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). *Improving language understanding by generative pre-training* (cit. on pp. 18, 19, 21).
- Radford, Alec, Jeffrey Wu, Rewon Child, et al. (2019). „Language models are unsupervised multitask learners“. In: *OpenAI Blog 1.8*, p. 9 (cit. on pp. 18, 20, 21, 23).
- Reed, Chris and Douglas Walton (2003). „Argumentation schemes in argument-as-process and argument-as-product“. In: (cit. on p. 28).
- Reisert, Paul, Naoya Inoue, Naoaki Okazaki, and Kentaro Inui (2015). „A computational approach for generating toulmin model argumentation“. In: *Proceedings of the 2nd Workshop on Argumentation Mining*, pp. 45–55 (cit. on p. 33).
- Rinott, Ruty, Lena Dankin, Carlos Alzate Perez, et al. (2015). „Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection“. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 440–450 (cit. on p. 1).
- Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). „Learning representations by back-propagating errors“. In: *nature 323.6088*, pp. 533–536 (cit. on pp. 10, 12).
- Samadi, Mehdi, Partha Pratim Talukdar, Manuela M Veloso, and Manuel Blum (2016). „ClaimEval: Integrated and Flexible Framework for Claim Evaluation Using Credibility of Sources.“ In: *AAAI*, pp. 222–228 (cit. on p. 32).
- Sato, Misa, Kohsuke Yanai, Toshinori Miyoshi, et al. (2015). „End-to-end argument generation system in debating“. In: *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pp. 109–114 (cit. on p. 33).
- Schiller, Benjamin, Johannes Daxenberger, and Iryna Gurevych (2020). „Aspect-Controlled Neural Argument Generation“. In: *arXiv preprint arXiv:2005.00084* (cit. on p. 34).
- Shermis, Mark D and Jill C Burstein (2003). *Automated essay scoring: A cross-disciplinary perspective*. Routledge (cit. on p. 1).
- Simpson, Edwin and Iryna Gurevych (2018). „Finding Convincing Arguments Using Scalable Bayesian Preference Learning“. In: *Transactions of the Association for Computational Linguistics 6*, pp. 357–371 (cit. on p. 2).
- Stab, Christian Matthias Edwin (2017). „Argumentative writing support by means of natural language processing“. PhD thesis. Technische Universität Darmstadt (cit. on p. 1).

- Stab, Christian Matthias Edwin, Johannes Daxenberger, Chris Stahlhut, et al. (2018). „ArgumentText: Searching for Arguments in Heterogeneous Sources“. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 21–25 (cit. on pp. 1, 32).
- Stab, Christian Matthias Edwin and Iryna Gurevych (2014). „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 (cit. on pp. 32, 38).
- (2017a). „Parsing Argumentation Structures in Persuasive Essays“. In: *Computational Linguistics* 43.3, pp. 619–659 (cit. on pp. 3, 5, 29, 32, 33, 38, 42–44, 52, 61, 71).
 - (2017b). „Recognizing insufficiently supported arguments in argumentative essays“. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 980–990 (cit. on pp. 2, 3, 5, 35, 40, 42, 45–47, 51–54, 69, 71).
- Stede, Manfred and Jodi Schneider (2018). „Argumentation mining“. In: *Synthesis Lectures on Human Language Technologies* 11.2, pp. 1–191 (cit. on p. 1).
- Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). „Sequence to sequence learning with neural networks“. In: *Advances in neural information processing systems*, pp. 3104–3112 (cit. on p. 12).
- Thomas, Stephen N (1981). *Practical reasoning in natural language*. Prentice-Hall (cit. on p. 29).
- Toledo, Assaf, Shai Gretz, Edo Cohen-Karlik, et al. (2019). „Automatic Argument Quality Assessment–New Datasets and Methods“. In: *arXiv preprint arXiv:1909.01007* (cit. on pp. 2, 71).
- Toulmin, Stephen E (2003). *The uses of argument*. Cambridge university press (cit. on pp. 28, 30).
- Van Eemeren, Frans, Rob Grootendorst, and Frans H van Eemeren (2004). *A systematic theory of argumentation: The pragma-dialectical approach*. Cambridge University Press (cit. on p. 28).
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, et al. (2017). „Attention is all you need“. In: *Advances in neural information processing systems*, pp. 5998–6008 (cit. on pp. 14–17, 19–21, 23).
- Wachsmuth, Henning, Khalid Al Khatib, and Benno Stein (2016). „Using argument mining to assess the argumentation quality of essays“. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1680–1691 (cit. on pp. 2, 33, 35, 37).
- Wachsmuth, Henning, Nona Naderi, Yufang Hou, et al. (2017a). „Computational argumentation quality assessment in natural language“. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 176–187 (cit. on pp. 1, 2, 8, 29–32, 35, 37, 40, 51, 71, 73).

- Wachsmuth, Henning, Martin Potthast, Khalid Al Khatib, et al. (2017b). „Building an argument search engine for the web“. In: *Proceedings of the 4th Workshop on Argument Mining*, pp. 49–59 (cit. on pp. 1, 32).
- Wachsmuth, Henning, Manfred Stede, Roxanne El Baff, et al. (2018). „Argumentation synthesis following rhetorical strategies“. In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3753–3765 (cit. on p. 34).
- Walton, Douglas (2005). *Fundamentals of critical argumentation*. Cambridge University Press (cit. on p. 28).
- Walton, Douglas, Christopher Reed, and Fabrizio Macagno (2008). *Argumentation schemes*. Cambridge University Press (cit. on pp. 28, 30).
- Wang, Lu and Wang Ling (2016). „Neural network-based abstract generation for opinions and arguments“. In: *arXiv preprint arXiv:1606.02785* (cit. on pp. 34, 48).
- Werbos, Paul J (1990). „Backpropagation through time: what it does and how to do it“. In: *Proceedings of the IEEE* 78.10, pp. 1550–1560 (cit. on pp. 10, 12).
- Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi (2019). „Bertscore: Evaluating text generation with bert“. In: *arXiv preprint arXiv:1904.09675* (cit. on p. 27).
- Zhu, Yukun, Ryan Kiros, Rich Zemel, et al. (2015). „Aligning books and movies: Towards story-like visual explanations by watching movies and reading books“. In: *Proceedings of the IEEE international conference on computer vision*, pp. 19–27 (cit. on pp. 19, 20).