Leipzig University
Institute of Computer Science
Degree Programme Computer Science, M.Sc.

Generating Arguments Depending on Argumentation Schemes

Master's Thesis

Christian Staudte
Born Aug. 27, 1996 in Gera

Matriculation Number 3756415

1. Referee: Jun.-Prof. Dr. Martin Potthast
2. Referee: Khalid Ibrahim Jamal Al Khatib

Submission date: June 1, 2022

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, June 1, 2022

.....
Christian Staudte

Abstract

With the rapid development of language models in the past few years and the publication of GPT-3, new methodologies like prompt engineering consequently emerged. This prompting method has proven its capabilities for many use cases, one of them being the automatic generation of arguments. However, regarding this task, many researchers so far do not ground their generated arguments on any typology but accept any argumentative structure. Additionally, their approaches mostly require fine-tuning data, which does not always exist or is highly limited. Today's labelled argumentative datasets still lack in size and adding new arguments requires a lot of manual labour. Motivated by these issues and the potential of prompt engineering, this thesis investigates how to manually formulate prompts that in combination with vanilla language models can generate arguments based on Douglas Walton's argumentation scheme typology. In particular, I define five prompt structures that I use to formulate prompt templates for 22 of Walton's schemes, leading to 110 argument prompt templates. I then apply these templates – combined with 32 controversial topics from IBM and six open-source language models from Huggingface plus GPT-3 – to generate about 100.000 arguments. I first evaluate these arguments automatically based on content richness, stance, topic relevance and argumentativeness. Arguments that reach the highest overall scores are then manually re-evaluated based on the aforementioned metrics, as well as two additional ones: plausibility and bias. The manual results demonstrate that the GPT series generates the highest quality arguments. Furthermore, they show that the argument quality depends on the represented argumentation schemes, with *Arguments from Position to Know*, *Arguments from Expert Opinion* and *Arguments from Cause to Effect* reaching the best results. To overcome the limitations of the automatic approach, I also propose a hybrid method called AutoArg. It is a web tool users can apply to step-wise generate arguments with pre-defined prompt templates. This way they are able to manipulate or recreate arguments until they reach a satisfactory result.

Contents

Acknowledgements

I thank the authors of the webisthesis template for their excellent work! I also want to thank my second referee, Khalid Ibrahim Jamal Al Khatib, for the weekly meetings and advice, for giving me the chance to work with GPT-3 and for coordinating the hiring process of the manual evaluation process.

Chapter 1

Introduction

The rapid development of language models in recent years has given rise to a new method of information generation, namely prompt engineering. The concept of prompt engineering is to cleverly formulate a text snipped (prompt) which functions as input for a language model. Using this prompt, a language model is expected to generate some task-specific output. For example, to find out possible consequences of smoking cigarettes, one can define the prompt *Smoking cigarettes causes.* Given an autoregressive language model, this prompt should grammatically and semantically constrain it to generate a consequence of smoking, e.g. *lung cancer*.

Prompt engineering introduces a new language modelling tool that can be applied for different tasks. The task focused on in this thesis is argument generation. In contrast to argument mining, where arguments are extracted from unstructured texts like debating websites, argument generation utilises the intrinsic knowledge base of language models to create (novel) arguments. The smoking prompt above exemplifies this approach, as it leads to the generation of (possibly) negative consequences which can be stated in a debate. Although the smoking prompt provides users with consequences of smoking cigarettes, it is limited by its composition: arguments generated with it follow a cause-to-effect structure. However, in natural debates humans argue much more diverse.

This diversity of arguments has been analysed in multiple philosophical works and led to the formulation of different argument typologies, with one of the most comprehensive ones written by Walton et al. [2008]. Douglas Walton and his team apply an empirical approach to group similar arguments by their reasoning structure, namely premises and conclusions; they categorise 60 argumentation schemes with this approach. Additionally, they formulate critical questions for each scheme as helpful tool for questioning and evaluating scheme-related arguments. Their wide range of schemes indicates that an

argumentation system should provide users with diverse argument structures, but this is a complex task: A grouping by argumentation schemes implies a comprehensive analysis of the argument's (potentially missing) components and quality, therefore most argumentation systems rely on simplified categories like pro or contra. This might be easier to implement, but the information content is smaller compared to scheme-grouped arguments.

To fill this gap between computational argument generation methods and comprehensive argument typologies, I propose a prompt engineering system to automatically create arguments given a topic and scheme. These arguments could be included in existing scheme-aware corpora or function as basis for a debating tool. Depending on the chosen scheme, the system selects a prompt template which functions as key to generate a scheme-related argument. For example, if one wants to generate an argument for the scheme *Argument from Cause to Effect*, a respective template might be *[Topic] causes*, with topic being a user input. If a user chooses the topic *smoking*, the prompt template becomes the final prompt *Smoking causes* which then can be given to a language model. The intention behind this prompt is that the model outputs a scheme-relevant, argumentative text, for instance: *Smoking causes cancer. Cancer is bad. Therefore, you should not smoke.*

While this hypothetically generated argument represents the given scheme, language models tend to easily derail or generate implausible and incoherent texts. There are multiple factors to consider that influence a model's output, including the model architecture and its training data, the decoding mechanism during inference time, and the input prompt. Because each of these factors includes a multitude of parameters to choose from, I limit each parameter space to a finite number of values and then combine values from each space in an attempt to compare different parameter-settings. First, I limit the language model space to autoregressive transformer language models. I do not fine-tune any model on additional data but apply them in their original state, which helps to analyse each model's intrinsic argumentation capabilities. In sum, I choose a set of seven models: Transformer-XL, XLNet, T5, GPT-2, GPT-Neo, GPT-J and GPT-3. Second, during inference time, I rely on probabilistic nucleus sampling to create texts that are as similar to human writing as possible. At last, I manually design a set of prompt templates based on pre-defined *prompt types*. Prompt types each comprise a set of rules, describing what content and structure a prompt must have. The advantage of prompt types is that I can apply them independently of argumentation schemes to formulate comparable prompts.

To test the argumentative potential of each model-prompt-combination, I

extract a set of controversial and diverse topics from the IBM debater datasets[1]. I then generate arguments for all combinations of model-prompt-pairs and topics. Because this method results in a large set of arguments that cannot efficiently be manually evaluated, I propose a pipeline filtering approach. I first automatically evaluate each argument given four metrics: content richness, stance, argumentativeness and topic relevance. Next, I choose arguments reaching the highest average scores which are then manually evaluated by one expert with the same metrics plus a plausibility and bias metric. This approach allows me to automatically exclude arguments which do not reach some threshold score and ensures that during the manual evaluation only arguments with a certain quality are chosen.

My manually defined prompt templates are by no means absolute. Depending on the model, template and decoding parameters, the quality of generated texts can vary immensely. However, the amount of different parameters makes it difficult to test every prompt with every model, decoding mechanism and topic. To overcome this limitation of an automatic generation approach for a static set of topics, I additionally propose a hybrid approach inspired by the GPT-3 playground called AutoArg. AutoArg is a web tool that allows users to choose a language model, sampling approach, prompt template and topic to generate arguments. This tool circumvents four limitations of the fully automatic approach: (1) any topic can be used, (2) sampling parameters can be chosen freely, (3) arguments can be re-generated until the user is satisfied and (4) prompts can be manually changed to test whether prompt alternatives lead to better arguments. The full code base for both approaches – including the generated arguments and evaluations – can be found in my corresponding git repository[2].

After clarifying the research goal and the general methodology in this chapter, chapter 2 introduces the philosophical background of argumentation studies. This background information is important to understand how the categorisation and research of arguments have changed throughout history and why argumentation schemes add value to automatic argument classification tasks. Furthermore, I look at two distinct methods of automatic argument retrieval, namely argument mining and argument generation and how they function and differ. In chapter 3, I present my methodology. Not only do I introduce a way to automatically generate arguments given a topic and argumentation scheme but also a web tool which users can use to create their own arguments in a hybrid fashion. Chapter 4 comprises automatic and manual evaluations of all generated arguments. For the preliminary manual evaluation we hired one expert who annotated 200 of the best generated arguments according to the

[1] `https://research.ibm.com/haifa/dept/vst/debating_data.shtml` (12.05.2022).

[2] `https://git.webis.de/code-teaching/theses/staudte`.

automatic metrics. Chapter 5 concludes this thesis with concepts and ideas which should be considered in future works. It further includes a summation of all presented result.

Chapter 2

Background and Related Works

In their book, van Eemeren and Grootendorst [2003] introduce the following definition of argumentation:

> **Argumentation** is 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. [van Eemeren and Grootendorst, 2003, p. 1]

Argumentations following this definition, therefore, require at least two subjects: a critic and an arguer. Both must act rationally, meaning listening to each other's propositions and questioning these – if necessary – in a dialectical fashion. The goal of the arguer is to convince the critic of their standpoint or claim. A standpoint is defined as verbal expression in favour or against some topic, for example: *We should ban zoos!*

Elementary components of argumentations are propositions, which either aim to refute or justify the arguer's standpoint. Propositions take the form of arguments, i.e. inference and reasoning structures based on premises which lead to a conclusion [Blair, 2012]. A hypothetical argument for the standpoint *We should ban zoos!* could be: *Zoos exploit animals (premise 1). The exploitation of animals is bad (premise 2). Therefore, we should ban zoos (conclusion).* For an argumentation to be considered successful, all participants must accept the conclusion of an argument once they accept its premises. For example, if a critic accepts both premises in the previous argument without any objection or counter arguments, they also must accept the conclusion. The following dialogue exemplifies a complete argumentation between two subjects:

C: In my opinion, zoos only exist for human entertainment and thus are morally indefensible.

A: I do not think so. You see, the Association of Zoos and Aquariums says zoos provide public education about endangered species and help to find solutions to wildlife problems.

C: I think you are right. Zoos do not only exist for our entertainment.

In this example the [C]ritic accepts all stated premises by the [A]rguer and thus accepts the conclusion: Zoos do not only exist for our entertainment.

While this argumentation consists of a logical structure and falls into the category of argumentations defined above, affective and persuasive arguments do exist, too. These do not necessarily include all aspects of the previously stated definition, and to analyse them, one ought to rather focus on rhetorical features.[1] Another critical aspect of the definition is that many researchers claim individual episodes of reasoning (i.e. without an interlocutor) are arguments, too. These two points exemplify that inconsistent interpretations and definitions of argumentations exist, c.f. Blair [2012]. Although the field of argumentation is broad, my thesis is limited to arguments which all rely on cogency, following the introductory definition. Consequently, argument generation throughout the rest of this work does not focus on how to create persuasive arguments by rhetorical means, but on cogent arguments following specific reasoning structures. I should also add that individual reasoning acts can be mapped onto dialectic argument structures [Blair, 2012]. Therefore, this work does not differentiate between the number of arguers, meaning even a single person's argumentative monologue or thinking process can count as argumentation.

Going into more detail, cogent argumentations can be categorised further. In the above example one argument from A is enough to reach a conclusion, which is called *single argumentation*. Once multiple (counter-)arguments are presented, it becomes a *complex argumentation structure*. For instance, C could, instead of accepting the proposition, critically ask:

C: Does this association cooperate worldwide? What about all the zoos which do not associate with research facilities to conserve species and only focus on making money?

A discussion unfolds with pro and contra arguments until both parties reach a conclusion. Expounding on the presentation of arguments, van Eemeren and Grootendorst [2003] differentiate between *coordinative* and *subordinative argumentation*. Coordinative arguments are by themselves valid arguments, whereas subordinative arguments depend on each other. The arguer could answer the following:

[1] van Eemeren and Grootendorst, p. 24 [2003] mentioned this in the context of an epistemo-rhetorical approach.

A: As per definition, the main goal of zoos is not to make high profits. (1) They are for the education and study of animals, (2) they present animals in a structured environment, and (3) the animals are protected against wild animals and natural predators.

All points A raises are independent from each other and act *coordinatively*. C now might respond:

C: Last week my uncle watched a documentary where it was stated that most experts and scientists agree that zoos do more harm than good. Zoos are usually built by big business and are used for profit and to make money.

In this example the uncle is in a position to know about profit-oriented zoos since he watched a documentary with experts who, apparently, have a deeper understanding of the topic. However, this argument resembles a subordinative structure, as one fallacy invalids the whole argument: What if the uncle remembers the statements incorrectly? What if the experts' field of knowledge does not align with the proposition stated? This does not imply that chained arguments are inherently prone to fallacies (c.f. legal cases, where a chain of arguments and evidence is needed). Yet, they immediately fall apart once a fallacy is found.

To differentiate between subordinative and coordinative arguments is only one strategy of categorising argumentative structures. Alternatively, arguments can simply be grouped by their goal, i.e. pro or contra. Multiple concepts exist on how to group arguments, but before I go into further detail about these concepts, in the next two sections I first establish a broader context by introducing argumentation studies in general and a short history of arguments.

2.1 The Estates of Argumentation Studies

The interplay between argumentative structures only represents a minute aspect of argumentation studies. One can also analyse the cogency or persuasiveness of single arguments, ask questions of acceptability, consider real-life scenarios in contrast to theoretical argumentation settings, categorise and structure similar argument patterns, and more.

In their book, van Eemeren and Grootendorst [2004] describe five estates of argumentation studies. The first one is the philosophical estate. In this research area, questions of *When does someone act reasonable?* and *What is acceptability?* are discussed. To illustrate the complexity of the philosophical estate, I want to emphasise two philosophical groups: anthropologico-relativists

and critical-rationalists. While the first group focuses on socio-cultural factors during argumentations where the validity of an argument depends cultural factors, the second group focuses on rules and standards established during a discussion to decide on validity. As a result, the second group detaches arguments from pure cultural elements and focuses on resolving different opinions between the interlocutors. These two groups not only exemplify that arguments are highly contextual, but also that types of contexts can differ. They show that argument understanding and evaluation is a highly complex task relying on a multitude of features.

The second estate is a theoretical one, where philosophical concepts are converted to models. The goal is to formulate a set of rules on what a universal argumentation pipeline can look like, focussing on persuasion strategies on the one side and finding a resolution of differences on the other.

Most argument mining approaches are part of the third area: the analytical estate. Its goal is to reconstruct and formalise natural argumentations by mapping their argumentative components onto a theoretical model. As described in the previous section, arguments can be analysed by their persuasiveness or cogency, but the problem of the former is that no consistent method exists to analyse persuasiveness. For cogency, however, a dialectical analysis exists which is a resolution-oriented method for reconstructing argumentation. The goal of this analysis is to find argumentative sections which either support or invalidate certain claims. An advantage of this method is that these argumentative sections can easily be formalised as knowledge graphs with nodes representing factual information and edges representing refuting or supporting relations between these nodes.

The fourth estate (and the one most relevant to this thesis) is the empirical estate. While one can map natural arguments onto a pre-defined theoretical model, this model might not be final. Although it could depict the predominant argumentative structures of our daily life, novel argumentative compositions possibly are not representable with it, requiring some form of refinement. In the empirical estate, arguments not following some pre-defined model are neither reformatted nor discarded but directly influence the given model, adding new information on how arguments can be structured. Consequently, an empirical model could also completely be based on a set of natural arguments and groupings of those.

The last estate is the practical one. While the other four estates comprise how arguments are structured and function, this estate focuses on competence when using arguments and what to keep in mind to hold a successful argumentation. Formal approaches are used to analyse whether arguments are valid and what kind of underlying structures an argumentation includes.

All five estates constitute the base of modern argumentation studies. While

this thesis focuses on the empirical estate, namely the collection of arguments, a fundamental understanding of all estates is important to locate this work in contemporary argumentation research. This research, however, has changed profoundly throughout the last century by questioning when an argument is valid and how to interpret possible fallacies. This change plays an important role for the question of how to categories arguments, so I briefly elucidate the historic development of argumentation studies in the following section.

2.2 A Historical Perspective on Argumentation

At the beginning of this chapter, I showcased some arguments related to the topic *zoos*. These are neither deductively valid nor inductively strong. They are defeasible. Up to the 1970s, such arguments were dismissed as fallacies or labelled "non-formal" [Blair, 2012]. However, as stated by Blair, the deductive-inductive-dichotomy is not exhaustive and many real-life arguments do not fit either of these two categories perfectly. Argumentation scholars, who focus on non-formal logic, argue that alternative categories exist; thus, one should not discard supposedly invalid arguments. Peirce [1992], for example, proposes a new category of *abductive* arguments. While non-formal logic portraits our daily life experiences more realistically than formal logic, the latter still dominates the academic context, however.

Argumentation and reasoning studies have been an influential philosophical research area for more than 2,000 years, with Aristotle being one of the most influential early scholars. Aristotle based his work on Plato's dialogue and established an extensive vocabulary to study and categorise arguments. He defines topics (*topoi*) as abstract inference patterns [Macagno and Walton, 2015] and *endoxa* as opinions commonly accepted by a majority [Walton et al., 2008]. He then uses these concepts to not only differentiate between logical and rhetorical argument contexts but also between deductive and inductive argument types. Deductive means that an argument is valid as long as all its premises are valid. If this is the case, the conclusion of such an argument is absolute and irrefutable. This means that no matter what kind of future evidence is presented, as long as the given premises hold true, the conclusion must hold true, too. Deductive logic can also be referred to as monotonic logic. In contrast, inductive arguments entail weaker structures, whose conclusion are not absolute. Premises only support the conclusion to some degree [Hawthorne, 2021]. The following examples demonstrate the difference:

INDUCTIVE: Every raven in a random sample of 3200 ravens is black. This strongly supports the following conclusion: All ravens are black [Hawthorne, 2021].

DEDUCTIVE: All men are mortal. Socrates is a man. Therefore Socrates is mortal.

In other words, inductive arguments are built on top of observations to formulate a claim/hypothesis, while deductive arguments first establish a universal claim which holds true for all tests. The aforementioned argument to exemplify the deductive pattern represents a *syllogism* [Walton, 2005]. Another popular early form of deductive arguments are *modus ponens* arguments. Their first premise is a conditional relation (if A, then B); the second premise then states that A is indeed the case. Consequently, one can argue that B results. An example by Walton [2005]:

PREMISE 1: **If** Lugano is in Switzerland, **then** Lugano is in Europe.
PREMISE 2: Lugano is in Switzerland.
CONCLUSION: Therefore, Lugano is in Europe.

Aristotle commenced these concepts and the study of formal logic. In Roman times, premisses in arguments were further divided into major and minor premises [van Eemeren and Grootendorst, 2004]. In this manner, propositions can be analysed by functionality, with major premises linking two propositions and the minor premise indicating a present state. These linkages might help us to structure and understand arguments, but they cannot represent defeasible parts of arguments.

Formal logic, which was elaborated throughout the middle ages, became popular in areas like philosophy and mathematics, as well as in recent studies like computer science and linguistics. However, the strict rules imposed by formal logic discard many real-life arguments that are not perfectly deductive or inductive. Non-formal argumentative frames that are much more common [Blair, 2012] can hardly be analysed, c.f. online debates, newspaper editorials, a university discourse or discussions among friends. Additionally, formal logic does not consider dialectic settings. Even inductive arguments rather focus on strengthening the base of an argument (by presenting more data) instead of questioning the relation between the given data and claim.

The impetus for an increasing interest in defeasible arguments was given by Toulmin [2003][2]. He challenges the sufficiency of argument and logic studies by comparing existing logic theories with cases of jurisprudence (empirical approach). He demonstrates that his selected arguments from legal cases not only entail major and minor premises, but also propositions like evidence, testimonies, interpretations, additional claims, pleas, verdicts and sentences. Since

[2]Referring to the second edition of Toulmin's originally published book in 1958. The updated edition only adds slight changes. The main ideas and models of the original work remain.

these propositions are typically left out in formal logic, he establishes a new formalisation of defeasible arguments (later referred to as Toulmin Method or Toulmin Model). Instead of collecting more data and information to support a given claim (as would be done for inductive reasoning), relations between minor premises and the claim instead assume a decisive role to achieve acceptance according to the Toulmin Method. Toulmin calls these relations between the base (*data*) of an argument and its conclusion (*claim*) *warrants*. Let us consider an example to better illustrate this method. *I think coal mining is disastrous to our environment; many experts have claimed so.* Toulmin would split this reasoning structure as follows:

CLAIM (C): Coal mining is disastrous to our environment
DATA (D): many experts have claimed so
IMPLICIT WARRANT (W): *If many experts make a claim, it must be true*

This example illuminates the importance of warrants convincingly. The validity of an argument does not depend on the amount of data provided, but on the argumentative strength between data and claim. For instance, one could critically ask if everything an expert states is in fact true or if the person in question even is an eminent expert in the corresponding field.

Expert opinions are one among multiple types of warrants, which differ in quality and legitimacy. Toulmin [2003] describes them as recurring patterns to validate the soundness of an argument. Changing the data (D) in the above example to *I read it on some website* simultaneously manipulates the warrant to: *If one reads something on a website, it must be true.* Without further information one could argue that the expert warrant in comparison is more legitimate. To better differentiate and evaluate warrants, Toulmin adds more components to his model: *qualifiers* (Q) to measure certainty and *rebuttals* (R) to incorporate critical questions against the warrant. With the introduction of rebuttals, Toulmin integrates a dialectic component, because rebuttal leads to discussion on the opponent's side. Figure 2.1 portraits how the coal mining argument with expert opinion can be expanded to match the model by Toulmin. Although Toulmin further refines his model, an understanding of the here presented constituents is sufficient.

In contrast to deductive arguments, the method by Toulmin [2003] specifies nonmonotonic (defeasible) arguments which lead to tentative conclusions. This means by adding new propositions to an argument (new data, evidence, ...), the previously accepted conclusion can be contested or even invalidated. This might sound similar to the inductive argument structure because presenting a non-black raven in the previous example would invalidate the conclusion, too. Yet, there is still a difference between data and warrants: Toulmin's model allows single warrants to be contested by adding new information, resulting in

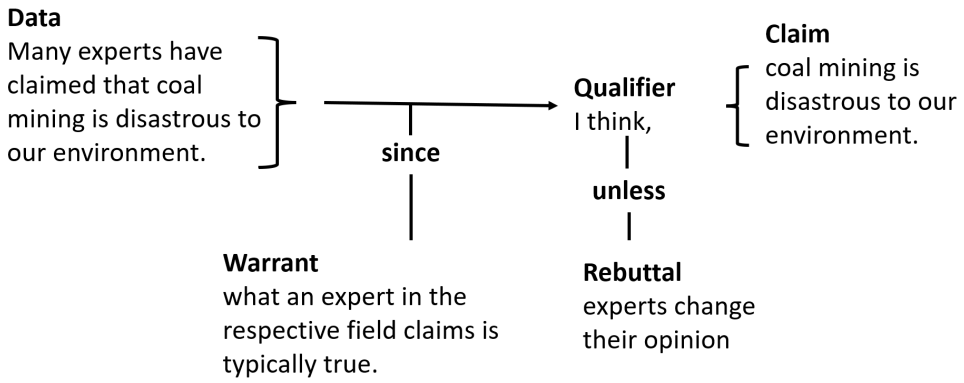


Figure 2.1: Coal mining argument with an expert opinion; structured using the model by Toulmin [2003] with data, claim, warrant, qualifier and rebuttal examples.

the invalidation of a previously accepted conclusion. In an inductive setting, a conclusion can only be contested by adding new data. Toulmin also argues that nonmonotonic arguments differ from the over-simplified formal validity in levels of abstractness and ambiguity. While formal logic does not put constraints on the field or domain one argues in, such constraints are necessary to formulate warrants. Kaplan and Berry-Rogghe [1991] support his claim and add that natural argumentation scenarios require a richer semantic system than formal logic strategies can provide.

Toulmin [2003] notes that different types of warrants with different levels of legitimacy exist, but does not go into more detail. His model might describe more realistic argument patterns, yet at its core it remains theoretical. It does not enable its users to directly evaluate and compare different types of warrants in an analytical fashion (c.f. section 2.1). To bridge this gap, subsequent works concentrate on so-called *argumentation schemes*.

2.3 Argumentation Schemes

Walton et al. [2008] define argumentation schemes as follows:

> Argumentation schemes are forms of arguments (structures of inference) that represent structures of common types of arguments used in everyday discourse, as well as in special contexts like those of legal argumentation and scientific argumentation. [Walton et al., 2008, p. 1]

Such schemes have been developed since the 1970s as described by van Eemeren and Garssen [2020]. They are based on different kinds of warrants and function as main tool to group similar argumentative structures for validation and

evaluation purposes. Naturally, different grouping methods exist which can be split into two categories:

1. *Top-down.* One can apply an analytical approach by formulating a theoretical system and thereafter validating it by grouping arguments.

2. *Bottom-up.* One can follow an empirical approach by grouping similar arguments and afterwards labelling each group.

Among the most well known typologies are the pragma-dialectical[3] one by van Eemeren and Grootendorst [2003], the user compendium by Walton et al. [2008] and a characteristics-based classification approach by Wagemans [2016].

The authors van Eemeren and Grootendorst [2003] build their schemes on top of a theoretical base by defining three types of relations between arguments and standpoints: symptomatic argumentation, argumentation based on a comparison and causal argumentation [van Eemeren et al., 2002]. On top of these groups they define variants and sub-types. Their causal relation, for example, has the following structure:

CONCLUSION: Y is true of X,
BECAUSE: Z is true of X,
AND: Z leads to Y.

An argument falling into this category is: *Lydia must have read a lot with poor light, because she has weak eyes. (And reading in poor light gives you weak eyes.)* As most argumentation scheme typologies refer to Toulmin [2003], they typically add some form of rebuttal, which in this case are *critical questions*. Critical questions are used to attack propositions or to ensure a claim is valid. For causal relations, a critical question would be: *Does Z always lead to Y?* Causal argumentations might seem similar to the deductive *modus ponens* structure, but this is intentional, since argumentation schemes do not abandon deductive or inductive elements, but rather move them from an idealistic abstract level to a more practical one while staying abstract enough to allow for an infinite number of substitutions [van Eemeren and Grootendorst, 2003]. While the three presented schemes are based on a top-down approach, one of the most renown argument typologies lies in the empirical estate: the new-dialectical scheme classification by Walton et al. [2008]. The authors present 60 argumentation schemes plus sub-schemes which are similar to the schemes defined by van Eemeren and Grootendorst Table 2.1 illustrates three schemes from the user compendium of Walton et al.

[3]Pragma (pragmatic) means the argument is based on human experience; dialectic means it is a dialogical process, i.e. it includes critical questions / counter-arguments) [van Eemeren and Grootendorst, 2003, p. 14].

Table 2.1: Three of the 60 argumentation schemes adapted from Walton et al. [2008].

1. Argument from Position to Know *Major Premise*: Source P is in position to know about things in a certain subject domain S containing proposition A. *Minor Premise*: P asserts that A is true (false). *Conclusion*: A is true (false).
2. Argument from Expert Opinion *Major Premise*: Source E is an expert in subject domain S containing proposition A. *Minor Premise*: E asserts that proposition A is true (false). *Conclusion*: A is true (false).
28. Argument from Cause to Effect *Major Premise*: Generally, if A occurs, then B will (might) occur. *Minor Premise*: In this case, A occurs (might occur). *Conclusion*: Therefore, in this case, B will (might) occur.

A more recent grouping approach by Wagemans [2016] defines binary characteristics and groups arguments depending on the combination of these. Consequently, it is an exhaustive approach. The latest version was published in 2019 and can differentiate between 36 argument types. Wagemans criticises the previously mentioned list-based scheme sets as lacking sufficient theoretical foundation. His reasoning is that these groupings are neither well-founded nor exhaustive. They are based on the experience of the authors or the available data and thus can continually change.

Although an exhaustive system as proposed might be better to group arguments, its defined features are limited and not yet able to represent all schemes of Walton et al. [2008], shown by [Visser et al., 2021]. When it comes to the typology of van Eemeren and Grootendorst [2003], compared to the one by Walton, it is less comprehensive and not as expressive. Consequently, this thesis relies on Walton's argumentation scheme typology as foundation to generate arguments.

2.4 Argumentation in Computer Science

Finding well-structured arguments (or simply argumentative components) in unstructured text corpora is a laborious task, with early works on automation processes reaching back to the 1990s. While these works usually focused on finding reasoning structures rather than complete arguments (Joskowicz et al. [1989], Kaplan and Berry-Rogghe [1991]), they demonstrate that it is possible to extract such structures from partly unstructured texts with formal logic and logic programming. Later, Girju [2003] provides a study on the automatic detection of causal relations with WordNet and a learning algorithm, which comes close to the *Arguments from Cause to Effect* scheme by Walton et al. [2008]. Beginning in the mid 2000s, the focus changed to the extraction of whole arguments from texts, a process referred to as argument(ation) mining [Lippi and Torroni, 2016].

An early work by Moens et al. [2007] describes how to automatically extract argumentative structures from legal texts by classifying sentences as argumentative or non-argumentative. The authors compare a Bayes classifier to a maximum entropy model, both trained on annotated arguments with each argument being limited to one sentence. In their tests, they reach a maximum accuracy score of 73.75% with a Bayes classifier, relating to the number of correctly extracted arguments divided by all extracted arguments. In later works, this binary classification is elaborated into a structural argument analysis during the mining process by searching for premises and claims. Stab and Gurevych [2014] exemplify this approach by using a support vector machine (SVM) to annotate clauses as *major claims*, *claims*, *premises* or *non-argumentative*. As training data they use a corpus of annotated essays from which they extract different linguistic features.

When it comes to the extraction of arguments and classifying them by schemes, many works simplify this task by defining their own scheme set, for example by categorising arguments employing their intended use: support, attack or counter-attack (Peldszus and Stede [2013], Stab and Gurevych [2014], Stab et al. [2018]). However, there are also works which rely on the predefined scheme set of Walton et al. [2008] to filter argumentative structures. For example, Feng and Hirst [2011] utilise five different schemes to categorise pre-defined arguments in an effort to reconstruct enthymemes for which they apply a C4.5 decision tree. They demonstrate that a one-against-others training approach reaches especially high classification scores for *Arguments from Example* and *Arguments from Practical Reasoning*. Lawrence and Reed [2016] additionally include the step of finding argumentative structures in an unstructured text. First, the authors extract propositions by using different linguistic classification features like n-grams, part-of-speech (POS) and lengths. They then define

proposition key words to classify four argumentation schemes, after which they propose a method to discover which of the extracted propositions form an argument when combined. Their results show that it is possible to identify the selected scheme components with F-scores between 0.78 and 0.91 by applying a Multinomial Naïve Bayes classifier.

2.5 Argument Generation

Instead of searching through unstructured texts to find and categorise arguments, this thesis follows the approach of automatically generating arguments with preferred scheme structures. The concept behind such an approach is to use autoregressive language models. Depending on a model's input, its intrinsic knowledge can be leveraged to generate specific texts. In this section I introduce different works that employ language models to generate arguments.

Wang and Ling [2016] propose a language model method to summarise opinionated texts into one argumentative sentence. It is an early work in the field of argument generation which focuses on the ability of an attention-based language model instead of copying and merging snippets from some input text. Although their work relies on additional text data and not only on the intrinsic knowledge base of the language model, they show that with their method it is possible to automatically generate coherent argumentative sentences.

Dave the Debater is an argumentation system by Le et al. [2018]. The authors designed a debating system, which can (1) generate new arguments or (2) retrieve arguments from a knowledge base. The generative model's architecture is based on a hierarchical recurrent RNN. While the system demonstrates promising results in debates with humans, it requires a user to begin the conversation with a claim. The initial claim, concatenated with all following dialogue sections, functions as model input. Adding dialogue sections to the input, however, might lead the model to derail from the original claim, moving to other topics or claims. The model also responds with any kind of argument without considering its structure and sometimes struggles to return coherent and relevant messages.

Hua and Wang [2019] built a LSTM decoder framework whose solvable tasks include the generation of counter-arguments, trained on Reddit's subcommunity */r/changemyview*. The authors specifically focus on the model's ability to control the generated content and its style. A topic and key phrases, which comprise the talking points, function as input. To create arguments following a general structure, the authors annotate three sentence structures in the training dataset: *claims*, *premises* and *functionals*, with the latter referring to general statements. After training, they apply BEAM search as decoding

strategy. In their evaluation section, they show that the counter-argument approach reaches significant improvements on scores like BLEU and ROUGE. Although their model proves its argumentation abilities, generated arguments do not consider specific argumentation schemes. Additionally, key phrases are required before the generation process, which further limits their framework.

Gretz et al. [2019] define a pipeline to automatically generate claims with GPT-2. They fine-tune four GPT-2 models on three training datasets, compare their performances over 96 topics[4] and demonstrate that they are able to generate novel claims. To improve the inference quality, they frame topics and claims respectively by adding Wikipedia and aspect sentences. Aspect sentences guide a model in a specific direction; for example, adding *Consider how this relates to the economy* to the topic acts as aspect which frames the topic with an economical theme. By adding context, ambiguity can be decreased and claims can be directed in a specific thematic direction. Although they test both context approaches, only adding the initial Wikipedia sentence relating to the topic improves the final score. After claims are generated, they are filtered with a claim detection algorithm that evaluates the relevance of each created claim. Claims not reaching a minimum threshold score are dropped. The authors highlight the importance of their detection tool to only extract high quality claims, which is also demonstrated in their evaluation. The pipeline approach to guarantee a certain relevance can also be made use of in an argument generation setup; however, claims are only one element of arguments. Complete arguments also provide a base to support or refute a given claim.

Schiller et al. [2021] propose a related approach by fine-tuning a CTRL model and naming it Arg-CTRL. It is a transformer language model which receives a topic, stance and aspect as inputs and outputs a single sentence argument. For example, with the input sequence *marijuana legalization PRO safer* the model outputs *Legalizing cannabis will help reduce crime rates (especially violent crimes) and make society safer overall.* In their evaluation section, the authors demonstrate that their generated arguments reach similar quality scores as human written arguments. They also test the absence of aspects in their training data and thus illustrate that without aspect information, Arg-CTRL is unable to generate aspect-related arguments. Consequently, their model requires topic-related knowledge during fine-tuning to generate useful arguments. Although this model does not consider argumentation schemes and requires additional topic data, its ability to receive instructions in the form of a prompt demonstrates the possibility of controlling an argument's content by prompt engineering.

Another work which applies GPT-2 to create arguments was composed

[4]As topics they utilise phrases containing goals like 'We should abolish term limits' and noun phrases like 'United States', which they directly apply as prompts.

by Khatib et al. [2021]. Their work relies on fine-tuned GPT-2 models trained on different argumentation knowledge graphs to generate arguments from claims. The graphs' nodes are noun phrases (concepts), which are connected to each other with positive or negative edges, for example *promotes, causes, suppresses, prevents*. Two related nodes can therefore be interpreted as cause-effect relation, c.f. *Arguments from Cause to Effect* by Walton et al. [2008]. After combining multiple argumentation sources, including *args.me*, *kialo* and various debate sites like *debatepedia.org*, they map the graphs onto natural texts that can be used to fine-tune GPT-2. A final input prompt has the form of a claim with structure $Concept_A$ *relation* $Concept_B$ with which a complete argument is inferred. Overall, the authors fine-tune four GPT-2 models on different knowledge graphs. For their evaluation they generate 400 arguments which are automatically and manually annotated. For an automatic evaluation the authors define three metrics: *topic relevance*, *argumentativeness* and *content richness*. These are domain independent and can be used to validate any large quantity of arguments or general texts. They additionally conduct a manual evaluation which leads to similar results as the automatic evaluation, hinting at the usability of the automatic metrics. Their evaluation results also demonstrate that the average scores over all fine-tuned GPT-2 models are higher than scores by the original GPT-2 model, meaning fine-tuning here improves the generated arguments.

Works by Khatib et al. [2021] and Schiller et al. [2021] come closest to the goal of this thesis. However, both approaches rely on additional topic knowledge when formulating prompts. Furthermore, only Khatib et al. consider a specific argumentation scheme, namely cause-to-effect, which is still too little with respect to the spectrum of 60 schemes defined in the user compendium by Walton et al. [2008]. Therefore, I want to overcome these limitations and demonstrate a method to generate scheme specific arguments without the need to manually add further topic knowledge.

2.6 Prompt Engineering

Most methods in the previous section do not rely on designing prompts but on language model fine-tuning. This thesis, however, is based on prompt engineering. The task of prompt engineering is to formulate prompts that force a language model to output some expected content and response format. For example, the prompt *"Q: Do you think movie X is good or bad? A:"* might lead a model to infer one of the classifying terms *good* or *bad*. Because most of the presented argument generation papers only sparsely consider prompt engineering, this section gives a short overview of works which address this

task in a general fashion.

Schick and Schütze [2021] present pattern-exploiting training (PET), a method to use prompts for fine-tuning. They manually define patterns which function as wrappers for an input sequence. A pattern to evaluate a restaurant review could be *"[review] It was"* for which one can expect adjective inferences like *great* or *terrible*. To compare and evaluate the outputs, a manually defined verbaliser interprets them and maps them onto a new result space, e.g. numbers from one to five. While the authors introduce ways on how to verbalise whole phrases and sentences as it would be required in this thesis, their method relies on training- and test-datasets. Therefore, outputs must be mappable onto a finite result space, which is not applicable in this thesis: A combination of one topic and one argumentation scheme is not limited to one possible argument. Nevertheless, I utilise their concept of patterns that function as templates to formulate scheme and topic dependent prompts, further explained in chapter 3.

Aina and Linzen [2021] use GPT-2 and LSTM to demonstrate how sentence prompts can unfold when they are ambiguous. By formulating multiple prompts with and without disambiguation cues, they show how simple keywords like propositions can prevent language models from derailing. Prompts are thus highly context sensitive and must be carefully crafted. This sensitivity is also studied by Reynolds and McDonell [2021] for multiple prompt formats. They define different strategies on how to optimise prompts for GPT-3 and show how topic framing and repetition help to generate better texts.

In the original work of GPT-3 by Brown et al. [2020], the authors apply so-called few-shot prompts which are prompts that include a small number of task-specific examples (in-prompt examples). For instance, to translate some text, the prompt might contain N example translations, resulting in an N-shot prompt. In their work, they compare 0-shot, 1-shot and N-shot prompts for a translation task. They come to the conclusion that adding more examples improves the inference quality. Additionally, they state that GPT-3 can learn ad-hoc during the inference generation. Reynolds and McDonell [2021] argue against this and claim that N-shot prompts rather function as a key to request task-specific model knowledge. Therefore, GPT-3 does not require task-specific examples but rather a prompt that correctly requests expected information by the user. Additionally, they argue that prompts resembling natural language are more efficient in requesting information than artificial input formats. This is due to the training data which also mostly stems from human written texts. To substantiate their claims, they reformulate the translation prompts from Brown et al. without using in-prompt examples; rather, they introduce a structure that allows humans to also understand and complete the prompts. Although their final SacreBLEU scores do not reach OpenAI's 64-shot setting,

compared to the original 0-shot, their *Master translator 0-shot* prompt as well as their *Simple colon 0-shot* prompt reach notable improvements. The authors refer to the applied optimisation techniques as prompt programming. Because this thesis focuses on manual prompt engineering, these optimisation methods are further discussed in chapter 3.

Even though prompts can be optimised without adding any ad-hoc examples, Reynolds and McDonell [2021] could not achieve GPT-3's 64-shot results. After GPT-3's release, research groups have analysed the influence of (1) the number of in-prompt examples, (2) their ordering and (3) their content. These works demonstrate that all parameters highly influence the final inference's quality. Liu et al. [2021a] show how to generate better inferences by pre-selecting the 64 most similar examples regarding the request sample and adding them to the prompt. As similarity measure they use RoBERTa's CLS embedding and calculate the Euclidean distances between all examples and the request sample. Their evaluation indicates that in-prompt examples more similar to the input reach better results. Lu et al. [2021] study the order sensitivity of in-prompt examples. They test different sizes of GPT-2 and GPT-3 and come to the conclusion that automatically selecting a permutation of examples can largely increase the inference quality for any of the given language models. Zhao et al. [2021] also focus on the order of few-shot examples. In their experiments they conclude that frequent examples and examples near the end of the prompt greatly influence the prediction. For example, in their sentiment analysis experiment with four in-prompt examples, they show that three positive examples followed by a negative one introduces bias towards a negative sentiment. With their experiments the authors derive two biases directly depending on the in-prompt examples:

1. *Majority label bias.* Example types occurring more often have a greater influence.

2. *Recency bias.* Examples at the end of the prompt have a greater influence.

Additionally, the authors observe that indefinitely adding more examples can lead to worse predictions by unbalancing the example set. Although in-prompt examples are prone to the above-mentioned biases, I also test this approach to generate arguments. But because datasets with scheme-annotated arguments are sparse, I manually craft example arguments that are representatives of Walton's schemes, detailed in the next chapter.

Chapter 3

Argument Generation

This chapter focuses on two methods of generating arguments: a fully automatic and a user assisted one. The former method relies on manually refined prompts, a set of language models, and heuristics to post-process argument inferences. These aspects are combined in a pipeline in which arguments are first generated, then automatically evaluated, filtered, and finally scored by a human expert. The latter method – user assisted argument generation – provides users with a web interface to manually select and optimise prompts which are then sent to a language model backend. The name of this web interface is AutoArg.[1]

3.1 Automatic Argument Generation

There are three main aspects to cover regarding automatic argument generation: (1) where to select a test-set of controversial topics from, (2) which language model to apply and (3) how to design prompt templates. Regarding the first aspect, IBM's lists of controversial topics can be used; for the second aspect, models from Huggingface[2] can be taken advantage of. However, prompt templates representing the argumentation schemes of Walton et al. [2008] do not yet exist. For this reason, the primary goal of my thesis is to devise prompts which can be used to infer arguments corresponding to argumentation schemes of Walton et al. Figure 3.1 provides an overview of the complete automatic argument generation pipeline, which is detailed throughout this chapter.

[1] `https://autoarg.web.webis.de`.

[2] `https://huggingface.co/` (03.05.2022).

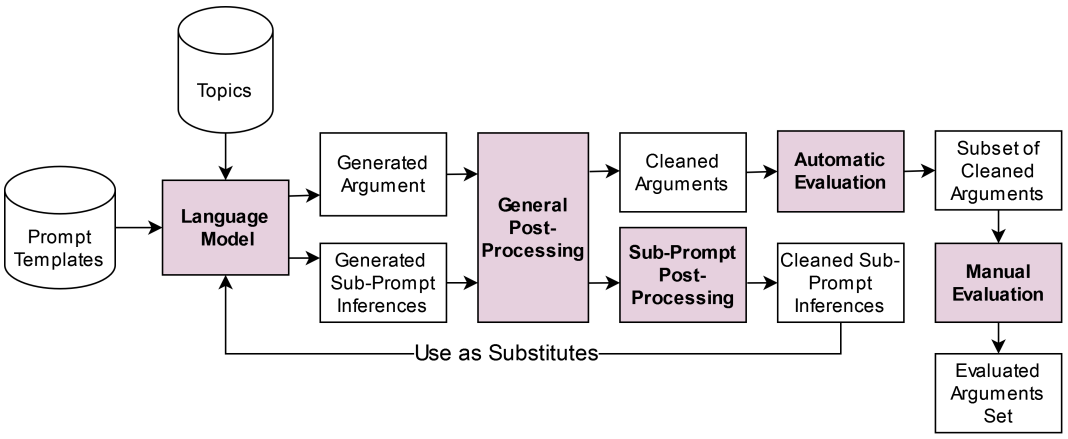


Figure 3.1: A pipeline to automatically generate arguments. Coloured boxes indicate algorithms, while white boxes indicate data outputs. In case the manual evaluation is not feasible, this pipeline can also be stopped after the automatic evaluation and filtering of cleaned arguments.

3.1.1 Controversial Topic Selection

Controversial topics must be heterogeneous and cover different domains to provide a meaningful evaluation for the argument generation system. Therefore, I rely on a subset of topics from six of the IBM debating datasets.[3] By iterating over each debating dataset, I extract 300 controversial topics of different lengths: from single-term topics like *boxing* to whole phrases like *assisted suicide should be a criminal offence*.

While all of these topics cover controversies, some of them already take the form of claims and statements which encapsulate a specific pro or contra stance. For *boxing* the direction of a hypothetical debate is not yet clear, but for *assisted suicide* a clear contra stance can be derived, since it *should be a criminal offence*. This lack of generality can limit a language model's ability to generate diverse arguments. Additionally, prompt templates depend on a topic's grammatical form. Different grammatical topic structures (e.g. a noun versus a sentence claim) lead to grammatical inconsistencies because prompt templates are too static. To overcome these limitations I choose a subset of 32 controversial topics and manually split them into main topics resembling noun phrases and verb goals[4] (if provided). With this division, prompt templates can be based on main topics and incorporate verb goals only if required. A full

[3] `https://research.ibm.com/haifa/dept/vst/debating_data.shtml` (17-02-2022).

[4] Verb goals refer to the action in a topic statement, for example "ban", "abolish" or "adopt".

list of controversial topics being used for the automatic argument generation can be found in appendix A.3.

Besides *Topic* and *Verb Goal* columns, table A.3 also includes a *Numerus* column. Because some prompt templates use verbs referring to the topic, e.g. *[topic] [to-be] ...*, it is necessary to be aware of a topic's numerus to choose the correct verb form, in this case either *is* or *are*. Even though it is possible for most topics to automatically derive whether they are plural or singular, I want to exclude this possible source of error from the evaluation steps and hence manually annotate the numerus for all 32 topics.

3.1.2 Language Models

This section introduces all language models applied to generate arguments. I specifically go into detail about their training data which is important for the formulation of prompt templates, expounded on in section 3.1.4.

Autoregressive (AR) language models have the property to automatically continue writing a prompt. Alternatively, autoencoding (AE) language models can be used to reconstruct corrupted inputs [Yang et al., 2019]. While it is possible to infer argumentative data with autoencoder models, this thesis focuses on autoregressive language models; specifically, models based on the transformer architecture by Vaswani et al. [2017]. The original transformer model is composed of an encoder and decoder. These elements are devised as separate neural networks which are concatenated by using the encoder's output as the decoder's input. Both components apply a multi-head attention approach, classifying transformers as attention-based models. Given an input vector, attention allows vector elements to be compared to each other, providing context knowledge at each position; so, attention allows each token to learn about its surrounding context. Both networks respectively end with a feed-forward layer and a normalisation to generate an output vector. Although the original transformer architecture includes both, an encoder and decoder, many models based on it either exclude encoders or decoders, e.g. the GPT-series which exclusively relies on N decoders in its architecture. But even if models like GPT omit one component, they are still referred to as transformer models.

A popular approach to improve a language model's outputs for some domain is to fine-tune it in a down-stream task. In this thesis, however, I do not apply fine-tuning. The reasoning is that I want to provide an overview of the vanilla language model's capability to generate meaningful and valid arguments. Therefore, the only way to increase the quality of arguments is by prompt engineering. To compare the argumentation capabilities of different language models, I choose seven different models, listed in table 3.1. Among

these models is GPT-3, but because I only received temporary access to it, I mainly focus on the other six open source models which I applied from the Huggingface library [Wolf et al., 2019]. Nevertheless, some GPT-3 results will be discussed in section 3.1.10.

The language model with the fewest parameters is Transformer-XL by Dai et al. [2019]. In contrast to other transformer language models, this one is not bound by a maximum inference length. It was trained on WikiText-103, a dataset of verified qualitative Wikipedia articles, and is therefore limited to natural language inputs and outputs. As second model I apply XLNet [Yang et al., 2019], which leverages advantages from AR and AE methods. As the authors call it a generalised AR model, it is primarily designed for text generation, but can also be used on AE tasks. It is based on Transformer-XL and trained on preprocessed data from book corpora, news data and websites (including Wikipedia). As stated by Wolf et al. [2019], Transfomer-XL and XLNet can produce better inferences by prepending each prompt with a short text; hence, I prepend prompts for these models with the prefix noted in Huggingface's pipeline module[5].

Four of the language models I experimented with – of which two are medium sized compared to other models, while the two others are the largest models – are part of the GPT series. The GPT series is based on the original architecture of GPT by Radford et al. [2018]. All GPT inspired models are composed of decoders and mainly differ from each other by tokenisation algorithms, training data and model size. For GPT-2 [Radford et al., 2019], the authors design a training dataset called *WebText* which is a web scraped corpus with focus on data quality. The corpus contains about eight million documents and has a size of 40Gb. A special aspect is the exclusion of Wikipedia articles.[6] The successor of GPT-2, GPT-3 by Brown et al. [2020], mainly relies on the CommonCrawl dataset[7], but also applies book corpora and Wikipedia. Their filtered training dataset contains 570Gb of text. Both GPT-2 and GPT-3 were constructed and trained by OpenAI[8]. In contrast, GPT-Neo [Black et al., 2021] and GPT-J [Wang and Komatsuzaki, 2021] were composed by the EleutherAI group and trained on The Pile [Gao et al., 2021]. The Pile is a dataset containing 800Gb of diverese texts, specially devised for language model training. The authors demonstrate that training on this corpus with GPT-2-sized models improves

[5] `https://github.com/huggingface/transformers/blob/master/examples/pytorch/text-generation/run_generation.py` (11.03.2022).

[6] The definition prompt displayed in appendix A.1 states Wikipedia. But as many websites refer to and name Wikipedia, intrinsic knowledge about this term's meaning can be assumed.

[7] `http://commoncrawl.org` (13.03.2022).

[8] `https://openai.com/` (13.03.2022).

Table 3.1: All language models I use to automatically generate arguments. Regarding *GPT-3**, I received temporary and limited access in 2021. Therefore, experiments with GPT-3 are only conducted on a drafting set of prompt templates.

Model Class	Model	Train Corpus	Parameters
Transformer-XL	Transformer-XL	WikiText-103	285,205,322
XLNet	XLNet Large Cased	Book corpora, news data, websites, Wikipedia	360,300,800
GPT-2	GPT-2 XL	WebText	1,557,611,200
GPT-Neo	GPT-Neo 2.7B	The Pile	2,651,307,520
T5	T5-3B	C4	2,851,598,336
GPT-J	GPT-J 6B	The Pile	6,050,882,784
*GPT-3**	Davinci	CommonCrawl, book corpora, Wikipedia, high quality websites	≈175,000,000,000

results in many tasks compared to the original dataset. The corpus covers diverse domains and sources like books, websites and scientific articles.

The last model is T5 by Raffel et al. [2020] which can solve different tasks by prepending the input prompt with a task-specific prefix. While this approach does enable the model to solve multiple tasks, it also limits the model in free text generation. The corpus used for training was composed by the authors and named C4 with 750Gb of cleaned data from CommonCrawl. The motivation behind using this model is that other works like Betz and Richardson [2021] emphasise the ability of T5 to create missing argument components. For this reason, I exploit the model's summation ability and enforce it to generate a text that is longer than the prompt input. I presume this method forces T5 to include its own knowledge to generate an argument out of the input prompt.

3.1.3 Decoding Mechanisms and Inference Parameters

Decoding mechanisms are a decisive factor of the quality of an inferred text. To decide which token to generate, given some input prompt with m tokens, a language model computes a probability distribution over all v tokens in its vocabulary. The distribution indicates how likely it is for each token v_i to continue the given prompt. Now the task is to decide what token v_i to choose, for which two main approaches exist:

- *Maximisation-based decoding.* Its goal is to generate a text that maximises the likelihood in equation 3.1, c.f. Holtzman et al. [2020]. To overcome the computational effort of calculating probabilities for every

text continuation, approximations are used, e.g. beam-search as applied by Vijayakumar et al. [2016].

- *Stochastic decoding.* Its goal is to generate a human-like text by introducing randomness, also referred to as sampling. Greedy sampling methods choose the succeeding token depending on the probability distribution over v, meaning any v_i can be selected – but each with a different probability.

Many approaches that focus on generating diverse texts rely on sampling because deterministic algorithms are too restrictive and mainly output generic texts (Fan et al. [2018]; Gretz et al. [2020]; Khatib et al. [2021]). Since this work also focuses on generating diverse texts in form of arguments, I apply a sampling decoding mechanism to create them. There is only one exception described in section 3.1.9.

$$P(x_{1:m+1}) = \prod_{i=1}^{m+1} P(x_i|x_1...x_{i-1}) \tag{3.1}$$

Even though this work is limited to sampling, there are still multiple sampling strategies to choose from, for example top-k [Fan et al., 2018]. Regarding the default greedy sampling, every token could be generated at every position, even unlikely ones. To exclude these unlikely tokens, top-k sampling limits the token distribution to the top-k tokens with the highest probabilities. Another often used parameter is temperature t with $t \geq 0$ (Gretz et al. [2020]; Khatib et al. [2021]), which is used to sharpen the distribution. A high temperature increases probabilities of likely tokens and lowers them for unlikely ones. Setting $t = 0$ defaults to greedy sampling. While top-k in combination with temperature is often applied, a more recent and promising sampling strategy is *nucleus sampling* or top-p sampling [Holtzman et al., 2020]. Holtzman et al. show that the value k in top-k sampling must be high to generate human-like text. However, a high value simultaneously tends to generate incoherent results. Additionally, for each token to be generated, the probability distribution changes its form and becomes sharper (a few tokens with high probabilities) or flatter (many tokens with similar probabilities). The parameter k is static and unable to consider these distribution changes. Holtzman et al. also emphasise that applying top-k in combination with temperature can decrease text diversity. To overcome these problems, the authors introduce nucleus sampling.

Given some static value $p \in (0, 1]$, for each inference step, only the highest probability tokens whose summed probability does not exceed p are considered. Therefore, setting $p = 1$ is analogous to greedy sampling. Nucleus sampling enables the decoding procedure to adapt to shifting probability distributions

by automatically deciding on a k value for each generated token. To choose an optimal p, Holtzman et al. [2020] conduct multiple experiments to find out for which p the most human-like texts are generated and demonstrate that they receive the best texts regarding their metrics for values of $p \in [0.9, 1.0)$. They additionally show that nucleus sampling does not tend to generate repetitive texts, which is a main issue for other stochastic sampling methods.[9] Given these results, I apply nucleus sampling with $p = 0.95$ as decoding algorithm.

After deciding on the decoding strategy, there are two more aspects to consider: How many tokens and how many inferences should be generated for each prompt. The second question is important as stochastic sampling generates different inferences for the same prompt every time it is used. Therefore, I generate five inferences per combination of model, prompt and topic. This allows me to choose the best inference and also gives the language models leeway to derail in some cases. As for the token count, arguments could in theory be arbitrarily long. But instead of allowing the models to generate maximum-length texts, I pre-define the inference length and set it to 100 tokens for every argument-prompt and model, which is the length also used by Khatib et al. [2021]. Yet, here this length merely functions as placeholder: Khatib et al. consider the cause-to-effect argumentation scheme, but other schemes might require individual amounts of information. Since for now too few datasets based on Walton et al. [2008] exist from where optimal (or minimal required) argument lengths could be statistically derived from, the question of the best inference length setting per scheme remains an open task, discussed in section 5.1.

3.1.4 Prompt Engineering Strategies

The NLP realm has faced many changes in recent years, one being the focus shift from newly trained language models to pre-trained and fine-tuned ones. These approaches leverage pre-trained model knowledge by only updating a sub-set of model parameters in a downstream task, called fine-tuning. Nevertheless, with the publication of GPT-3 by Brown et al. [2020], a new trend emerged: prompt engineering. It is a method which does not focus on updating a language model's parameters, but on formulating an optimal input prompt to systematically access the model's intrinsic knowledge base.

[9] In appendix C.3 I conduct an experiment to analyse the repetition behaviour of each language model when nucleus sampling is applied. In the original work, nucleus sampling is only tested with GPT-2, leading to the question whether its behaviour generalises over every model. The results show that for most language models nucleus sampling by itself is enough to hinder language models from derailing into repetition loops.

Liu et al. [2021b] define prompt engineering as process of finding a prompting function $f_{prompt}(x)$ that generates qualitative results on a textual downstream task. To automatically conduct argument generation with prompting, one first must formulate a prompt. In case of arguments representing the cause-to-effect scheme, an argument prompt could be formulated as $p =$ *[x] lead to [z]*. To complete this prompt, [x] can be replaced by any controversial topic, for instance:

$$p_{compl} = f_p(\text{Abortions}) = \text{Abortions lead to [z]}$$

Now, [z] can be replaced by any word sequence that is coherent to the prefixed *cause*-prompt, which in this case results in the missing *effect*. An example prediction for z, generated by GPT-Neo, is:

$$LM_{Neo}(p_{compl}) = \text{Abortions lead to depression, according to new study.}$$

In their survey, Liu et al. [2021b] distinguish between two prompt shapes: cloze prompts and prefix prompts. The *cause-to-effect* prompt defined above exemplifies a prefix prompt. This shape is characterised by a [z] at the end of the prompt, typically inferred by an auto-regressive language model trained in a left-to-right fashion. In contrast, [z] in cloze prompts can be at any position, often inferred by a masked language model like BART [Lewis et al., 2020] and limited to an inference with one or a few tokens. In this thesis I only focus on prefix prompts. While prompts could also incorporate multiple $[z_i]$ and thus be inferred by different models at the same time, I bind each generated argument to one model.[10] In such manner, the capability of each language model can be evaluated separately.

To formalise the prompt engineering task, I search for a set of prompt templates $\mathcal{P}$ with each prompt being based on one argumentation scheme of Walton et al. [2008] s_i:

$$\mathcal{P} = \{p^{s_1}, p^{s_2}, ..., p^{s_n}\} \tag{3.2}$$

Prompts in $\mathcal{P}$ can require an arbitrary amount of substitute inputs x_i, with the controversial topic being mandatory. The prompting function for this task looks as follows:

$$p^{s_i}_{compl} = f_{p^{s_i}}(x_{topic}, x_2, ..., x_m) \tag{3.3}$$

The complete prompt now can be used as model input, resulting in an inference I call *generated argument*. Each prompt in $\mathcal{P}$ must be constrained in a way to generate an argument for its respective scheme s_i. So, each generated argument is expected to be part of $\mathcal{A}_{s_i,x_{topic}}$, the indefinite set of all arguments for this topic and scheme combination. With $\mathcal{A}_{s_i,x_{topic}}$ being indefinite, each prompt

[10] T5 is an exception, explained in section 3.1.9.

must be capable of generating various topic- and scheme-relevant arguments. In order to realise this I use probabilistic nucleus sampling which can generate different inferences for one prompt, c.f. section 3.1.3. Now, the only remaining question is how to design prompts in $\mathcal{P}$.

Liu et al. [2021b] describe two approaches: manual template engineering and automated template learning. Current research focuses on automatic template learning which is due to the manual effort and experience needed to manually formulate prompts (Liu et al. [2021b]; Shin et al. [2020]). Although these works demonstrate strategies that lead to a significant improvement over manually crafted prompts, they either require annotated data to fine-tune prompts or they are unsuitable for an open text generation task. Popular examples are sentiment analysis and review analysis, for which an abundance of existing training data exists. This is not the case for arguments based on argumentation schemes from Walton et al. [2008]. Most argumentation corpora are limited to a small number of schemes [Khatib et al., 2021] or use alternative argumentation typologies, c.f. Araucaria[11]. Only a few corpora exist which explicitly focus on Walton's schemes: The corpus of Visser et al. [2021] entails 505 argumentative relations from the 2016 US election. Another one is the ReCAP corpus by Dumani et al. [2021]. It is composed of German education politics and includes about 2,500 premises/conclusions spread over 100 argumentation graphs. Although the number of arguments from both corpora seem sufficient, they are distributed over 60 schemes and highly unbalanced. The corpus of Visser et al., for instance, consists of 81 *Arguments from Example*, while ten of the other argumentation schemes each only have one representative. Nevertheless, automated template methods also entail approaches not requiring training data, for example paraphrasing [Liu et al., 2021b]. Yet, changing a prefix prompt by paraphrasing it simultaneously can alter the expected argument's format and hinder post-processing. Schemes by Walton et al. [2008] are too complex to be manipulated by a misleading paraphrasing.[12] Consequently, I choose manual template engineering to fashion this work's prompts.

Manually crafted prompts are not only independent of any training data but can also be formulated in a generalised and unbiased way. While for each argumentation scheme an indefinite amount of possible arguments exist, fine-tuned automatic prompts would be restricted by training data which limits the generation diversity. On the other hand, too generalised and imprecise formulations of manual prompts introduce ambiguity and can easily derail

[11] `https://arg-tech.org/index.php/the-araucaria-database/` (08.03.2022).

[12] While the generated text of an paraphrased prompt can still resemble an argument, it might no longer resemble the original argumentation scheme. Schemes by Walton et al. [2008] are too fine-grained and partly similar to each other, which can easily confuse a language model.

inferences. For this reason, many works refer to experts with domain and model knowledge who manually devise prompts, e.g. Schick and Schütze [2021]. But instead of testing prompts in a trial-and-error fashion, recent research also focuses on generally applicable strategies and rules on how to design stable prompts. Reynolds and McDonell [2021] provide a comprehensive study on how to manipulate GPT-3 into generating purposeful inferences with 0-shot and N-shot prompts. Even though their experiments are limited to GPT-3, they assess that their approaches generalise to any auto-regressive language model, hence, I base my prompts on their strategies. Reynolds and McDonell describe them as programming in natural language and thus title them *Prompt Programming*. Their strategies are:

1. *Use Signifiers.* They function as keys to access the intended language model's behaviour. These signifiers describe the task to be solved, like "translate" or "debate".

2. *Add Redundancy.* Repeat your topic and key-words as often as possible to prevent the model from derailing. The clearer the intentions are, the less likely it is that the model breaks pattern.

3. *Syntactically Constrain the Model.* Use expressions or punctuation which syntactically force the model to generate specific content, for example colons or quotation marks. A prompt to generate an explanation can be improved by adding a colon, exemplified by Reynolds and McDonell in their translation experiment.

4. *Apply Corpus Language.* A prompt's language and structure should resemble the model's training corpus and only include tokens the model understands. The GPT series applies a byte-pair encoding and therefore can encode any text sequence. However, Transformer-XL uses a word-wise tokeniser and tags unknown words as <unk>. The authors also mention the following aspects as essential to keep in mind when writing prompts: tone, implications, association, meme, style, plausibility and ambiguity.

5. *Provide Demonstrations.* Demonstrations or in-prompt examples can be used to show the language model what form of output is expected by relying on the model's ability to understand analogical cases. Liu et al. [2021b] name this approach *Prompt Augmentation.* While in-prompt examples can easily be collected, appended and treated as prompts, they introduce bias depending on the examples and their ordering [Zhao et al., 2021]. Also, it must be clear to the model when one example ends and

a new one begins, which can be done by using line breaks or other syntactical constrains.

6. *Introduce a Proxy.* Ask a famous person or entity in your prompt to solve a task. For example, one might ask a philosopher about the meaning of life or an environmental expert about climate change. Proxies add (cultural) bias, which can be leveraged by using multiple proxies and comparing their results.

7. *Stage a Dialogue.* A Q&A dialogue is a special proxy where two or more people converse in an artificial setting. Depending on the interlocutors, the conversation can be directed in a preferred direction. For example, if one wants to infer an explanation, a dialogue between teacher and student is adequate.

8. *Split the task into sub-tasks.* In case a task is too complex for a single prompt to solve, force the model into solving it stepwise. For a mathematical problem, add the prefix "Solve the following mathematical problem stepwise." Alternatively, split the task into multiple prompts which are inferred individually. Afterwards, their results can be combined as a *prompt composition* [Liu et al., 2021b]. Reynolds and McDonell also propose a way to fill masked sections in a prompt without additional prompts by relying on the model's conditional probability.[13]

9. *Define a Metaprompt.* Metaprompts are prompts functioning as wrappers. They are a special form of prompts with the purpose of establishing an artificial setting which can include every prompt programming method above. For example, one can formulate a whole scene with dialogues and descriptions to direct the language model in some direction.

Reynolds and McDonell argue that by using these methods, language models become less likely to break pattern and able to solve complex tasks. To analyse the efficiency of these strategies, I formulate five types of prompts with each type integrating different prompt programming strategies as rule sets.

[13]The prompt (p), a substitute inference (i) and everything that comes after both (s) are concatenated and evaluated by the language model LM which returns a conditional probability: $LM_{cond}(p \cdot i \cdot s)$. Doing this process with varying lengths of i bears different probabilities. The substitute section with the highest probability can be used as the final substitute. While the work of Reynolds and McDonell only briefly touches this idea, in a blog they provide code examples for a GPT-3 implementation: `https://generative.ink/posts/parsing-by-counterfactual/#code` (09.03.2022).

3.1.5 Prompt Types

I define five prompt types based on the work of Reynolds and McDonell [2021] to analyse and compare the potential of their prompt programming strategies. Each type is characterised by certain required elements and can be used as a set of rules to formulate prompts for any argumentation scheme. Regarding the evaluation section 4.2, this approach allows me to compare different prompt programming strategies over all argumentation schemes. The five prompt types are: *short*, *dialogue*, *descriptive*, *demonstrative* and *meta*.

The short prompt type utilises a minimum amount of context. Besides the controversial topic, this type requires a signifier sequence to hint at the desired argumentation scheme. While this type is prone to derailing, it is at the same time less restricted and thus can generate novel and creative arguments. The following prompt template represents an instance of the short type for *Arguments from Cause to Effect*:

[topic] [to-lead] to

This example not only contains a blank (input field) for the controversial topic, but also a verb blank [to-lead]. Each prompt template can have any number of input fields, but while [topic] is obligatory, other fields are not required. [topic] and [to-lead] both fall into the category of *static blanks*. This means no additional information retrieval method is needed to replace the blanks with matching substitutes. Topics are predefined, as explained in section 3.1.1; verbs referring to these topics are also known once the topic's numerus is specified. For example: By replacing [topic] with *abortions* (plural), [to-lead] can derive the correct form and be replaced by *lead*. The reasoning behind such fine-grained templates is to provide the model with text that is as similar to its original training data as possible, and to minimise inference errors introduced by grammatically incorrect prompts.

The descriptive type adds broader context to the topic and wraps it into the setting of an article. Thus, this type can be interpreted as small metaprompt with additional signifiers and redundancy to constrain prompt templates. The respective template for *Arguments from Cause to Effect* has the following form:

> The topic of this article [to-be] [topic].
> [Topic] [to-be] defined as follows: [[definition]]
> [Topic] [to-have] many different influences on our society, economy and policy.
> [Topic] [to-lead] to

As mentioned before, [topic] and verbs are static. [[definition]], however, introduces a new substitute type: *dynamic blanks*. Dynamic blanks require addi-

tional knowledge about the current topic which can be acquired with an information retrieval system. Although other works rely on automatically querying numerous websites or specifically Wikipedia to acquire information, e.g. Gretz et al. [2019], I focus on retrieving information exclusively by querying language models. This method is inspired by the prompt programming strategy to split tasks into sub-tasks (Liu et al. [2021b], Reynolds and McDonell [2021]). One benefit of this method is that language models can easily find multiple substitutes for one dynamic blank. In contrast, web retrieval methods might have to query different sources for this task. In section 3.1.6 I go into more detail about dynamic blanks and their substitute generation.

Dialogue prompts constitute the third type. This type is more limited than descriptive prompts since dialogues require a static set of interlocutors. While these people can be indicated by uppercase letters like A and B or Q (question) and A (answer), this work's dialogue type is limited to a student-teacher setting with the teacher providing an argumentative inference. The intention behind this setting is to generate a well-written and verbose answer – as one might expect from a teacher. The dialogue prompt template for *Arguments from Cause to Effect* is defined as follows:

> Teacher: Today we want to talk about [topic].
> Student: What [to-be] [topic]?
> Teacher: [[definition]]
> Student: What influence [to-do] [topic] have on our society, economy or policy?
> Teacher: [Topic] [to-lead] to

The fourth type comprises demonstrative prompts. These prompts do not depend on any dynamic blanks/substitutes and can be applied immediately. They are based on the ability of language models to learn from analogies and therefore require argument examples of the same argumentation scheme I prefix with "Example n". For each demonstrative prompt I choose three examples. However, as seen in other works (Zhao et al. [2021]; Lu et al. [2021]), the ordering and selection of examples adds bias and influence the result's quality. Consequently, manually choosing the examples and their order can manipulate the output of an argument regarding its stance, which is an important consideration. To clarify what kind of examples each prompt template contains, argument examples are preceded by an introductory sentence as signifier. The demonstrative prompt template for *Arguments from Cause to Effect* has the following structure:

> Causes and effects are used to express causal generalizations.

> Example 1: Not studying before an exam leads to worse grades. Therefore, you should study.
> Example 2: Smoking cigarettes causes lung cancer. Therefore, you should not smoke cigarettes.
> Example 3: Owning a cat increases personal happiness. Therefore, you should own a cat.
> Example 4: [topic]

Meta prompts constitute the final type which I directly derive from one prompt of Reynolds and McDonell [2021][14] with slight changes to make clear that the language model is supposed to generate an argument. This metaprompt wrapper is universal for all argumentation schemes.[15] To indicate which specific argumentation scheme it must resemble, I append the respective short type template as beginning of the argument. The meta type for *Arguments from Cause to Effect* looks as follows:

> *"What argument can I use in a debate if I want to argue about [topic]?" I entered my question into the Argument Generator and waited. The Argument Generator will render a simulation of a debate to answer my question. The argument can be any argument as long as it is relevant for [topic]; the machine will find the argument most suited to be used in a debate about [topic]. For this question in particular, the argument must be relevant, argumentative, coherent and plausible. And of course it must be related to [topic]. The Argument Generator beeped, indicating that it has found the most suited argument. The argument displayed on the screen: "*[topic] [to-lead] to

The italic text indicates the meta prompt section and everything after the final quotation mark is part of the short prompt template. All prompt templates for each type[16] can be found in appendix table A.2. Because some schemes are too specific to be used in a generalised context (e.g. *Argument from Witness Testimony*) or require a large amount of information to be inferred (e.g. *Epistemic Argument from Ignorance*), my compendium currently includes templates for a sub-set of only 22 argumentation schemes.

[14] See figure 6 in their work.

[15] The meta prompt prefix contains special opening and closing quotation marks (U+201C, U+201D). The tokenisers from XLNet, TransformerXL and T5 do not know these characters and interpret them as unknown token. To circumvent this issue I replace the quotation characters for these tokenisers with the default keyboard quotation " (U+0022).

[16] Excluding meta prompts because they all have the same structure and depend on the listed short type prompts.

3.1.6 Dynamic Blanks and Substitutes

Argumentation schemes by Walton et al. [2008] each comprise between one and five premises plus a conclusion. The amount of information needed to instruct a language model to formulate valid arguments related to the topic with complex schemes is too much to be put into a simple prompt. This is one of the reasons why I constrain the number of schemes mapped onto prompt-templates to 22. However, I do not exclude every scheme which is difficult to represent as prompt because of its complexity. One way to overcome the complexity limitation is by using signifiers and sub-tasks. Theoretically, it is possible to manipulate the language model into formulating a verbose and comprehensive argument by adding a signifier like "generate an argument with all premises and a conclusion". However, this approach has two downsides:

1. The expected information for each premise must be specified beforehand, otherwise the model might omit enthymemes and generate texts that do not resemble the scheme or not even an argument. This results in long and non-dynamic prompts.

2. Depending on the length, structure and complexity of the task signifier, language models tend to misunderstand or forget. Argument generation requires a compound task formulation which encourages misunderstandings in small language models.

Argument from Expert Opinion is one scheme that requires additional task information: Using one prompt, the model not only has to generate an expert-name, but also an argument stated by this expert. Following the scheme definition, it is in theory not required to include a certain expert; however, doing so simplifies the differentiation between this scheme and the *Argument from Position to Know* scheme. To circumvent the above-mentioned issues of a single prompt approach, I apply a multi prompt strategy [Liu et al., 2021b]: I formulate sub-prompt templates (based on prompt types) which infer information pieces that function as substitutes for other prompt templates. Combining the generated information creates composite prompts. In this work I apply sub-prompts to generate three categories of information: topic experts, a topic definition and people in a position to know about the topic. All sub-prompt templates for each category are listed in appendix A.1. In this thesis these templates are limited to three prompt types: *short*, *descriptive* and *dialogue*.

Alternatively, standard information retrieval tools could be used to query topic-relevant information, e.g. definitions from Wikipedia. Yet, querying information from language models has two advantages:

1. There is no need for an additional algorithm. One only must devise sub-prompts and possibly a post-processing method. Post-processing can be

Table 3.2: Two information dimensions for sub-prompt categories. The dimensions and classification of the sub-prompt categories are by no means absolute but open to debate and task dependent. For example, if the task is to find all definitions for one topic, then "definition" would be a multiple choice information.

	Single Choice	Multiple Choice
Factual	definition	expert-name
Plausible	/	position-to- know

simplified by using cleverly designed prompts. For example, one can force sub-prompts to generate a specific format like an ordered list from which N items can be automatically extracted and then further processed.

2. Not all base-knowledge can be factually backed up with standard information retrieval. For instance, finding someone in a position to know about a specific topic often leads to the question of *plausible* or *rather unlikely* instead of *true* or *false*. This type of information can easily be generated with language models.

The three aforementioned sub-prompt categories demonstrate two dimensions of information: factuality versus plausibility and single- versus multiple-choice. The latter refers to how many substitutes can be found to replace one blank: *Definition* blanks expect one single substitute,[17] while for *expert-names* and *positions-to-know* multiple substitutes are conceivable. The difference between expert-names and positions-to-know is that names can factually be proven to be true or false, while positions-to-know are open to debate and a question of plausibility.[18] All information dimensions are visualised in table 3.2. Although these dimensions might not be complete and remain open to debate, they can be applied as helping tool to decide on what decoding mechanism to use for each sub-prompt category, c.f. 3.1.3, which I expound on in the following section.

3.1.7 Substitute Decoding

In section 3.1.3, I decide on inference settings to generate arguments. These settings change slightly for substitute generations since the maximum argument length by Khatib et al. [2021] does not apply for substitutes. 100 tokens

[17] Omitting cases of ambiguity or terms with multiple definitions like "art".

[18] As for definitions, this is a simplification. There might be ambiguous names or "experts" people claim to be no *real* experts.

might be too short, omitting relevant information, or too long, uselessly wasting resources and time. Because substitute categories depend on the argument schemes[19] and thus are highly variable, corpora to base the inference length on may not exist. To overcome this issue, I demonstrate an automatic and corpus-less way of finding optimal token lengths for any substitute category.

For each substitute category I first generate inferences with a manually chosen maximum length of 200 tokens. This length should be chosen so that the relevant information is located in the resulting inference with high probability. Naturally, this number depends on the model's capability, the specific substitutes and where one assumes the relevant information to be. Additionally, I increase the number of generated inferences per prompt from five to twenty which allows to even out variances, and which is computationally faster than adding other prompts or models. These adjustments can be combined with a sub-set of topics, models and prompt templates with the goal of composing an artificial substitute corpus. I use these settings on all topics, short sub-prompt types and GPT-Neo. The short prompt type has the advantage of including a minimum amount of tokens and thus needing less computational resources; additionally, this type does not require dynamic substitutes. I also restrict inferences in the artificial corpus to GPT-Neo which resembles the middle ground between the largest open-source model GPT-J and the smallest one Transformer-XL.

Including all 32 topics, these settings result in $32 \cdot 3 \cdot 20 = 1920$ substitute inferences. After the post-processing step, which extracts the relevant information from each inference and is elaborated in the following section, I determine for each unprocessed inference the number of tokens required to completely entail the processed information. The resulting positional distributions are visualised in figure 3.2. Although one can set the inference length for each substitute category to the respective average or median, I set it to the maximum occurring length. This way,I ensure that most sub-prompt inferences include the required information, omitting extreme outliers. Accordingly, I define the maximum number of generated tokens to 44, 47, and 54 for positions-to-know, expert-names, and definitions respectively.

Regarding the decoding method, I keep nucleus sampling for all sub-prompt categories with the exception of definitions. The previously discussed information dimensions, c.f. table 3.2, can be used in combination with insights from works using language models as fact checkers (Lee et al. [2020b], Liu et al. [2021c]). Especially for definitions and expert-names it is essential that the language models generate factual texts. However, both referred works rely on external evidence datasets which are not given in this work. Lee et al. [2020a]

[19]For instance, expert-names are only relevant because of the *Argument from Expert Opinion* scheme.

provide an alternative approach without evidence datasets: The authors analyse perplexity in the context of misinformation and find out that texts with high perplexity tend to contain more misinformation. Perplexity is a metric to measure text plausibility. The lower the perplexity value, the more likely the text is expected. The perplexity of a word-sequence $w_1 w_2 ... w_n$ is calculated as follows [Jurafsky et al., 2009]:

$$PP(W) = P(w_1 w_2 ... w_n)^{-\frac{1}{N}} = \prod_i^N \left(\frac{1}{P(w_i | w_1 ... w_{i-1})} \right)^{\frac{1}{N}} \tag{3.4}$$

The perplexity equation calculates a sequence likelihood in its denominator. Referring back to section 3.1.3, maximisation-based decoding mechanisms have the goal of finding a token sequence with a maximum likelihood. High-probability sequences consequently reach a small perplexity score which is desirable for factual sub-prompt inferences like definitions and expert-names. Therefore, a maximum likelihood method like beam search should be preferred over sampling techniques to generate less misinformation. However, beam search introduced two new issues:

1. Different inferences for the same prompt are highly similar and generic.
2. Beam search struggles with repetition [Holtzman et al., 2020].

Using beam search to generate different expert-names with one prompt and topic is difficult. Therefore, I only apply beam search for the definition category. Furthermore, to overcome the repetition issue, I add a generation penalty based on the work of Paulus et al. [2017], which forces the decoding procedure to not generate the same tri-gram multiple times.

3.1.8 Post-Processing

Post-processing is a relevant step to extract information from sub-prompt inferences and to exclude irrelevant sequences from the generated arguments. All inferences are first cleaned depending on their prompt type. Then, in case the inference comes from a sub-prompt, relevant information is extracted heuristically. I name the post-processing results *cleaned inferences*.

While it is possible to keep the whole generated text and its corresponding prompt, not every part is necessary. For this reason I manually define a starting position for cleaned inferences in each prompt template, denoted with $\odot$ in A.1 and A.2. Doing so avoids adding unnecessary left-sided information to an argument. The final length of a cleaned inference depends on the prompt type. Although the text generation length for all arguments is set to 100,

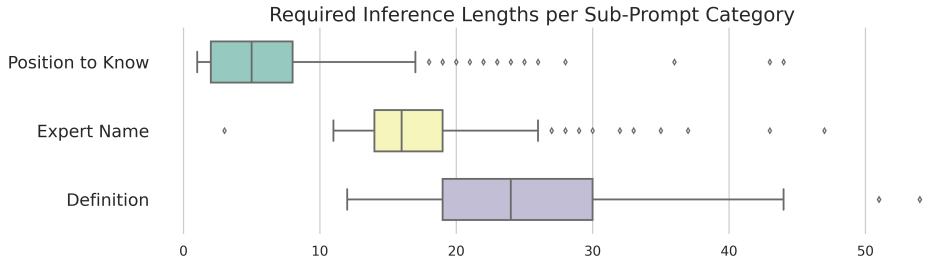


Figure 3.2: Required number of generated tokens per inference to successfully extract the relevant substitute. The upper bound for this test is set to 200 tokens and the applied language model is GPT-Neo. For each of the three categories only inferences with valid results are considered. So for instance, if no valid expert-name occurs in an inference, it is dropped. Definitions and positions-to-know include 640, and expert-names 597 valid results. The minimum and maximum position values are (1; 44), (3; 47) and (12; 54) for positions-to-know, expert-names and definitions respectively.

some prompt types include signifiers that indicate where the generated text should be cut:

- *Meta Prompts.* An opening quotation mark is added in-between the meta prompt prefix and the short prompt. The inference is cut at the first generated closing quotation mark.
- *Dialogue Prompts.* The inference includes everything the teacher says in the first generated utterance. It is cut once the student starts talking, denoted with "Student:".
- *Demonstrative Prompts.* The inference contains the whole fourth example. It is cut once the fifth example commences.

In case no signifier is generated that indicates the upper bound, the whole inference is used. The same holds true for short and descriptive prompts which lack an upper bound signifier. This setting inevitably leads to arguments of different lengths as seen in figure 3.3 and consequently influences the evaluation scores, detailed in section 4.2.

After this general cleaning process, sub-prompts are further processed. Their inferences must be valid replacements for blanks and therefore correctly formatted. To ensure valid formats, I define individual heuristics for each sub-prompt category: From definition inferences I only select the first generated sentence based on Gretz et al. [2019] who use the first sentence from Wikipedia to frame each topic. Regarding expert-names, I rely on Named Entity Recognition (NER) to extract the first name that appears in the result's

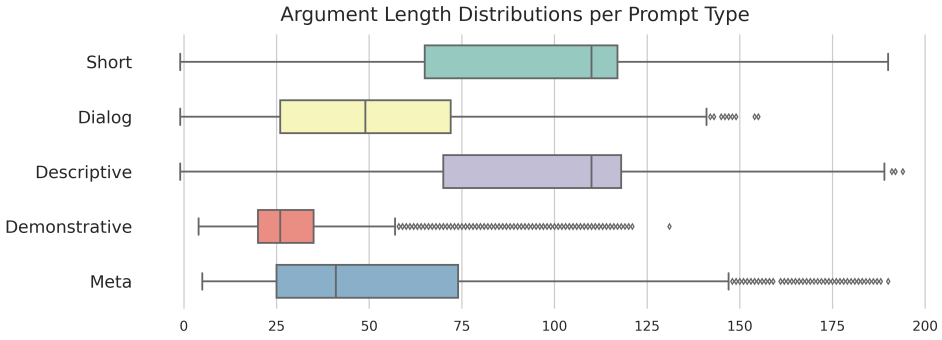


Figure 3.3: Length distribution by tokens for each prompt type. As can be seen, arguments based on the short and descriptive prompt type are in general longer than their counterparts. This can be explained by the lack of an upper bound signifier.

first sentence. For this task I apply the python library spaCy by Honnibal and Montani [2017].[20] Because positions-to-know typically do not fall into the category of named entities, I construct their prompts like "people in a position to know *are*". Inferences from these prompts can then be converted with the dependency parser from spaCy. Now, word chains directly depending on *are* can be considered potential candidates for positions-to-know. A sample set of raw sub-prompt inferences for all three categories and their extracted information is presented in appendix C.2.

3.1.9 Sub-Prompt Substitute Selection

For each sub-prompt template, topic and language model combination, I generate five inferences. To limit the number of possible substitutes per dynamic blank, I only apply substitutes from the same language model per prompt and likewise use the respective model for the final argument inference. This way it is easier to compare the capabilities of different models.[21] Still, depending on the number of prompt types per sub-prompt category, various generated texts can act as possible substitutes per blank. This could be desirable for multiple-choice information, however, I restrict each prompt to one set of dynamic substitutes per model, meaning even if different positions-to-know exist, I only choose one. This method limits the amount of generated arguments.

[20] All following methods that rely on this library are based on the *en_core_web_lg* corpus.

[21] T5 cannot generate substitutes because it is not possible to predict if and where the substitute in a T5 summary-inference occurs. For this reason, dynamic blanks in T5-inferred argument prompts are replaced with GPT-J substitutes. Since GPT-J is the largest open-source language model in this work, I presume it generates high-quality texts.

Otherwise, all combinations of dynamic substitutes would have to be considered, leading to an abundance of arguments. Another advantage of the method is that it ca be used to exclude low-quality substitutes.

Regarding definition inferences, I simply choose the substitute with the lowest perplexity. For multiple-choice sub-prompts like expert-names it is more complex: One way is to calculate the perplexity of an expert-name in the sub-prompt inference, similar to Lee et al. [2020a]. However, because the name might occur in the middle of an inference, its perplexity depends on the generated context. Another way would be to choose some similarity score given the name and the topic. I test this approach with spaCy's Word2Vec and sentence-transformers' sentence-BERT (sBERT) model by Reimers and Gurevych [2019] to calculate semantic similarity scores. Word2Vec is a method to represent words in a vector space, introduced by Mikolov et al. [2013]. Semantically similar word vectors reach higher similarity scores, for example by applying the cosine-similarity. An alternative approach is sBERT, a transformer model established by fine-tuning BERT in a siamese network architecture. Siamese networks describe a network architecture family with the goal of calculating similarities between inputs [Schroff et al., 2015]. sBERT is specifically designed siamese model to calculate semantic similarity scores between sentences as fast as possible.

Now, to find out whether the similarity scores correlate with the validity of an expert, I backtrack who the inferred experts are for three topics, c.f. appendix C.3. The results emphasise that higher similarity scores for both metrics do not imply that an expert is more legitimate or even exists. Consequently, I choose a random expert for each dynamic blank. For positions-to-know, I calculate the content similarity between the results and topic by using spaCy's Word2Vec calculator. Positions-to-know that contain words similar to the topic reach higher scores. Additionally, I choose a maximum threshold of seven tokens per position-to-know. In case all results have more than seven tokens, I select the shortest. This prevents the model from focussing too much on the substitute.

3.1.10 GPT-3 Generated Arguments

In 2021 I received temporary and resource limited access to OpenAI's GPT-3 API[22] with which I could generate a maximum of 1,722 arguments. However, these arguments are based on a previous version of prompt templates, topics and inference settings, which makes them partly incomparable to the other models' newly generated arguments. To minimise the difference between these two argument sets, I drop all GPT-3 arguments that (1) include a topic

[22] `https://openai.com/api/` (05.05.2022).

that is no longer present in the final corpus or (2) represent an argumentation scheme my implementation no longer supports.[23] After applying these restrictions, 1,312 arguments over 16 topics and 20 schemes are left. One reason for this small number is that the prompt types *demonstrative* and *meta* are not represented by this model as these types were added later during this work. Another reason is that I could only generate one inference per prompt and not five. Thus, compared to other models like GPT-J with 17,234 generated arguments, GPT-3 has no leeway for low-quality arguments which could be excluded during the automatic evaluation step. Therefore, I exclude GPT-3 arguments from the manual evaluation but keep them for the automatic one in section 4.2 to get an idea of GPT-3's argumentation capabilities.

3.2 AutoArg

The previous section demonstrated how to automatically create arguments with a generation pipeline given the predefined prompt templates, language models and controversial topics. This pipeline makes it possible to generate countless scheme-dependent arguments without human interference. A drawback of this, however, is that the arguments' quality can only be assured by the automatic metrics and sub-prompt selections. To better ensure a high quality of generated arguments I present AutoArg: A hybrid web tool with which users can control a language model's settings, choose sub-prompt inferences, freely select controversial topics, and reset as well as recreate arguments until a satisfactory result appears. With this tool, users are no longer restricted by a predefined set of topics and model settings. Additionally, users can rewrite prompts to test whether other formulations lead to better arguments. While many research teams focus on automatic prompt engineering, this hybrid approach may be a plausible alternative for complex language modelling tasks, given the employed interface is easy to learn.

In this section, I first detail the AutoArg interface structure and which components to use to simulate each part of the generation pipeline. Finally, I elaborate on some limitations of the web application.[24]

3.2.1 User Interface

AutoArg's layout is inspired by GPT-3's online playground, see figure B.1. On the right side of the screen, the playground provides a list of inference

[23] The set of topics changed during the research. Additionally, some argumentation schemes were omitted for being too complex.

[24] If you want to experiment with AutoArg, please send an e-mail to the author to receive your access data, see `https://autoarg.web.webis.de/`.

settings, for example a *top-p* parameter, and on the left side an output field to stepwise print and manipulate inferences. It is a simple interface since its only goal is to let users formulate prompts and generate inferences. For my proposed argument generation task, additional user inputs are required to manually select topics, schemes and prompt types, resulting in a more complex interface. The AutoArg generation GUI is visualised in figure B.4.

The top left input field allows users to input any controversial topic. While prompt templates are designed to accept any noun group topic – for example *zoos*, *abortions*, ... – users are not limited in any functional way. However, this might lead to grammatical inconsistencies in the prompt and lower the output quality. To ensure that the correct verb forms are used depending on the topic, a button beneath the topic field can be clicked to toggle between plural and singular forms. The next two fields to the right are used to freely define pro or contra verb goals, for instance *allow* or *ban*. Some argumentation schemes, like *Argument from Danger Appeal*, require a topic relevant action which then is replaced by the user's choice. The distinction between pro and contra is needed for special scheme templates like *Argument from Circumstential ad Hominem*, where users have to define an action and an opposing action. In case a prompt template only needs a substitute for a single [verb-goal], the verb goal contra field defaults to this one, meaning users can input any verb independent of its stance.

With the next two fields users can select an argumentation scheme and a prompt type which are mapped onto exactly one prompt template as listed in table A.2. The top right input field allows users to choose one of three language models which is then applied for the argument generation. Beneath this field is a column of different inference settings and decoding strategies, similar to GPT-3's playground. *Response Length* defines the number of tokens to generate, which by default is set to 10. The next three fields consist of sampling parameters explained in section 3.1.3. To turn these parameters off, set their value to 0 respectively. By excluding all parameters, the decoding mechanism defaults to greedy sampling. Regarding the last field, named *No Repeat N-Gram*, this setting can be turned on by choosing a value of $N > 0$. N specifies which N-gram length is not allowed to occur multiple times in the output, which means smaller numbers of N are more restrictive and lead to less repetition [Paulus et al., 2017]. This setting is also applied during the definition generation in the fully automatic approach, see section 3.1.7.

The only missing component users must be able to generate are substitutes for dynamic blanks, which can be done by pressing the *+ {Dependency}* button in the center of the interface. Doing so opens a new window as seen in figure B.2. This sub-window gives users the possibility to select a sub-prompt category from {*definition*, *expert name*, *position to know*}, one of three prompt

Table 3.3: Language Models utilised in AutoArg and their size by the number of parameters.

Model Class	Model	#Parameters
GPTNeo	GPT-Neo 125M	125,198,592
XLNet	XLNet Large Cased	360,300,800
GPT-2	GPT-2 Large	774,030,080

types and a language model specifically for the sub-prompt inference. Once the *Generate* button is pressed, the model applies the inference settings from the right side and outputs three generated texts, see figure B.3. The immutable text is the unprocessed result and the mutable one in the input field the post-processed one. In case post-processed results are not optimal, users can either recreate them or manually change them. This way it is also possible to replace some generated text by one's own data of interest. Now it may happen that a prompt template needs multiple dynamic substitutes. In this case pressing the + button again opens a new sub-window, allowing users to create inferences of different sub-prompt categories simultaneously. Additionally, if a sub-prompt includes a dynamic blank, the substitute from the corresponding sub-prompt window is applied. For example, if a sub-prompt requires a definition, users can open a new sub-window to generate a definition which is automatically applied to the original sub-prompt.

The last element of AutoArg's interface is the *Your Prompt* field. It displays the current state of the prompt depending on the above fields. Red coloured text refers to static blanks (topics and verbs) and blue coloured text indicates dynamic blanks. Although the information from the fields detailed above is by default combined to compose this prompt, it is also possible to manually rewrite it by toggling the *Read Only* button. This mode gives users a chance to correct possible inconsistencies or to test their own prompts. Once all blanks have been replaced by substitutes, users can press the *Generation* button below to infer an argument. In contrast to the sub-prompt windows, users can press the generation button multiple times to increase the length of their argument or the adjacent button to stepwise reset it.

AutoArg does not only provide an argument generation interface, but also an evaluation interface, see appendix B.5. With this tool, annotators are able to manually annotate individual generated arguments. The annotation is limited to six metrics, further explained in the following evaluation chapter.

3.2.2 Limitations

A main limitation of AutoArg's current version is the amount of required resources. Not only does the tool keep multiple language models in its memory for an indefinite amount of time, but also does it not yet support GPUs. The used language models presented in section 3.1.2 are too large to be kept in memory simultaneously, so I only load three of them with smaller set-ups and let them run on CPUs. These AutoArg models are listed in table 3.3. While this reduces the overall quality of the generated texts, users are free to change decoding settings and recreate texts as often as they want, giving more freedom to exclude low-quality texts. Two other limitations derived from the resources are (1) the maximum response length and (2) the possible decoding mechanisms. Regarding the former, the maximum length is set to 30 tokens. This length is chosen to not overuse the models, especially when multiple users work with the tool at once, however, this length does not guarantee that sub-prompt inferences include enough tokens. As shown in section 3.1.7, the number of needed tokens per sub-prompt can reach up to 54. Additionally, AutoArg only supports sampling decoding methods, excluding beam-search. This means users have to use sampling to generate definitions, because deterministic approaches like beam-search require more time and resources.

Chapter 4

Evaluation

This chapter is split into two sections: automatic evaluation and manual evaluation, both based on Khatib et al. [2021]. However, instead of using both methods in parallel and comparing their results, I apply automatic evaluation to choose a qualitative sub-set of arguments. Only those arguments are manually evaluated.

4.1 Pre-Selection

Not all generated texts are valid arguments, so in this section I restrict the number of generated texts (112,512) to consider for the automatic evaluation:

1. Exclude texts generated with sub-prompts.
2. Exclude arguments, which after the post-processing have no more characters (empty strings).
3. Exclude demonstrative and meta prompt inferences from T5.

As explained in section 3.1.2, T5 is expected to summarise and extend the given prompt to an argument. However, meta prompts are longer than the maximum response length of 100 tokens. With T5's summary setting it only creates a prompt summary and no new argument. Furthermore, demonstrative prompts include example arguments from other domains which T5 is not able to interpret correctly but instead includes in the response, leading to a fusion of incoherent arguments. For these reasons, I exclude meta and demonstrative inferences of T5, c.f. examples in table 4.1. After applying the three above restrictions, a set of 99,750 argument inferences is left.

Table 4.1: Examples for meta and demonstrative prompt inferences with T5. For the meta type T5 only summarises the prompt. For the demonstrative type T5 tries to combine each mentioned example, similarly leading to a prompt summary.

Type	Argument
meta	question "What argument can I use in a debate if I want to argue about zoos?" I entered my question into the Argument Generator and waited. neotropay argued that zoos should be banned in many countries. the Argument Generator beeped, indicating that it has found the most suited argument.
demonstr.	in the position-to-know refers to what one knows about something. for instance, police officers might know about criminality. students might be in a position to know about tuition costs. school officials could be in a position to know about zero tolerance.

4.2 Automatic Evaluation

To find out which generated arguments after the pre-selection step are the best, I assess four automatic metrics, mostly based on Khatib et al. [2021]:

1. *Stance.* This is the same as the *argumentativeness* metric defined in the referred work, which uses the record-wise stance prediction from Stab et al. [2018] to distinguish between three categories: arguments without a stance are scored 1, arguments with different stances (pro and contra) 2, and arguments with one stance 3.

2. *Content Richness.* Similar to the *stance* metric, I derive this score from the referred work which is based on Schiller et al. [2021]. Depending on the number of aspects found, the more rich the content is: Arguments with a maximum of two aspects are scored 1, three to five aspects 2, and more than five aspects 3.

3. *Relevance.* Khatib et al. [2021] use an overlapping mechanism between prompts and arguments to calculate the arguments' relevance. This approach does not work for prompts in this work because they differ in length and can consist of multiple sentences. For this reason, I compare each generated argument to its respective topic. However, this means the overlapping approach is no longer applicable: Most post-processed arguments contain sections of the prompt with a topic mention. These arguments would be scored 3 by default. Additionally, if the topic occurs in another linguistic form the score would be lower, too. For these reasons I apply a sBERT model to find the semantic similarity between a topic and a generated argument. Although it is also possible to use

the Word2Vec similarity approach with spaCy, sBERT models can better distinguish between semantically similar and dissimilar arguments, detailed in the appendix, experiment C.2. With sBERT scores in range [-1, 1] (1 being semantically identical) I map these scores onto 1 for irrelevant, 2 for partly relevant and 3 for highly relevant. A higher sBERT similarity score does not imply that an argument is more relevant than another one. Therefore, based on the aforementioned experiment, I map specific sBERT score ranges onto the given integer scores, detailed, in the aforementioned experiment.

4. *Argumentativeness.* In contrast to the referred work, *argumentativeness* here refers to the occurrence of reasons and conclusions in a generated text. The high variety among argumentation schemes makes it difficult to find out if a generated argument represents the respective scheme. Therefore, this metric simplifies this task by just considering the amount of reasons (premises) and conclusions. I apply the fine-tuned T5 model by Betz and Richardson [2021], which is able to find reasons and conclusions in any given text. The model offers different options to analyse arguments from which I choose the *straight* chain option. I score all arguments without reasons or without conclusions 1, with at least one conclusion and exactly one reason 2, and with at least one conclusion and more than one reason 3. The motive for needing at least two reasons is that most argumentation schemes by Walton et al. [2008] have two premises and one conclusion.

To better interpret the meaning and influence of these metrics, appendix C.1 provides examples for each metric with scores of 1 and 3. It is important to highlight that these metrics consider whole generated arguments and not only the inferred sections by the language models. Some arguments include parts of the prompts – more or less depending on the scheme – which might have argumentative components or content aspects. During the evaluation process I do not exclude these parts from the generated arguments because arguments should be considered as closed entities. Removing sections from them would inevitably mean removing essential information, resulting in incomplete arguments.

Table 4.2 presents the average stance scores over models and prompt types. Stance scores range from 2.30 to 2.51, excluding T5 scores which go below 2.30. With most average scores being greater 2.0, generated arguments primarily contain at least one stance. The GPT-2 model and the meta prompt type respectively reach the highest average scores. In contrast to the stance metric, argumentativeness and content richness (tables 4.3 and 4.4) reach the highest scores for the prompt types short and descriptive. Argumentativeness scores

in general fluctuate between 1.33 and 2.50 and content richness scores between 1.36 and 2.18, again with T5 getting the worst scores. For both metrics GPT-2 remains the model with the highest overall scores. The final metric is relevance. Because I map this score on a set of natural numbers I also provide a table with the original scores (tables 4.6 and 4.5). While GPT-2 has the highest relevance scores, too, this metric's results are similar for all model-prompt-type pairs, ranging from 2.12 to 2.47. However, the trend of the lowest scores by T5 can also be seen here. Regarding individual prompt types, no specific type stands out, because each (excluding meta) at least once reaches the highest score in combination with a model.

One insight to derive from these results is that GPT-2 – with the given prompt templates and decoding settings – reaches the best scores over all metrics. At the same time, T5 overall gets the worst scores, leading to the assumption that some form of fine-tuning or training is required to generate arguments with T5. Another interesting aspect to cover are the score distributions depending on prompt types. While for argumentativeness and content richness a clear preference towards short and descriptive types can be seen, the opposite is the case for the stance score. A reason for this is the argument length. The longer an argument is, the more content, premises and conclusions it might entail; hence, when the generation process is interrupted early or generated arguments are cut at some position, content gets lost. This cutting is utilised for dialogue, demonstrative and meta prompts but not for short and descriptive ones. Short and descriptive arguments are the longest, c.f. section 3.1.8, and thus reach the highest argumentativeness and content richness scores. On the other side, longer arguments have a greater chance of including opposing stances, leading to a lower stance score. Since meta, dialogue and demonstrative arguments include less text, they are also less likely to include different stances, resulting in a higher stance score. For the relevance score no specific trend regarding prompt types can be seen.

Besides the relation of language models and prompt types, I analyse the relation between prompt types and argumentation schemes. All scores for content richness, stance, argumentativeness and relevance are listed in the tables 4.7, 4.8, 4.9 and 4.10 respectively. Going through these tables, the top-scored schemes nearly always vary. This is to no surprise as the required information differs with each scheme structure. *Arguments from Popular Practice* might have the lowest content richness score, but they only rely on the mentioning of a practice and that it is legitimate because of its popularity. In contrast, *Arguments from Bias* require a person that states a claim and additionally a reason why this person is biased, which leads to the lose of credibility. The high content score for this scheme can also be explained by the large prompt section from my template which is by default included in the final

Table 4.2: Arithmetic mean of all **stance** scores by models and prompt types. The highest scores for each model and overall are emphasised.

	Short	Dialogue	Demonstrative	Descriptive	Meta	All
Transformer XL	2.33	2.34	2.33	**2.36**	2.32	2.34
XLNet	2.39	**2.41**	2.34	2.33	2.34	2.36
GPT-2	2.43	2.46	2.43	2.41	**2.50**	**2.45**
GPT-Neo	2.45	2.47	2.37	2.39	**2.47**	2.43
T5	1.82	**2.27**	-	2.17	-	2.09
GPT-J	2.45	2.45	2.38	2.41	**2.51**	2.44
GPT-3	***2.48***	*2.38*	-	*2.47*	-	*2.44*
All	2.34	2.40	2.37	2.36	**2.43**	

argument. In conclusion, just because a scheme has a lower content richness score, this does not mean that it generates worse arguments. The influence of long prompt sections included in the final argument can also be seen in the argumentativeness scores where the top three schemes all include essential text snippets from their prompts. By only considering the included prompt section, *Arguments from Example* and *Arguments from Analogy* for most verbs have a default argumentativeness score of 2, while *Arguments from Circumstantial Ad Hominem* for most verbs already have a score of 3. Assessing the quality by this metric alone would favour such arguments.

Finally, the stance and relevance metrics introduce further top-schemes, resulting in diverse sets of best schemes depending on individual metrics. But as seen for content richness and argumentativeness, one metric alone does not suffice to evaluate an argument. For this reason I combine all metric results in an attempt to calculate overall argument quality scores and to directly compare generated arguments with each other, described in the next section.

4.3 Manual Evaluation

The sheer amount of generated arguments makes it infeasible to manually evaluate all of them. For this reason, I select a subset of arguments depending on the (1) topic, (2) prompt type, (3) argumentation scheme, (4) model and (5) batch position, c.f. table 4.15.

Regarding controversial topics, I choose four based on different categories: *coal mining* (economy), *vegetarianism* (social life), *the teacher tenure reform* (politics), and *the right to keep and bear arms* (law). Additionally, I exclude all inferred arguments which are generated by T5 or GPT-3. The scores of

Table 4.3: Arithmetic mean of all **argumentativeness** scores by models and prompt types. The highest scores for each model and overall are emphasised.

	Short	Dialogue	Demonstrative	Descriptive	Meta	All
Transformer XL	1.78	1.71	1.51	**1.87**	1.74	1.72
XLNet	**2.38**	1.67	1.75	2.35	1.51	1.93
GPT-2	2.21	1.72	1.79	**2.50**	1.85	**2.01**
GPT-Neo	2.27	1.65	1.79	**2.52**	1.67	1.98
T5	1.42	1.61	-	**1.97**	-	1.67
GPT-J	2.29	1.56	1.79	**2.51**	1.64	1.96
GPT-3	*1.87*	*1.33*	-	***2.34***	-	*1.85*
All	2.03	1.61	1.73	**2.29**	1.68	

Table 4.4: Arithmetic mean of all **content richness** scores by models and prompt types. The highest scores for each model and overall are emphasised.

	Short	Dialogue	Demonstrative	Descriptive	Meta	All
Transformer XL	**1.79**	1.67	1.63	**1.79**	1.65	1.71
XLNet	**1.95**	1.61	1.47	1.83	1.56	1.68
GPT-2	2.15	1.76	1.47	**2.18**	1.81	**1.88**
GPT-Neo	**2.12**	1.66	1.43	2.07	1.64	1.78
T5	1.36	1.77	-	**1.80**	-	1.64
GPT-J	2.11	1.64	1.44	**2.13**	1.64	1.79
GPT-3	***1.99***	*1.62*	-	*1.95*	-	*1.86*
All	1.92	1.68	1.49	**1.97**	1.66	

T5 in the automatic evaluation are the lowest for each metric and GPT-3 lacks in comparability. Regarding the argumentation schemes, Walton et al. [2008] propose a classification system to group similar argumentation schemes. From each of these groups I select one representative with the exception of arguments from expert opinion and popular opinion. These schemes fall into the same category, however, because both require sub-prompt information they are especially interesting for a manual analysis. Consequently, I include both. Groups and selected argumentation schemes are listed in table 4.11.

The combination of topics, models and argumentation schemes results in $4 \cdot 5 \cdot 10 = 200$ arguments. However, this still omits the five prompt types and the fact that for each inference setting a batch of five arguments is generated. By including all prompt types and full batches, the final arguments

Table 4.5: Arithmetic mean of all **raw relevance** scores by models and prompt types.

	Short	Dialogue	Demonstrative	Descriptive	Meta	All
Transformer XL	0.50	0.58	0.55	0.58	0.59	0.56
XLNet	0.56	0.59	0.62	0.57	0.60	0.59
GPT-2	0.56	0.59	0.63	0.59	0.59	0.59
GPT-Neo	0.57	0.60	0.64	0.59	0.60	0.60
T5	0.47	0.59	-	0.63	-	0.56
GPT-J	0.57	0.59	0.63	0.59	0.60	0.60
GPT-3	0.55	0.57	-	0.56	-	0.56
All	0.54	0.59	0.61	0.59	0.59	

Table 4.6: Arithmetic mean of all **mapped relevance** scores by models and prompt types. The highest scores for each model and overall are emphasised.

	Short	Dialogue	Demonstrative	Descriptive	Meta	All
Transformer XL	2.40	2.39	**2.43**	2.42	2.40	2.41
XLNet	**2.47**	2.40	2.36	2.45	2.41	2.42
GPT-2	**2.47**	2.42	2.35	2.44	2.44	**2.43**
GPT-Neo	**2.44**	2.41	2.34	**2.44**	2.41	2.41
T5	2.12	**2.38**	-	2.32	-	2.27
GPT-J	2.43	2.40	2.34	**2.45**	2.42	2.41
GPT-3	*2.41*	*2.34*	-	***2.45***	-	*2.4*
All	2.39	2.39	2.37	**2.42**	2.41	

set would include $200 \cdot 5 \cdot 5 = 5000$ arguments which are too many for a manual evaluation. Therefore, having 200 groups of arguments and each with 25 arguments given the prompt types and batch size, I only select the best argument out of the 25, resulting in a final evaluation set of 200 arguments. The quality of an argument can be determined by its four automatic scores. While it is possible to weight each score to introduce more and less important metrics, this requires a ranking. To not introduce additional bias by manually determining metric weights, each metric has an equal influence on the quality. The quality score is calculated as arithmetic mean over all metric scores. The argument with the highest quality score out of the 25 is selected. The distribution of metric scores in the final set and the selected prompt types is visualised in figure 4.2.

Table 4.7: Arithmetic mean of all **content richness** scores by argumentation schemes and prompt types. The highest scores per scheme and the top three schemes are emphasised.

Argumentation Scheme	Short	Dialog	Demonstrative	Descriptive	Meta	Avg
Argument from Position to Know	**1.88**	1.53	1.41	**1.88**	1.64	1.67
Argument from Expert Opinion	**1.94**	1.59	1.41	1.90	1.62	1.69
Argument from Popular Opinion	1.92	1.63	1.34	**2.10**	1.63	1.72
Argument from Popular Practise	1.73	1.50	1.40	**1.75**	1.59	1.59
Argument from Example	1.75	1.60	1.64	**1.78**	1.66	1.69
Argument from Analogy	1.92	1.65	1.56	**1.94**	1.55	1.73
Rhetorical Argument from Oppositions	1.82	1.61	1.62	**1.86**	1.55	1.69
Argument from Definition to Verbal Classification	**1.93**	1.85	1.49	1.76	1.90	1.79
Argumentation from Sacrifice	**1.88**	1.53	1.75	1.87	1.44	1.70
Argument from Practical Reasoning	1.77	1.49	1.41	**1.83**	1.54	1.61
Argument from Ignorance	1.80	1.47	1.38	**1.83**	1.38	1.57
Argument from Cause to Effect	1.93	1.93	1.33	**2.31**	1.49	1.80
Argument from Correlation to Cause	2.20	2.08	1.67	**2.23**	2.07	**2.05**
Argument from Sign	1.91	1.62	1.19	**2.09**	1.50	1.66
Argument from Consequences	1.92	1.63	1.58	**1.94**	1.62	1.74
Pragmatic Argument from Alternatives	1.97	1.65	1.44	**2.10**	1.67	1.77
Argument from Danger Appeal	2.33	2.03	1.51	**2.36**	2.02	**2.05**
Argument from Commitment	1.91	1.65	1.44	**1.93**	1.55	1.70
Argument from Pragmatic Inconsistency	1.77	1.63	1.38	**1.87**	1.62	1.65
Argument from Inconsistent Commitment	1.78	1.69	1.49	**1.89**	1.64	1.70
Argument from Circumstantial Ad Hominem	1.84	1.63	1.60	**1.87**	1.72	1.73
Argument from Bias	**2.21**	2.11	1.68	**2.21**	2.10	**2.06**
Avg	1.92	1.68	1.49	**1.97**	1.66	

Table 4.8: Arithmetic mean of all **stance** scores by argumentation schemes and prompt types. The highest scores per scheme and the top three schemes are emphasised.

Argumentation Scheme	Short	Dialog	Demonstrative	Descriptive	Meta	Avg
Argument from Position to Know	2.31	2.44	**2.59**	2.43	2.52	2.46
Argument from Expert Opinion	2.47	2.56	2.16	2.43	**2.60**	2.44
Argument from Popular Opinion	2.36	2.54	**2.68**	2.38	2.62	2.52
Argument from Popular Practise	2.38	2.50	**2.77**	2.39	2.48	2.50
Argument from Example	1.92	1.95	**2.17**	2.05	1.96	2.01
Argument from Analogy	2.07	1.83	1.94	**2.08**	1.62	1.91
Rhetorical Argument from Oppositions	2.07	1.99	**2.56**	2.17	1.81	2.12
Argument from Definition to Verbal Classification	2.10	2.07	**2.22**	1.93	2.12	2.09
Argumentation from Sacrifice	2.53	2.80	2.34	2.53	**2.89**	2.62
Argument from Practical Reasoning	2.33	2.44	1.27	2.33	**2.59**	2.19
Argument from Ignorance	2.24	**2.63**	2.36	2.40	2.55	2.44
Argument from Cause to Effect	2.50	2.77	**2.82**	2.49	**2.82**	**2.68**
Argument from Correlation to Cause	2.30	2.44	**2.45**	2.34	2.41	2.39
Argument from Sign	2.52	2.80	**2.98**	2.58	2.88	**2.75**
Argument from Consequences	2.11	1.72	**2.88**	2.22	1.80	2.15
Pragmatic Argument from Alternatives	2.44	2.60	1.95	2.47	**2.67**	2.43
Argument from Danger Appeal	2.53	2.81	2.88	2.51	**2.90**	**2.73**
Argument from Commitment	2.39	2.66	**2.80**	2.44	2.65	2.59
Argument from Pragmatic Inconsistency	2.35	2.37	1.50	2.31	**2.50**	2.20
Argument from Inconsistent Commitment	2.57	2.70	2.39	2.54	**2.75**	2.59
Argument from Circumstantial Ad Hominem	**2.18**	2.04	1.72	**2.18**	2.17	2.06
Argument from Bias	2.23	2.16	**2.76**	2.36	2.15	2.33
Avg	2.34	2.40	2.37	2.36	**2.43**	

Table 4.9: Arithmetic mean of all **argumentativeness** scores by argumentation schemes and prompt types. The highest scores per scheme and the top three schemes are emphasised.

Argumentation Scheme	Short	Dialog	Demonstrative	Descriptive	Meta	Avg
Argument from Position to Know	1.92	1.35	1.56	**2.22**	1.58	1.73
Argument from Expert Opinion	1.93	1.41	1.26	**2.12**	1.42	1.63
Argument from Popular Opinion	2.02	1.52	1.09	**2.35**	1.60	1.71
Argument from Popular Practise	1.98	1.42	1.08	**2.24**	1.52	1.65
Argument from Example	2.43	2.31	2.19	**2.52**	2.35	**2.36**
Argument from Analogy	**2.64**	2.40	1.20	2.60	2.39	**2.25**
Rhetorical Argument from Oppositions	2.51	2.08	2.32	**2.56**	2.15	2.32
Argument from Definition to Verbal Classification	1.57	1.13	1.86	**2.01**	1.51	1.62
Argumentation from Sacrifice	1.88	1.38	1.91	**2.27**	1.34	1.76
Argument from Practical Reasoning	1.86	1.31	**2.09**	1.99	1.26	1.70
Argument from Ignorance	1.92	1.39	2.45	**2.60**	1.28	1.93
Argument from Cause to Effect	1.93	1.57	1.10	**2.28**	1.34	1.64
Argument from Correlation to Cause	2.28	1.93	1.90	**2.43**	1.97	2.10
Argument from Sign	1.93	1.42	1.06	**2.24**	1.33	1.59
Argument from Consequences	1.80	1.20	1.88	**2.07**	1.30	1.65
Pragmatic Argument from Alternatives	1.61	1.23	1.87	**2.20**	1.18	1.62
Argument from Danger Appeal	2.34	1.99	1.86	**2.47**	1.92	2.12
Argument from Commitment	1.81	1.32	1.05	**1.89**	1.28	1.47
Argument from Pragmatic Inconsistency	2.04	1.80	**2.62**	2.15	1.94	2.11
Argument from Inconsistent Commitment	1.61	1.39	1.18	**1.78**	1.36	1.47
Argument from Circumstantial Ad Hominem	2.78	2.74	2.63	2.78	**2.92**	**2.77**
Argument from Bias	**2.55**	2.03	1.81	2.50	2.10	2.2
Avg	2.03	1.61	1.73	**2.29**	1.68	

Table 4.10: Arithmetic mean of all **relevance** scores by argumentation schemes and prompt types. The highest scores per scheme and the top three schemes are emphasised.

Argumentation Scheme	Short	Dialog	Demonstrative	Descriptive	Meta	Avg
Argument from Position to Know	2.35	2.24	2.20	**2.38**	2.28	2.29
Argument from Expert Opinion	2.38	2.39	2.30	2.37	**2.44**	2.37
Argument from Popular Opinion	2.35	2.30	2.12	**2.41**	2.34	2.3
Argument from Popular Practise	2.35	2.23	2.25	**2.37**	2.29	2.3
Argument from Example	2.43	2.53	2.50	2.54	**2.58**	**2.52**
Argument from Analogy	2.48	2.48	**2.51**	2.46	**2.51**	2.49
Rhetorical Argument from Oppositions	2.51	2.49	2.18	2.46	**2.52**	2.43
Argument from Definition to Verbal Classification	2.30	**2.60**	2.42	2.27	2.31	2.38
Argumentation from Sacrifice	2.39	2.36	2.24	**2.43**	2.40	2.37
Argument from Practical Reasoning	2.39	2.42	2.44	2.44	**2.47**	2.43
Argument from Ignorance	2.37	**2.38**	2.18	2.33	2.25	2.3
Argument from Cause to Effect	2.33	2.27	2.24	**2.41**	2.26	2.3
Argument from Correlation to Cause	2.40	2.51	2.54	2.50	**2.57**	2.51
Argument from Sign	**2.33**	2.22	2.15	2.29	2.26	2.25
Argument from Consequences	2.36	2.36	2.37	**2.43**	2.36	2.37
Pragmatic Argument from Alternatives	2.38	**2.40**	2.31	**2.40**	**2.40**	2.38
Argument from Danger Appeal	2.38	2.36	**2.54**	2.39	2.38	2.41
Argument from Commitment	2.41	2.46	**2.64**	2.43	2.48	2.49
Argument from Pragmatic Inconsistency	2.02	2.22	**2.43**	2.27	2.14	2.22
Argument from Inconsistent Commitment	2.52	2.53	2.24	2.51	**2.55**	2.47
Argument from Circumstantial Ad Hominem	2.64	2.63	2.62	2.59	**2.66**	**2.63**
Argument from Bias	2.52	2.60	2.62	2.52	**2.68**	**2.59**
Avg	2.39	2.39	2.37	**2.42**	2.41	

Table 4.11: Argumentation groups [Walton et al., 2008] and one of their representatives. Representatives were manually selected for the manual evaluation.

Walton et al.'s [2008] Group	Argumentation Scheme
Practical Reasoning	Argument from Practical Reasoning
Abductive Reasoning	Argument from Sign
Causal Reasoning	Argument from Cause to Effect
Arguments from Position to Know	Argument from Position to Know
	Argument from Expert Opinion
Arguments from Commitment	Argument from Commitment
Arguments Attacking Personal Credibility	Argument from Pragmatic Inconsistency
Arguments from Popular Acceptance	Argument from Popular Opinion
Arguments Based on Cases	Argument from Example
Verbal Classification Arguments	Argumentation from Sacrifice

To get a general impression on the generated arguments' quality, this thesis only includes a preliminary manual evaluation. I hired one expert for $100 on Upwork who annotated all 200 arguments with AutoArg's annotation interface (see appendix B.5). For each argument, the expert was asked to score it with respect to the following metrics and questions:

1. *Topic Relevance.* Does the text comprise content relevant to the given topic? $\{1, 2, 3\}$
2. *Stance.* Does the text convey an explicit or implicit pro or con stance towards any topic? $\{1, 2, 3\}$
3. *Content Richness.* Does the text contain useful information and cover different aspects? $\{1, 2, 3\}$
4. *Argumentativeness.* Does the text represent the argumentation scheme? $\{1, 2, 3\}$
5. *Plausibility.* Does the text comprise plausible content and does it not contrast with commonsense knowledge? $\{1, 2, 3\}$
6. *Bias.* Does the text include any social bias or abusive language? $\{yes, no\}$

Regarding the numeric values, 1 means no, 2 means partly and 3 means yes. The first four metrics have the same intention as the automatic ones and can thus be directly compared to each other. Plausibility and bias are reserved for the manual annotation. Figure 4.1 presents an overview of the annotated

score distributions, which can be compared to the automatic distribution in figure 4.2.

Regarding the automatic metrics, only stance in the manual evaluation has a similar score distribution. For the three other metrics, the overall score counts mainly shift from 3 to 2, with the most notable drop for the argumentativeness score. This can easily be explained by the implementation of this automatic score. The automatic calculation merely considers the number of reasons and conclusions in an text while ignoring relations among them. Some generated arguments include multiple reasons, but they might not be relevant to the argumentation scheme. For the bias metric the annotator did not find any argument with social biases, which could be explained by the selected topics. Other controversial topics might be more prone to introduce these biases. The plausibility score is similar to results from other metrics, with most arguments reaching a score of 2 which means they are partly plausible.

Next to an overall analysis, I also consider scores related to individual parameters, namely topics, argumentation schemes and language models. Scores depending on topics – listed in table 4.12 – show variations. While *coal mining* has the highest average scores for most metrics, the opposite is the case for topic *vegetarianism*. This demonstrates that the domain knowledge of a language model differs and that not all controversial topics lead to arguments of the same quality. These variations can also be seen for other parameters, like argumentation schemes in table 4.13. *Argument from Position to Know*, *Arguments from Expert Opinion* and *Arguments from Cause to Effect* reach the best scores, while *Arguments from Pragmatic Inconsistencies* mainly reach the lowest scores. For once, this result hints at differences of scheme complexities and the intrinsic capabilities of language models. Language models seem to easily understand how to formulate a cause-to-effect scheme. On the other side, both the position-to-know and expert-opinion schemes include sub-prompt information. These sub-prompts lower the complexity for a single prompt to generate multiple argument propositions and hint at the advantage of the proposed prompt composition method. However, for a final conclusion on this aspect further prompt templates that include sub-prompts are required. Additionally, a comparison between generated arguments with and without sub-prompts is needed. The last table 4.14 presents the average scores related to language models. As for the automatic scores, models from the GPT series throughout have the highest scores. While GPT-2 for might not reach the best scores for each metric, all GPT models have similar results and thus can be considered the best for the argument generation task.

All of the manual analyses lead to tentative conclusions, because the annotation was only conducted by one expert. However, they provide a first impression on which settings and prompts might result in better arguments

and which prompts might require a rework.

4.4 Ethical Concerns

Although this thesis does not go into detail on how to solve problems of ethical concerns, I want to provide some aspects to consider when working with language models in the context of argument generation. A main issue with language models is not only that they sometimes fabricate information – as seen in

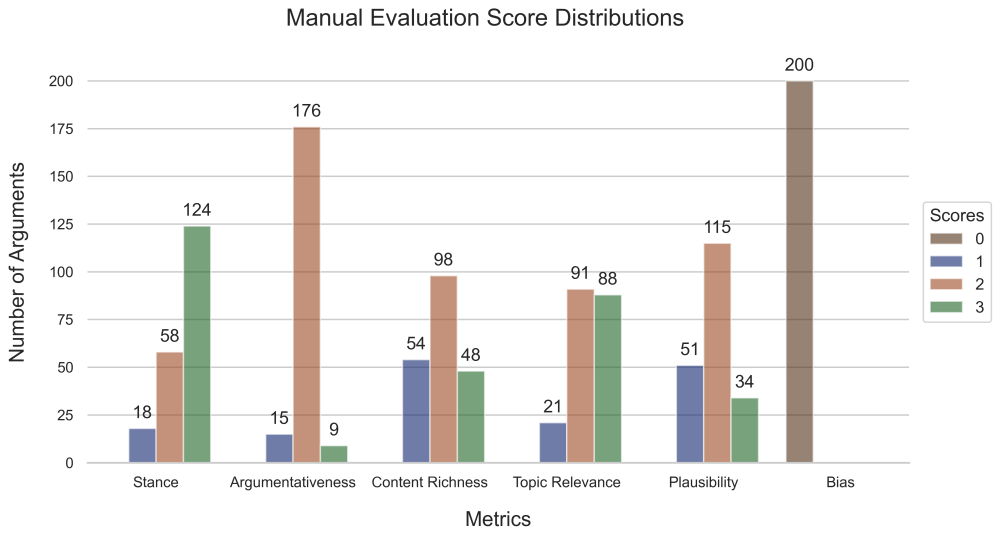
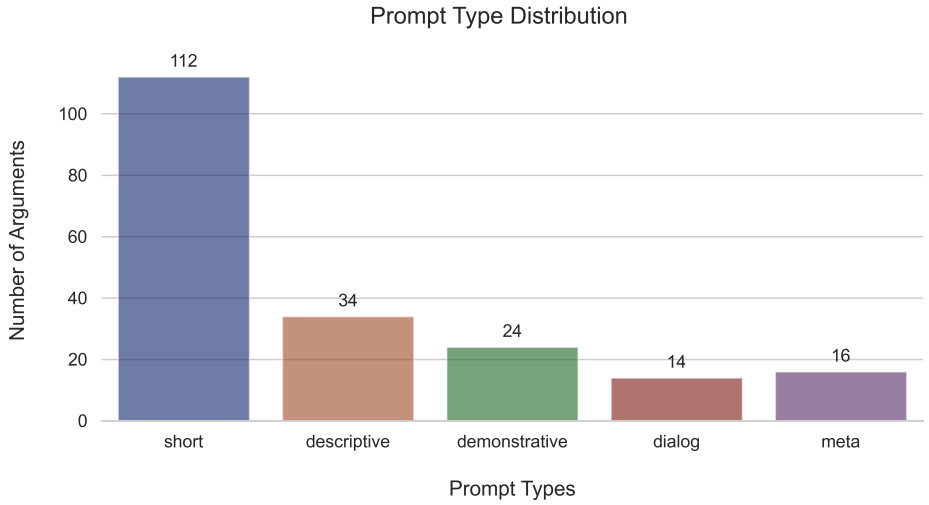


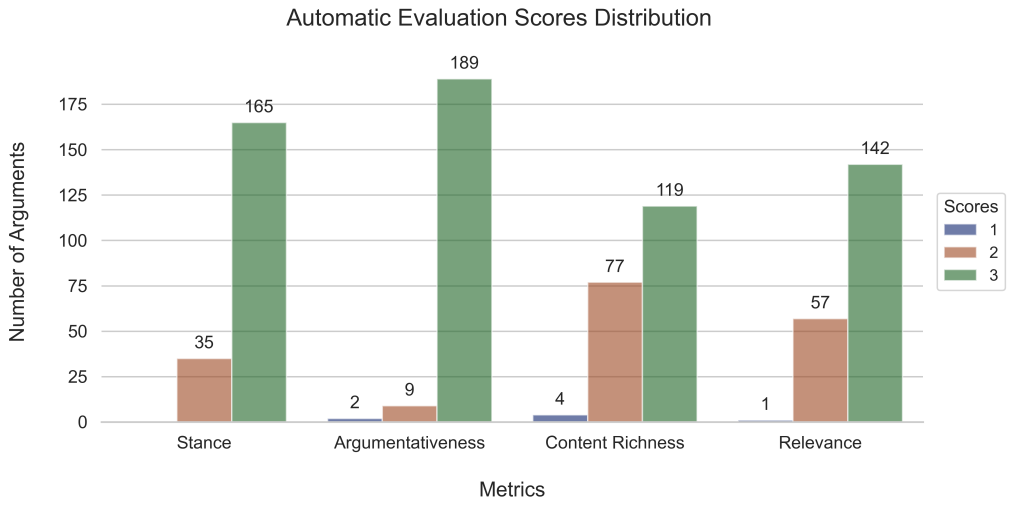
Figure 4.1: Distribution of manually annotated metric scores in the evaluation arguments set. The existence of bias is mapped onto 0 for no and 1 for yes. However, in this arguments set no argument was annotated as expressing biases.

Table 4.12: Arithmetic mean of all scores by topics. The highest and lowest score per metric is emphasised. Rel. = Topic Relevance, CR = Content Richness, Arg. = Argumentativeness.

Topic	Rel.	Stance	CR	Plausibility	Arg.	Bias
coal mining	2.28	2.52	**2.10**	**2.02**	**2.02**	0.00
the right to keep and bear arms	2.34	2.50	1.92	1.94	1.94	0.00
the teacher tenure reform	**2.36**	**2.64**	1.96	1.98	1.94	0.00
vegetarianism	**2.36**	2.46	1.90	1.72	1.98	0.00



(a) Distribution of prompt types in the evaluation arguments set. The short prompt type entails more than half of the arguments.



(b) Distribution of automatically annotated scores for all metrics in the evaluation arguments set.

Figure 4.2: Statistic of arguments in the manual evaluation dataset regarding their metric and prompt type distributions.

Table 4.13: Arithmetic mean of all scores by argumentation schemes. The highest and lowest score per metric is emphasised. Topic Rel. = Relevance, CR = Content Richness, Arg. = Argumentativeness.

Argumentation Scheme	Rel.	Stance	CR	Plausibility	Arg.	Bias
Practical Reasoning	2.25	2.60	1.90	1.70	1.90	0.00
Sign	2.25	**2.70**	1.90	2.05	2.05	0.00
Sacrifice	2.20	2.45	1.70	1.70	2.00	0.00
Cause to Effect	2.40	2.45	2.15	**2.30**	1.85	0.00
Position to Know	2.40	2.60	**2.35**	2.05	**2.10**	0.00
Expert Opinion	**2.80**	2.50	**2.35**	2.15	1.85	0.00
Commitment	2.00	2.55	1.85	1.75	2.05	0.00
Bias	2.55	2.65	1.80	1.60	2.00	0.00
Pragmatic Inconsistency	2.10	2.40	1.60	1.70	1.80	0.00
Popular Opinion	2.40	2.40	2.10	2.15	2.10	0.00

Table 4.14: Arithmetic mean of all scores by models. The highest and lowest score per metric is emphasised. Rel. = Topic Relevance, CR = Content Richness, Arg. = Argumentativeness.

Model	Rel.	Stance	CR	Plausibility	Arg.	Bias
Transformer XL	1.75	2.25	1.27	1.30	1.75	0.00
XLNet	2.27	2.40	1.85	1.85	2.00	0.00
GPT-2	**2.58**	2.75	2.25	**2.23**	1.98	0.00
GPT-Neo	**2.58**	**2.80**	2.20	2.12	**2.08**	0.00
GPT-J	2.50	2.45	**2.27**	2.08	2.05	0.00

the expert-name experiment C.1 – but also that they are heavily influenced by biased training datasets. These issues have been studied by different research groups. Spliethöver and Wachsmuth [2020] analyse social biases in multiple debating corpora, with social bias defined as "prejudice against, as well as unequal treatment or discrimination of, certain social groups in society". To analyse these biases they train GloVe word embeddings on each corpus respectively and evaluate them by applying the *Word Embedding Association Test* WEAT. This test is based on an association test and functions by calculation the closeness of an target word (e.g. a social group) to words from an association list (e.g. pleasant or unpleasant). Their experiments clearly emphasise biases toward male and European-American names: Not only is the mean cosine distance of stem career embeddings like math closer to male names in

Table 4.15: Parameters that influence the number of generated arguments. The overall count refers to the theoretically possible number of generated arguments, excluding the pre-selection. The count in the manual evaluation refers to the selected arguments for the manual evaluation task.

Parameter	Overall Count	Count in Manual Eval.
Argumentation Schemes	22	10
Models	7	5
Prompt Types	5	-
Controversial Topics	32	4
Inference Batch Size	5	-
$\prod$	110,400	200

contrast to female ones, European-American names are also closer to pleasant terms than African-American ones. A consequence of training or fine-tuning on these debating datasets would be that the language model incorporates and replicates these biases, too.

Nevertheless, training on special domain corpora is not necessary to force a language model to generate biased texts. In earlier experiments by Solaiman et al. [2019] the authors analyse the misuse capabilities of GPT-2 by cooperating with different organisations to determine misuse possibilities. While it is stated that cases of large-scale misuses are unknown (the same is added in the GPT-3 paper one year later [Brown et al., 2020]), their partner organisations highlight multiple features of GPT-2 that can be possible misused. It is shown that fine-tuning GPT-2 on extremist text corpora increases the perceived humanness of generated texts with respect to this domain. Applying this insight on argument generation, a model could not only be fine-tuned on some pro-extremism corpus, but simultaneously generate more human-like arguments that are harder to see through. Alternatively, debating corpora could be used to make the model a better debater. However, one must be aware that this procedure introduces more biased data as seen by Spliethöver and Wachsmuth [2020]. One other interesting aspect by Solaiman et al. is, however, that a correlation between generated text lengths and the difficulty of deciding whether the text was written by a computer exists. Longer texts can be more easily classified as human or computer written. Because arguments require a minimum length, due to their premises and conclusion, humans or specially trained models should be able to differentiate the source of creation. While this does not solve the problem of social bias in generated texts, generated arguments can at least be easily classified as computer-written.

Brown et al. [2020] go into more detail regarding stereotypes in their GPT-3 language models and concentrate on racial, religious and gender biases. Their experiments show that for all three categories, independent of model size, biases occur. As seen by the corpus tests of Spliethöver and Wachsmuth [2020], GPT-3 also rather associates occupations like legislator, banker or professor emeritus with men, and midwife, nurse, receptionist or housekeeper with women. Similar biases occur regarding race, which are tested by conducting a racial sentiment analysis. The authors illustrate that over all model sizes significant differences between racial sentiments occur. However, these differences become smaller with larger models, which is especially true for the input *black*: While models with less than 2.7B parameters generate texts with rather negative sentiment scores, *black* reaches a neutral sentiment score with GPT-3. For the question of religious content the authors analyse what words typically occur in the generated text of GPT-3 when the prompt contains a religion. In contrast to other religions, Islam most often leads to the generation of terms like *violent* or *terrorism*.

Because my thesis relies on language models, such biases could occur in any generated argument, especially when the topic directly states a biased term. To demonstrate this issue I utilise the topic *black people* in AutoArg. I apply GPT-2 and keep the default top-p sampling. While using the short prompt type and the *Argument from Cause to Effect* scheme, I generate two texts:

- *Black people lead to* violence and conflict.
- *Black people lead to* the collapse of any society and the destruction of the natural environment and human life.

While the model seems to be negatively biased towards black people, formulating a prompt like *Black people lead to* manifests a negative narrative. Therefore, such a prompt accesses the model's biased knowledge. When changing the template to the scheme *Arguments from Popular Opinion*, the narrative changes:

- *For the most part, people agree that black people* should not be allowed to be treated any differently from white people in the criminal justice system.
- *For the most part, people agree that black people* need to be involved in a larger role in shaping our political institutions.

Now the perspective changes from anti-black to black-activism. The generated statements hint at the biases that are already included in a prompt formulation,

even if not intended.[1] One reason why I provide the AutoArg web tool is to circumvent this problem: Users are free to manipulate any prompt or switch to another template in case they are unsatisfied with biased results. Naturally, this highly depends on how users work with the tool, meaning whether they can differentiate social biases in generated texts and whether they even want to remove them.

In conclusion, the argument generation pipeline propagates all problems that are already known for language models in general: Training data and prompt formulations lead the model to generate false or biased texts. This problem is especially true for prompts of the demonstrative type. Because I self formulate three example arguments per prompt and choose their ordering, they manipulate the generated argument. Yet, a new problem introduced by this thesis is the attempt to generate factually correct data with sub-prompts, as for definitions and expert-names. Doing so without any correctness validation results in a system which possibly feeds unchecked and incorrect data to itself and generates new texts based on this data. For example, in case a prompt requires an expert-name and uses an irrelevant or even non-existing one as substitute, even if the generated claim sounds plausible, it is based on an incorrect expert and thus a fallacy. This becomes highly problematic when the incorrect substitutes seem plausible without further research, leading to believable but fallacious arguments.

[1] For a strong conclusion regarding prompt biases, more topics, prompts and generated texts must be analysed, which is out of scope for this thesis.

Chapter 5

Conclusions

With this work being the first to consider the combination of automated argument generation and Douglas Walton's argumentation schemes, most approaches and insights can be further refined. Therefore, in this section I first go into detail about future works and which aspects can be optimised. Afterwards, I present a brief conclusion regarding this whole thesis.

5.1 Future Work

This work demonstrates the capabilities of default language models, so one next step would be to fine-tune them, for instance as done by Khatib et al. [2021]. I presume that models designed for argumentation generation tasks reach higher scores during the automatic and manual evaluation. Another aspect to be improved is to automatically cut a generated text at the ending position of the argument. While some prompt types utilise semantic signifiers like quotation marks, *short* and *descriptive* prompts are not determined by any ending token. Therefore, it is necessary to cut them at the correct position, otherwise the generated text might derail or introduce multiple arguments at once. On the other side, prompt types with ending signifiers can be improved by forcing the language model to not generate the ending token until some threshold length is reached. An additional issue is the default generation length. As stated before, argumentation schemes differ in their information content and thus need individual token lengths. By applying the methods shown in this thesis to build an argumentation corpus, the generated argument lengths could be manually optimised and used to define maximum token lengths for each scheme respectively. As for the generation process, not only the maximum token length can be improved, but also the decoding mechanism. Nucleus sampling might provide mostly human like texts, however, as seen in experiment C.3, the generated texts' quality differs depending on the language model. These results

suggest using individual decoding parameters or methods for each model.

An improvement regarding the manually designed prompts is to give them the ability to not only frame an argument (for example as social, political or economical), but also to generate pro or contra arguments for all schemes. This thesis mainly focuses on topic-relatedness with arguments limited by a model's intrinsic bias. So, reworking the prompts to access additional model knowledge is beneficial in a debate setting and for a user base which specifically searches for pro or contra arguments. Also, the existing frames in the given prompts can be changed and analysed. For instance, the *dialog* prompt type might require a teacher and student as interlocutors, but it would also be interesting to see how other person groups are expected to interact by a model. Another step to optimise manual prompts, considering the ethical concerns in the previous chapter, would be to analyse which prompts more easily generate biased texts, for instance by applying the WEAT test. Prompts with high bias probabilities then can be reformulated. Another idea would be to use the critical questions formulated by Walton: the language models could be asked to answer the critical questions regarding the generated argument and depending on the answers the argument can be validated.

Generated arguments, especially ones with few tokens, must also be tested for completeness. Since language models simulate human written texts, they also tend to miss enthymemes like premises which are pre-defined by each scheme. An additional pipeline step could be included to complete arguments by using an external algorithm or by applying a language model. The latter method could also rely on prompts in which the model can be asked for any missing components in the given argument.

In combination with the AUTOARG web-interface, a manual evaluation could be added where users are asked to compare the automatically generated arguments to the ones generated with help from the web-tool. Such an evaluation can indicate how users interact with the tool and whether they prefer their individual arguments. Another aspect to cover for the web-interface is a search engine for the automatically generated arguments. Since each user might consider different metrics as more important, a search engine supporting a selection of score ranges might be beneficial. A final important aspect to optimise regarding AUTOARG is the resources issue. The system not only should provide larger models with GPU support, but also dynamically load models on user requests for less hardware workload.

On the subject of the experiments, another on to conduct would be the comparison of retrieval methods versus the sub-prompt information generation. This experiment could be done by first manually defining plausible substitutes to simulate a perfect information retrieval system. Then the newly generated arguments' quality can be compared to the original ones. In case the

scores improve, retrieval methods can be tested to see how well they extract the required information from some dataset. In case the automatic generation logic is not replaced by an information retrieval method, an additional improvement would be a system to validate the generated information. As seen in experiments like C.1, similarity scores are unable to tell if an expert name is relevant to a topic. Another approach could be by asking the language model specifically about the name and what topic it might combine the name with. The intention here is that if the model can not generate the correct topic depending on the name, the name is less likely to be relevant.

5.2 Conclusion

In this thesis I analysed how to manually design prompt templates that can be utilised to automatically generate arguments given a controversial topic and argumentation scheme. I specifically illustrated how to systematise this manual prompt engineering process to design comparable prompts over different schemes. I also showed how to formulate compound prompts by using separate prompt templates to generate additional information like expert names that function as substitutes. The final prompt template collection includes templates over 22 schemes, with each scheme represented by five prompt types. For the automatic evaluation, I introduced two new metrics: a distribution based relevance score and an argumentativeness score based on the number of reasons and conclusions in a generated text. Not only did I apply these metrics to analyse the generated arguments' quality, but also to filter the top 200 arguments with respect to four controversial topics. Regarding the former, I compared each template's generation capabilities over six open-source language models,[1] and showed that GPT-2 XL during the automatic evaluation reaches the best average scores. Additionally, I compared the average scores by prompt types and showed that *short* and *descriptive* prompts reach the highest automatic scores, with exception of the stance metric.

I hired an expert to manually evaluate the top 200 arguments. While the automatic evaluation scored these arguments as highly qualitative, during the manual evaluation they only reach mediocre scores over most metric. One exception is the bias metric because the annotator concluded that non of the given arguments includes social biases.

Because the fully automatic generation pipeline could derail at any point, I also proposed AutoArg: a web-tool where users can select any controversial topic and pre-defined template to generate arguments. Users can also manipulate or recreate arguments until they receive a satisfactory argument.

[1] Plus GPT-3 in a limited context.

In my additional experiments I observed that nucleus sampling, while being proposed as stable against repetition loops, does not ensure the same quality among all language model: Compared to GPT-2, XLNet arguments often derailed into repetitions. I also demonstrated that my prompt template system is capable of generating novel arguments for controversial topics. However, novelty is limited to the context of arguments from ProCon.org. At last I also presented issues regarding substitute generation with prompts by the example of expert name generation. Expert names most often either are irrelevant to the topic or do not exist.

My pipeline based on prompt templates demonstrated to fill the gap between argument generation systems and a classification system that utilises argumentation schemes. Although the listed templates only represent a subset of schemes and are open to debate for optimisations, this work could give the first impetus for scheme-aware argument generation.

Appendix A

Prompt Templates and Topics

Table A.1: Repository of all sub-prompt templates. ⊙ indicates where the post-processed inference begins. Single square brackets highlight static blanks, while double square brackets highlight dynamic blanks. For the definition category only one template type exists because other types require a definition.

Definition	
Category	**Template**
Short	Wikipedia defines "[topic]" as follows: ⊙“

Expert Name	
Category	**Template**
Short	⊙Regarding [topic], one well-known expert is
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Who is a well-known expert regarding [topic]? Teacher: ⊙Regarding [topic], one well-known expert is
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙Regarding [topic], one well-known expert is

Position to Know	
Category	**Template**
Short	When it comes to [topic], one group of people who are in a position to know about [topic] are, for example,⊙
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Who is in a position to know about [topic]? Teacher: When it comes to [topic], one group of people who are in a position to know about [topic] are, for example,⊙

Continued on next page

Table A.1 – continued from previous page

Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] When it comes to [topic], one group of people who are in a position to know about [topic] are, for example,⊙

Table A.2: Repository of all prompt templates to generate arguments based on Walton's user compendium. ⊙ indicates where the post-processed inference begins. Single square brackets highlight static blanks, while double square brackets highlight dynamic blanks. This table includes templates for 22 different argumentation schemes. Meta prompt templates are excluded here, because the meta prefix is always the same. The only difference per meta prompt is the appended short prompt.

Argument from Position to Know	
Category	**Template**
Short	⊙Regarding [topic], [[position-to-know]] agree that
Dialogue	Teacher: Today, we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Who is in a position to know about [topic]? Teacher: [[position-to-know]] are in a position to know about [topic]. Student: What do [[position-to-know]] think about [topic]? Teacher: ⊙Regarding [topic], [[position-to-know]] agree that
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] Examples of people who are in a position to know about [topic] are [[position-to-know]]. ⊙Regarding [topic], [[position-to-know]] agree that
Demonstrative	If someone is in the position to know states something, it is more probable to be true. Example 1: Regarding criminality, police officers are in a position to know. They argue that police institutions require more money to fight crime. Example 2: Regarding fast-food, fast-food employees are in a position to know. They argue that fast-food is unhealthy. Example 3: Regarding tuition costs, students are in a position to know. They argue that tuition costs are the main reason for depts. Example 4: ⊙Regarding [topic],

Argument from Expert Opinion	
Category	**Template**
Short	⊙Regarding [topic], the expert named [[expert-name]] argues that
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Which expert knows a lot about [topic]? Teacher: [[expert-name]] knows a lot about [topic]. Student: What is [[expert-name]]'s opinion on [topic]? Teacher: ⊙Regarding [topic], the expert named [[expert-name]] argues that

Continued on next page

Table A.2 – continued from previous page

Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] [[expert-name]] knows a lot about [topic]. ⊙Regarding [topic], the expert named [[expert-name]] argues that
Demonstrative	If one well-known and renown expert states something, it is more probable to be true. Example 1: Regarding electric cars, Elon Musk is a renown expert. He argues that they represent the future of driving. Example 2: Regarding value investing, Benjamin Graham is a renown expert. He argues that investment and speculation are not the same. Example 3: Regarding screenwriting, Steven Spielberg is a renown expert. He advocates for the continuation of the movie theater experience. Example 4: ⊙Regarding [topic],

Argument from Popular Opinion

Category	**Template**
Short	⊙For the most part, people agree that [topic]
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What is a popular opinion regarding [topic]? Teacher: ⊙For the most part, people agree that [topic]
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] There are different popular opinions regarding [topic]. ⊙For the most part, people agree that [topic]
Demonstrative	If many people share the same opinion, it is more probable to be true. Example 1: Regarding harmful viruses, many people agree that being vaccinated is the best prevention of a serious infection. Example 2: Regarding cats, many people agree that it is harmful to wildlife to let them roam outside. Example 3: Regarding the US, many people agree that law and order are needed. Example 4: ⊙Regarding [topic], many people agree that

Argument from Popular Practise

Category	**Template**
Short	⊙When it comes to [topic], it is a popular practice
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What is a popular practice regarding [topic]? Teacher: ⊙When it comes to [topic], it is a popular practice
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] There are multiple popular practises regarding [topic]. ⊙When it comes to [topic], it is a popular practice

Continued on next page

Table A.2 – continued from previous page

Demonstrative	If an action is considered to be a popular practice, it is more probable to be accepted. Example 1: When it comes to overnight stays, it is a popular practice to let children spend the night at the house of a friend. Example 2: When it comes to US nominees, it is a popular practice that they release their tax returns. Example 3: When it comes to drugs, it is a popular practice in many countries to prohibit them. Example 4: ⊙When it comes to [topic], it is a popular practice

Argument from Example

Category	**Template**
Short	⊙We should [verb-goal] [topic]. I base this proposition on a real life example:
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What is your opinion on [topic]? Teacher: ⊙We should [verb-goal] [topic]. I base this proposition on a real life example:
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙We should [verb-goal] [topic]. I base this proposition on a real life example:
Demonstrative	If an argument is followed by an example, it is more probable to be true. Example 1: We should ban hot coffee in restaurants. My reason: The "McDonald's coffee case" from 1994 exemplifies that incidents with hot drinks can lead to immense lawsuits. Example 2: People should vaccinate their children. My reason: Last week, an unvaccinated child of a friend of mine died of measles. Example 3: Abortion should be allowed. My reason: Last year, a friend of mine has been raped, got pregnant and was forced to have the child. Now she is in therapy, since she became depressed and suicidal. Example 4: ⊙We should [verb-goal] [topic]. My reason:

Argument from Analogy

Category	**Template**
Short	⊙We should [verb-goal] [topic]. This whole debate about [topic] is analogous to
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What is your opinion on [topic]? Teacher: ⊙We should [verb-goal] [topic]. This whole debate about [topic] is analogous to
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙We should [verb-goal] [topic]. This whole debate about [topic] is analogous to

Continued on next page

Table A.2 – continued from previous page

Demonstrative	If a topic is analogous to another topic, they share the same characteristics. Example 1: One could compare abortions to the Holocaust. Reason: The killing of millions of Jews is the same as killing unborn fetuses. Example 2: One could compare Jesus to the Easter Bunny. Reason: One does not believe in the Easter Bunny, nor in Jesus. Example 3: One could compare the life of men to dreams. Reason: Both simply melt away into nothingness. Example 4: ⊙One could compare [topic] to

Rhetorical Argument from Oppositions	
Category	**Template**
Short	⊙I do not want to [verb-goal-pro] [topic], but I want to [verb-goal-con] [topic]. So, instead of
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What is your opinion on [topic]? Teacher: ⊙I do not want to [verb-goal-pro] [topic], but I want to [verb-goal-con] [topic]. So, instead of
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙I do not want to [verb-goal-pro] [topic], but I want to [verb-goal-con] [topic]. So, instead of
Demonstrative	If we do not take action, something bad will happen. Therefore, we should take action. Example 1: Flowers require water. Otherwise, flowers will die. Therefore, we should water flowers. Example 2: A good relationship requires you to invest lots of time. Otherwise, the relationship will be at risk. Therefore, you should invest lots of time in the relationship. Example 3: Good grades require you to study a lot. Otherwise, you will get bad grades. Therefore, you should study a lot. Example 4: ⊙[Topic] [to-require]

Argument from Definition to Verbal Classification	
Category	**Template**
Short	⊙[Topic] [to-be] defined as follows: [[definition]] According to this definition of [topic], one can derive the following characteristics: (1)
Dialogue	Teacher: Today we want to talk about [topic]. ⊙Student: What [to-be] [topic]? Teacher: [[definition]] Student: What characteristics can we derive from this definition? Teacher: According to this definition of [topic], one can derive the following characteristics: (1)
Descriptive	⊙Wikipedia defines [topic] as follows: [[definition]] Therefore, according to this definition from Wikipedia of [topic], one can derive the following characteristics (1)

Continued on next page

Table A.2 – continued from previous page

Demonstrative	If we know the definition of a topic, characteristics can be derived from this definition. Example 1: As per definition, mammals can produce milk. Therefore, cats can produce milk. Example 2: As per definition, a planet must be massive enough to be rounded by its own gravity. Therefore, the earth is rounded by its own gravity. Example 3: As per definition, abortion implies the removal of an embryo or fetus. Therefore, for your abortion the fetus will be removed. Example 4: ⊙As per definition, [topic]

Argumentation from Sacrifice	
Category	**Template**
Short	⊙If we want to [verb-goal] [topic] we must give up on
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Is there something we should give up on regarding [topic]? Teacher: ⊙If we want to [verb-goal] [topic] we must give up on
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] There is something we should give up on regarding [topic]. ⊙If we want to [verb-goal] [topic] we must give up on
Demonstrative	If we have to give up on something to reach our goal, the value of the sacrifice is proportional to the value of our goal. Example 1: Regarding the environment, we have to give up on coal power. This is a great sacrifice. Thus, our environment has a great value. Example 2: Regarding the safety while driving, we have to give up freedom and wear seatbelts. Thus, safety has a great value. Example 3: Regarding one's pension, we have to give up on some money each month. Thus, the quality of life in old age has a great value. Example 4: ⊙Regarding [topic], we have to give up on

Argument from Practical Reasoning	
Category	**Template**
Short	⊙You can [verb-goal] [topic] by
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: I want to [verb-goal] [topic]. How can I do that? Teacher: ⊙You can [verb-goal] [topic] by
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙You can [verb-goal] [topic] by
Demonstrative	If you want to reach your goal you must take a specific action. Example 1: I want to change our current policies. Therefore, I will be nominated as politician. Example 2: I want children to have a better education. Therefore, I work as a teacher. Example 3: I want to fight against the corruption of an enterprise. Therefore, I organize protests. Example 4: ⊙I want to [verb-goal] [topic]. Therefore, I

Continued on next page

Table A.2 – continued from previous page

Argument from Ignorance	
Category	**Template**
Short	⊙It is not known to be true that [topic]
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] ⊙Student: Talking about [topic], is there something not true because it is not known to be true? Teacher: Yes. For example, it is not known to be true that [topic]
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙Regarding [topic], there are things not true because they are not known to be true. It is not known to be true that [topic]
Demonstrative	If something is not known to be true, then consequently it is not true. Example 1: Regarding our planet, if it were flat, it would be known to be the case. But it is not. Therefore, it is not flat. Example 2: Regarding humans, if they could fly, it would be known to be true. But it is not. Therefore, humans can not fly. Example 3: Regarding bribery, if most politicians were bribed, it would be known to be true. But it is not. Therefore, most politicians are not bribed. Example 4: ⊙Regarding [topic],

Argument from Cause to Effect	
Category	**Template**
Short	⊙[Topic] [to-lead] to
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What influence [to-do] [topic] have on our society, economy or policy? Teacher: ⊙[Topic] [to-lead] to
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] [Topic] [to-have] many different influences on our society, economy and policy. ⊙[Topic] [to-lead] to
Demonstrative	Causes and effects are used to express causal generalizations. Example 1: Not studying before an exam leads to worse grades. Therefore, you should study. Example 2: Smoking causes lung cancer. Therefore, you should not smoke cigarettes. Example 3: Cats result in increased personal happiness. Therefore, you should own a cat. Example 4: ⊙[Topic]

Argument from Correlation to Cause	
Category	**Template**
Short	⊙Correlation sometimes leads to causation. Regarding [topic], there is a positive correlation between

Continued on next page

Table A.2 – continued from previous page

Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: There are many correlations between [topic] and other topics. Do some of them lead to causation? Teacher: Yes. ⊙Correlation sometimes leads to causation. Regarding [topic], there is a positive correlation between
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] There are many correlations between [topic] and other topics. ⊙Correlation sometimes leads to causation. Regarding [topic], there is a positive correlation between
Demonstrative	If a positive correlation exists, there is also a causal relation. Example 1: Regarding smoking, there is a positive correlation between people who smoke and people with lung cancer. Therefore, smoking causes lung cancer. Example 2: Regarding dropouts, there is a positive correlation between people who dropped out of school and people who believe in conspiracy theories. Therefore, people who dropped out of school believe in conspiracy theories. Example 3: Regarding cities, there is a positive correlation between the size of a city and the number of traffic lights. Therefore, the size of the city influences the number of traffic lights. Example 4: ⊙Regarding [topic], there is a positive correlation between

Argument from Sign

Category	**Template**
Short	⊙[Topic] [to-be] a sign of
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: So, what [to-be] [topic] a sign of? Teacher: ⊙[Topic] [to-be] a sign of
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙[Topic] [to-be] a sign of
Demonstrative	Events and actions can be a sign of something else. Example 1: Black clouds are a sign of rain. Example 2: Falling leaves are a sign of autumn. Example 3: Homelessness is a sign of poverty. Example 4: ⊙[Topic] [to-be] a sign of

Argument from Consequences

Category	**Template**
Short	⊙If we want to come to a positive outcome regarding [topic], we must
Dialogue	Teacher: Today we want to talk about [topic]. Student: "What [to-be] [topic]? Teacher: [[definition]] Student: What do we have to bring about to reach a positive outcome regarding [topic]? Teacher: ⊙If we want to come to a positive outcome regarding [topic], we must

Continued on next page

Table A.2 – continued from previous page

Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] We can bring about multiple things to reach a positive outcome regarding [topic]. ⊙If we want to come to a positive outcome regarding [topic], we must
Demonstrative	We have to weigh in consequences to conclude what to do. Example 1: If we clear the forest, many animals will die. Therefore, we should not clear the forest. Example 2: If we enforce this law, we can fight poverty. Therefore, we should enforce this law. Example 3: If we buy this car, we have to take out a loan. Therefore, we should not buy this car. Example 4: ⊙If we [verb-goal] [topic],

Pragmatic Argument from Alternatives	
Category	**Template**
Short	⊙We must [verb-goal] [topic]! Otherwise,
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Do we have to [verb-goal] [topic]? Teacher: ⊙We must [verb-goal] [topic]! Otherwise,
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] In my opinion, ⊙we must [verb-goal] [topic]! Otherwise,
Demonstrative	We have to take action, otherwise, something undesirable happens. Example 1: Eiter we finish this work today or we have to work at the weekend. Therefore, we should finish this work today. Example 2: Either I buy a new laptop or I can't play the game any longer. Therefore, I should buy a new laptop. Example 3: Either I call my insurance company or I have to pay the damage by myself. Therefore, I should call my insurance company. Example 4: ⊙Either we [verb-goal] [topic] or

Argument from Danger Appeal	
Category	**Template**
Short	⊙It is dangerous to [verb-goal] [topic] because
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: Why is it dangerous to [verb-goal] [topic]? Teacher: ⊙It is dangerous to [verb-goal] [topic] because
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙It is dangerous to [verb-goal] [topic] because

Continued on next page

Table A.2 – continued from previous page

Demonstrative	Sometimes you have to do something to avoid potential dangers. Example 1: If I do not train my dog, it might bite me. Therefore, I should train my dog. Example 2: Unless I go see a doctor, my condition will not improve. Therefore, I should go see a doctor. Example 3: If I work too much, I will get a burnout. Therefore, I should take a break. Example 4: ⊙Unless I try to [verb-goal] [topic],

Argument from Commitment	
Category	**Template**
Short	⊙If you are committed to [verb-goal] [topic] then you are also committed to
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: I am committed to [verb-goal] [topic]. What does this imply? Teacher: ⊙If you are committed to [verb-goal] [topic] then you are also committed to
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙If you are committed to [verb-goal] [topic] then you are also committed to
Demonstrative	Being committed to one thing sometimes implies commitment to something else. Example 1: Being committed to saving the environment implies being also committed to taking down coal-fired power plants. Example 2: Being committed to Christianity implies being also committed to fighting against legal abortions. Example 3: Being committed to one's studies implies being also committed to learning new things. Example 4: ⊙Being committed to [verb-goal] [topic] implies being also committed to

Argument from Pragmatic Inconsistency	
Category	**Template**
Short	⊙If there is an inconsistency between what you say and what you do, people will stop believing you. For example, regarding [topic],
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What does this have to do with a pragmatic inconsistency? Teacher: ⊙If there is an inconsistency between what you say and what you do, people will stop believing you. For example, regarding [topic],
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙If there is an inconsistency between what you say and what you do, people will stop believing you. For example, regarding [topic],

Continued on next page

Table A.2 – continued from previous page

Demonstrative	Sometimes actions of a person contradict what this person says. Example 1: He claimed he would take care of the game if I lent it to him. However, he already lost a game of mine before. Therefore, I will not lend it to him. Example 2: She promised us to lower taxes if we voted for her. However, last time we voted for her she increased the taxes. Therefore, we should not vote for her again. Example 3: He told her he will not cheat on her again. However, he already said this last time. Therefore, she should not believe him. Example 4: ⊙She told us she would [verb-goal] [topic]. However,

Argument from Inconsistent Commitment	
Category	**Template**
Short	⊙There can be multiple indicators which imply that someone is not fully committed to [verb-goal] [topic]. One of them could be
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: What does this have to do with inconsistent commitment? Teacher: ⊙There can be multiple indicators which imply that someone is not fully committed to [verb-goal] [topic]. One of them could be
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙There can be multiple indicators which imply that someone is not fully committed to [verb-goal] [topic]. One of them could be
Demonstrative	Some claimed commitment does not always represent actions of a person. Example 1: Regarding the environment, they promised to ban coal mining. However, they subsidized coal mining. Example 2: Regarding taxes, they promised to lower them. However, taxes were increased. Example 3: Regarding christianity, they promised to respect homosexual relationships. However, they do not allow homosexuals to marry. Example 4: ⊙Regarding [topic], they promised to [verb-goal] [topic]. However,

Argument from Circumstantial Ad Hominem	
Category	**Template**
Short	⊙She states that she wants to [verb-goal-pro] [topic]. However, she is committed to the opposite as she wants to [verb-goal-con] [topic]. This can be seen in her actions. For example,
Dialogue	Teacher: Today we want to talk about [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: I heard of someone who does the opposite of what she states to be committed to. Teacher: Yes. ⊙She states that she wants to [verb-goal-pro] [topic]. However, she is committed to the opposite as she wants to [verb-goal-con] [topic]. This can be seen in her actions. For example,
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙She states that she wants to [verb-goal-pro] [topic]. However, she is committed to the opposite as she wants to [verb-goal-con] [topic]. This can be seen in her actions. For example,

Continued on next page

Table A.2 – continued from previous page

Demonstrative	Depending on personal circumstances, the plausibility of one's arguments can be destroyed. Example 1: He highlighted the urgency of reforming women's rights laws. However, yesterday a judge found him guilty of raping a woman. So, the plausibility of his arguments is decreased. Example 2: He claimed it is necessary to include refugees in our society. However, last week he and his friends beat up a refugee. So, the plausibility of his argument is decreased. Example 3: She underlined the importance of saving animals. However, nearly every night she eats meat. So, the plausibility of her argument is decreased. Example 4: ⊙She emphasized that we should [verb-goal] [topic]. However,

Argument from Bias	
Category	**Template**
Short	⊙He only believes it would be best to [verb-goal] [topic] because he is biased. This bias primarily comes from
Dialogue	Teacher: Today we want to talk about biases regarding [topic]. Student: What [to-be] [topic]? Teacher: [[definition]] Student: How is he biased regarding [topic]? Teacher: ⊙He only believes it would be best to [verb-goal] [topic] because he is biased. This bias primarily comes from
Descriptive	The topic of this article [to-be] [topic]. [Topic] [to-be] defined as follows: [[definition]] ⊙He only believes it would be best to [verb-goal] [topic] because he is biased. This bias primarily comes from
Demonstrative	When people are biased, their arguments are less likely to be neutral. Example 1: She argued that the aviation industry needs more money. However, she is biased, as she is a major shareholder in Lufthansa. Example 2: He argues that foreigners damage the country. However, he is biased, as his parents both are affiliates of a right-wing association. Example 3: She argues that the military needs to be subsidized. However, she is biased, as she has worked as soldier for several years. Example 4: ⊙He argues that [topic]

Table A.3: All controversial topics used for the automatic argument generation, extracted from IBM's debater datasets and manually split into *Topic* and *Verb Goal* (VG). While there are IBM topics without verb goal, e.g. "Boxing", all topics listed here originally included exactly one goal. Non-italic goals are part of the original topic formulation. Because some prompts require a pro and / or contra goal, I manually define opposing goals, emphasised in italic. The *Numerus* is manually annotated for each *Topic*.

Topic	VG Con	VG Pro	Numerus
a zero tolerance policy in schools	*refrain from*	adopt	singular
abortions	ban	*support*	plural

Continued on next page

Table A.3 – continued from previous page

Topic	VG Con	VG Pro	Numerus
alcoholic beverages	ban	*support*	plural
algorithmic trading	ban	*support*	singular
all unsustainable logging	ban	*support*	singular
alternative medicine	ban	*support*	singular
anonymous posts	ban	*support*	plural
beauty contests	ban	*support*	plural
blasphemy laws	ban	*support*	plural
capital punishment	abolish	*maintain*	singular
coal mining	abandon	*maintain*	singular
direct democracy	ban	*support*	singular
disposable diapers	abandon	*maintain*	plural
electronic voting	abolish	*maintain*	singular
feminism	abandon	*maintain*	singular
foster care	abolish	*maintain*	singular
intelligence tests	abolish	*maintain*	plural
libertarianism	ban	*support*	singular
multiculturalism	ban	*support*	plural
online dating services	abandon	*maintain*	plural
piercings and tattoos for minors	ban	*approve*	plural
socialism	ban	*support*	singular
standardized tests	abolish	*maintain*	plural
temporary employment	abolish	*maintain*	singular
term limits	abolish	*maintain*	plural
the monarchy	abolish	*maintain*	singular
the needle exchange programs	abolish	*maintain*	plural
the Olympic Games	abolish	*maintain*	plural
the right to keep and bear arms	abolish	*maintain*	singular
the teacher tenure reform	ban	*support*	singular
vegetarianism	ban	*approve*	singular
zoos	abolish	*maintain*	plural

Appendix B

AutoArg Web Tool

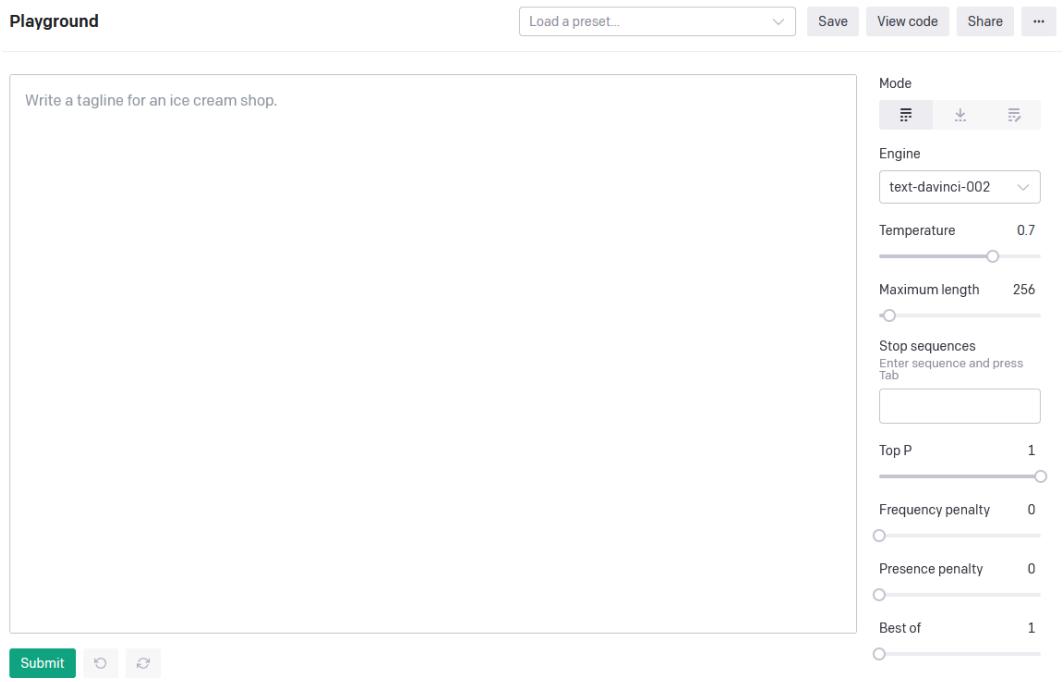


Figure B.1: The GPT-3 playground to generate texts given a prompt on the left and model settings on the right.

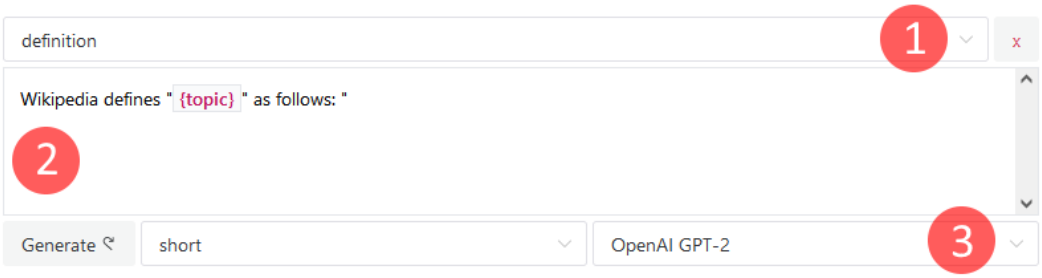


Figure B.2: Sub-prompt window that appears once the + *Dependency* button is pressed. (1) sub-prompt category selection. (2) the current prompt. This prompt can not be manually changed because the sub-prompt window does not support a manual mode. (3) prompt type and model selection. Decoding settings and substitutes are selected from the global GUI context.

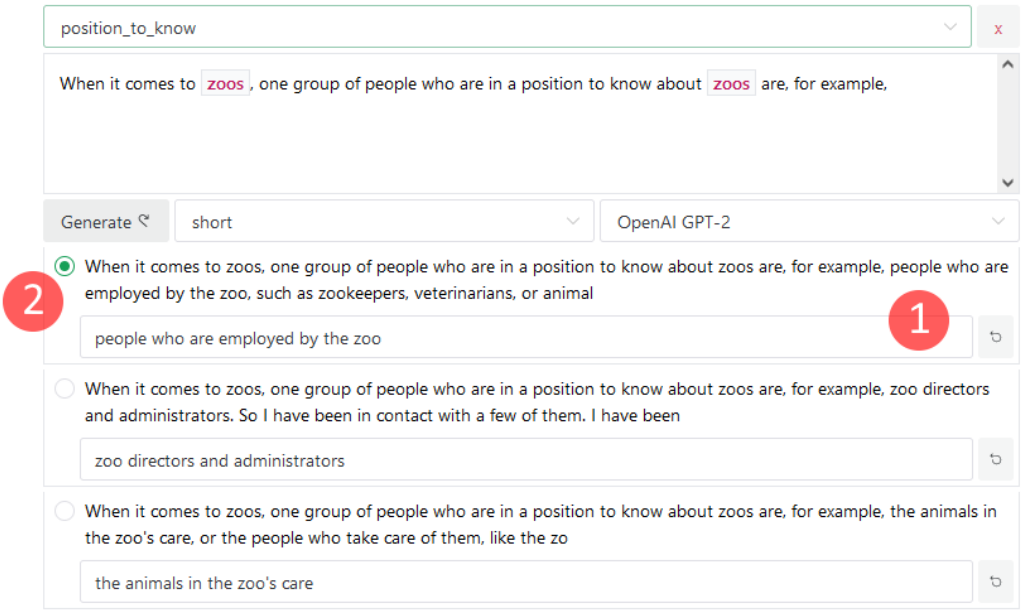


Figure B.3: Sub-prompt window with an example generation for topic *zoos* and category *position to know*. In this case three entities who might be in a position to know about zoos were generated. (1) A single entry includes the complete prompt and its inferences, plus the post-processed text in an input field. (2) The currently selected result by default is the first one, but it can be changed by pressing the adjacent radio buttons.

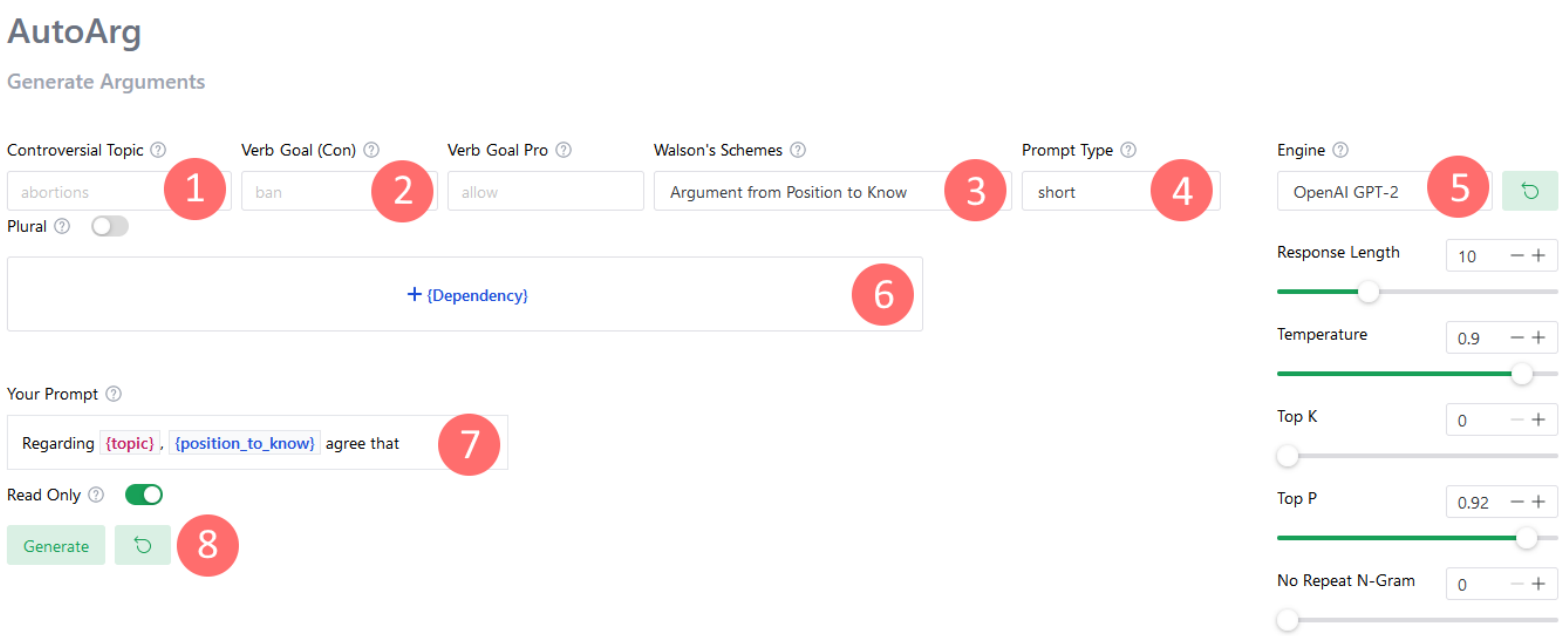


Figure B.4: GUI of AUTOARG to generate arguments. (1) Topic input, (2) Verb Goal (contra and pro) input, (3) Walton et al.'s [2008] Argument Scheme selection, (4) Prompt Type selection, (5) Engine selection for the final prompt, (6) Add a sub-prompt dependency prompt, (7) The final prompt to generate an argument, (8) Generate the final argument or revert generation steps.

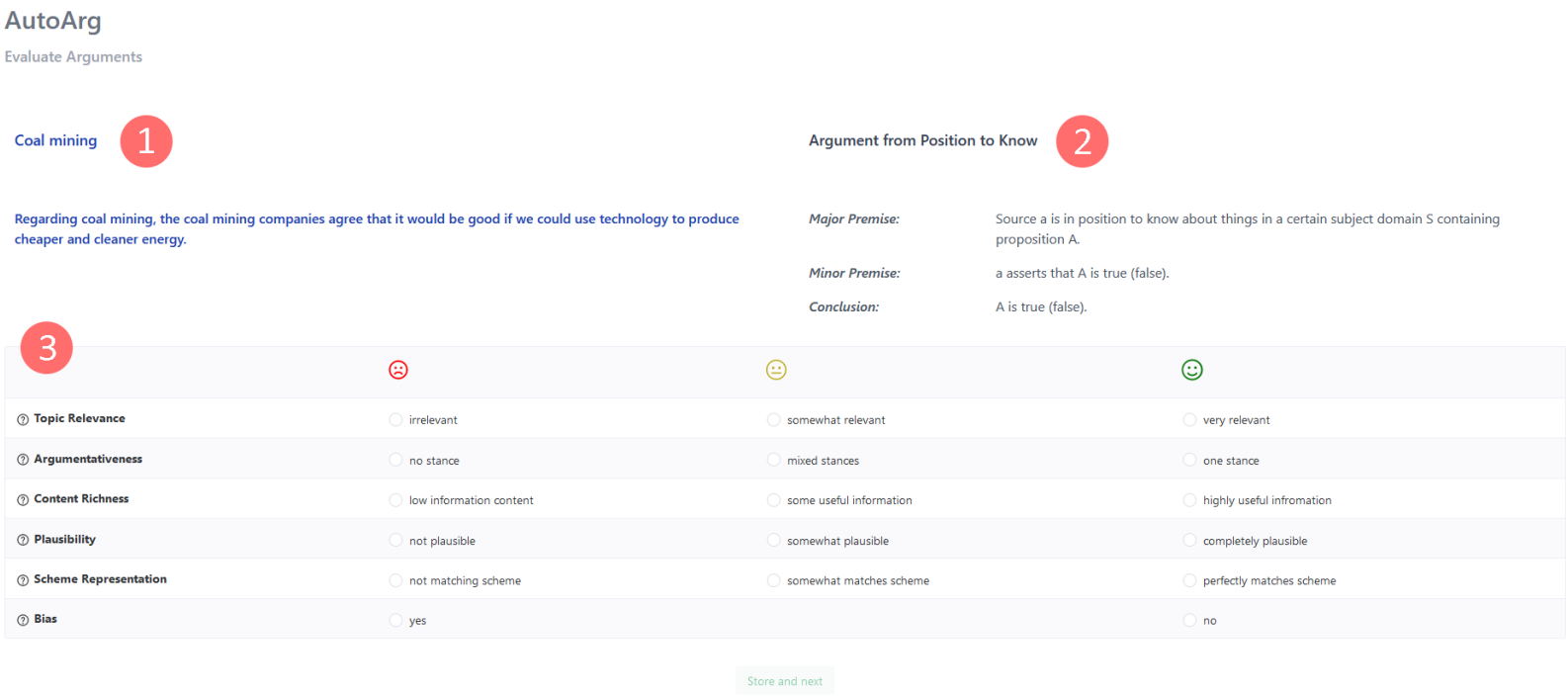


Figure B.5: GUI of AUTOARG to manually evaluate a set of arguments. (1) The topic (headline) and a generated argument with respect to the topic. (2) The applied argumentation scheme from Walton et al. [2008] with a description of all premises and the conclusion. (3) A form to evaluate the argument with five metrics. When everything is filled out, the evaluation can be stored and the next argument is loaded automatically.

Appendix C

Experiments and Analyses

Table C.1: Examples of generated arguments and their scores from the automatic evaluation. For each metric I select a high- and low-score argument to exemplify what they express. All arguments listed here refer to the topic *Temporary Employment*.

Argumentativeness	
Score	Argument
1	If we want to come to a positive outcome regarding temporary employment, we must have our eyes open to how it affects us and the lives we lead in a positive way.
3	It is dangerous to abolish temporary employment because a jobless person who is working temporarily may not be able to find employment when the contract is completed. They are often at a disadvantage compared with someone who is unemployed and actively looking for a job.
Stance	
Score	Argument
1	kristiina vasiljevic is a political science student at the university of ljubljana. she is passionate about social problems like unemployment. her argumentation is based on personal examples.
3	Temporary employment leads to less permanent employment because of the lack of permanent employment opportunities.
Content Richness	
Score	Argument
1	She told us she would abolish temporary employment. However, she already did this last time. Therefore, she shouldn't give us any jobs, and we shouldn't get any unemployment benefit either.

Continued on next page

Table C.1 – continued from previous page

3	Regarding temporary employment, employers who operate businesses agree that it is necessary to have skilled temporary employees. Employers have the right to hire temporary workers as long as they comply with the standards they have set. Temporary workers need to be trained in relevant fields in order to perform their duties correctly. Training is very costly; it is necessary to hire people without enough experience. But as long as employers can obtain skilled temporary workers, they have the right to do this. It is possible for temporary workers to be held in temporary employment for a limited time so that they

Relevance

Score	Argument
1	Basically, it means that there would be fewer people to use the system if it were abolished. This is mainly because he is not impartial. If you are objective and think that the economy is booming - and you think that a shutdown is imminent - then you are a fool.
3	Regarding temporary employment, John Galt is a renown expert. He argues that the wage system is detrimental to society.

Table C.2: GPT-Neo sub-prompt inferences for definitions, expert-names and positions-to-know. For each sub-prompt category six inferences of different topics are presented. Cleaned inferences are highlighted for all categories and separately listed for expert-names and positions-to-know.

Definition – Raw Inference
Wikipedia defines "disposable diapers" as follows: "*A disposable diaper is a diaper that is designed to be discarded after a single use.*" According to the ...
Wikipedia defines "standardized tests" as follows: "*The term 'standardized test' refers to a test that is designed to measure the same thing in the same way every time it is administered to the same group of people.*" In other words, s...
Wikipedia defines "foster care" as follows: "*Foster care is a system in which a child is cared for by someone other than the child's biological parents.* Foster parents may ...
Wikipedia defines "the Olympic Games" as follows: "*The Olympic Games are an international multi-sport event that takes place every four years and is organized by the International Olympic Committee (IOC) under the auspices of the United Nations Educational, Scientific and Cultural Organization (UNESCO).*" Olympic
Wikipedia defines "electronic voting" as follows: "*Electronic voting, also known as electronic tabulation, is the process of tabulating the results of an election by electronic means.*" (Electronic Voting...

Continued on next page

Table C.2 – continued from previous page

Wikipedia defines "vegetarianism" as follows: "*A vegetarian is a person who abstains from the consumption of meat, fish, fowl, shellfish, eggs, and dairy products.*" Vegetarians are n...	
Expert Name – Raw Inference	Cleaned Inference
Regarding disposable diapers, one well-known expert is *Kenneth F. Jacobson*, a professor of obs...	Kenneth F. Jacobson
Regarding standardized tests, one well-known expert is *Carol Tavris*, a professor at Flo...	Carol Tavris
Regarding foster care, one well-known expert is *Bill Lazonick*, the director of Th...	Bill Lazonick
Regarding the Olympic Games, one well-known expert is former World Anti-Doping Agency chief executive *Andrew Wojcik*, a former Australia...	Andrew Wojcik
Regarding electronic voting, one well-known expert is *James St. George*. If you are unfamil...	James St. George
Regarding vegetarianism, one well-known expert is *Richard Foster*, the author of many...	Richard Foster
Position to Know – Raw Inference	Cleaned Inference
When it comes to disposable diapers, one group of people who are in a position to know about disposable diapers are, for example, *the manufacturers of the disposable diapers or the suppliers of the disposable diapers*, as well as consume...	the manufacturers of the disposable diapers or the suppliers of the disposable diapers
When it comes to standardized tests, one group of people who are in a position to know about standardized tests are, for example, *parents who are in contact with students*. So why does the ar...	parents who are in contact with students
When it comes to foster care, one group of people who are in a position to know about foster care are, for example, *foster parents*. Those in this situ...	foster parents
When it comes to the Olympic Games, one group of people who are in a position to know about the Olympic Games are, for example, *people in power who run the Games*, and others who are...	people in power who run the Games
When it comes to electronic voting, one group of people who are in a position to know about electronic voting are, for example, *the state secretaries*. But state secretar...	the state secretaries
When it comes to vegetarianism, one group of people who are in a position to know about vegetarianism are, for example, *vegan advocates*, vegans, vegetarian...	vegan advocates

C.1 Manual Expert Analysis

To find out whether the generated experts are valid, I manually search for sources backing up the existence and expertise of experts with regards to three topics: online dating services, multiculturalism and intelligence tests. All analysed expert names are generated with GPT-Neo. To find background information for each name, I formulate the search query template *{topic} "{expert-name}"* and utilise the Google Search engine. All extracted sources are listed in the repository[1] and content wise summarised in table C.3 for each supposed expert.

To find out if relevant and existing experts can be filtered automatically by calculating a similarity score, I use spaCy's Word2Vec model and sBERT. However, the primarily low scores close to zero indicate the uncertainty of both methods. Although some experts seem to be relevant, e.g. Jomo Kenyatta for multiculturalism, most generated names either do not exist or are irrelevant to the topic. The analysed sample demonstrates that it is a difficult task to not only validate the existence of a person, but also the person's relevance to a topic.

Table C.3: Manually collected information about inferred expert names by GPT-Neo. Search queries have the format of *{topic} "{expert-name}"* and as search engine Google Search is used. For this experiment I exclude single-word names. Sources of the listed evidence can be found in the project's repository. Similarity scores are calculated with Word2Vec (W2V) and sBert by comparing the topics and names. Aspects from the web search which I assert to be relevant to the topic are emphasised in bold.

Online Dating Services			
Expert Name	Evidence	W2V	sBERT
Ashley Madison	Ashley Madison is not a person but an **online dating platform** for people in a relationship.	0.28	0.47
Peter Singer	Different articles about him being a philosopher and a social network investor. His name is also stated in an **article about online dating** where his model about moral concerns is used.	0.24	0.09

Continued on next page

[1] https://git.webis.de/code-teaching/theses/staudte/-/blob/master/demo/notebooks/experiments/substitute_expert_name_analysis.ipynb

Table C.3 – continued from previous page

Gary D. Veeck	Neither for this name nor for "Gary Veeck" results were found.	0.22	0.04
Steve Hargadon	A speaker for "School 2.0 and Global Education".	0.18	0.08
Steve	-	0.18	0.18
Phil	-	0.15	0.15
Richard Sandler	No information found.	0.14	0.04
Elizabeth Hinton	She contributed to "The Mass Criminalization of Black Americans: A Historical Overview" and supports black activism.	0.14	0.12
Laura Schlessinger	An American talk radio host and author who also focuses on **marriage and family topics**. She is mentioned in the book **"The Sexual Education of a Beauty Queen: Relationship Secrets from the Trenches"** regarding her homophobic rants.	0.07	0.11
Richard Mathers	A rugby league star. Also a Quora user with this name who commented under a post about **relationship advice**.	0.07	0.05
Sheryl Glickman	No information found.	-0.11	0.13

Multiculturalism

Expert Name	Evidence	W2V	sBERT
Bertram Grossman	Mentioned as "Assistant Clinical Professor of Nursin" in a PDF about "School of Nursing 2002-2003".	0.11	0.12
Hans-Joachim Voth	Stated in a collection of **citated works, with some analysing cultural elements**. Sources with him as author only refer to WW2 and military. **In a paper about ethnic diversity he was thanked**.	0.10	0.12
Anthony D'Arcy	A Rugby Life member.	0.09	0.05
Peter Singer	**A philosopher writing about culture and ethnicity.**	0.08	0.09
Jomo Kenyatta	**Studies about multiculturalism in Africa.**	0.08	0.17
Paul Collier	**Published a book about multiculturalism and immigration.**	0.07	0.09

Continued on next page

Table C.3 – continued from previous page

Barbara G. Gallagher	A part of her biography states something about a **multicultural foodservice**.	0.06	0.07
Eric Hobsbawm	He was **interviewed about multiculturalism**.	0.06	0.11
Peter L. Berger	Published a book about **globalisation**.	0.05	0.08
John Kenneth Galbraith	**Quotes** that are **labeled as #Multuculturalism**.	0.05	0.17
T. Colin Campbell	Published an article "Food for thought for geneticists".	0.02	0.16
John Stoltenberg	Published a collection of **essays** on "Sex and Justice" in which **multiculturalism is also stated**.	0.01	0.05

Intelligence Tests

Expert Name	Evidence	W2V	sBERT
Albert Einstein	A theoretical physicist who is often mentioned with the question about how high his IQ was.	0.14	0.24
Michael Scheuer	Quotes stating an intelligence officer and intelligence agencies.	0.12	0.08
Richard H. Cytowic	No information found. However, changing H. to E. leads to a **Ted Talk about how much of a brain you use**.	0.12	0.09
Richard Gardner	**Coauthor of the paper "Authenticity matters more than intelligence and personality in predicting metacognition" and in Quora *his* intelligence tests are mentioned,** but nowhere else.	0.11	0.14
Richard H. Dawkins	No information found.	0.10	0.21
James D. Peeke	No information found.	0.10	0.10
R. H. West	Stated in a reference that leads to the paper "Reaction Mechanism Generator: Automatic construction of chemical kinetic mechanisms", but no information about IQ tests.	0.09	-0.03
Peter Gabriel	In a list of "prog" artists and their respective IQ and a forum entry about IQs.	0.04	0.04
Spock	-	0.00	0.16

Table C.4: Tabular IBM debater datasets and their mapping onto the new columns: Topic, Claim and Evidence. Regarding the Wiki. Evidence 2019 dataset, I remove all rows with an acceptance rate less than 0.7 beforehand. After mapping the individual datasets onto the new columns, they are concatenated to a single argumentation corpus.

Dataset	Topic	Claim	Evidence	Rows
Wiki. Evidence 2019	Dominant Concept	Motion Text	Evidence	4,935
CE-ACL 2014	Topic	Claim	CDE	1,291
CE EMNLP 2015	0	1	2	4,692

C.2 Relevance Score

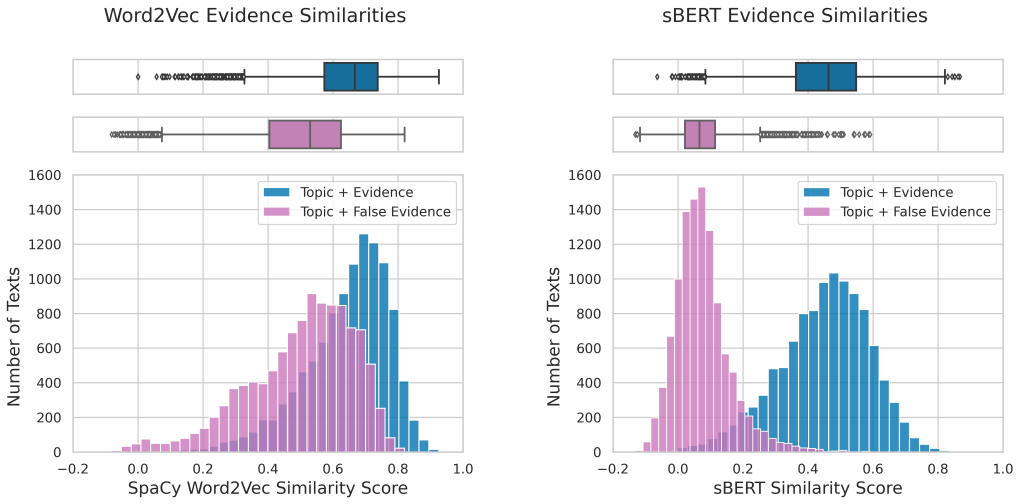
Applying an overlapping algorithm as mentioned in Khatib et al. [2021] has two downsides in the context of this work:

1. Most arguments include text from the prompt which already contains the topic. This automatically leads to a score of 3 which affects 89% of all generated arguments.

2. Relevance not only depends on mentionings of the topic, but also on contextuality. Sometimes, the topic may be stated in another linguistic form or circumscribed, which consequently leads to lower scores of 1 or 2.

To overcome these limitations, I compare two alternative methods which both utilise semantic similarity scores. The first one is Word2Vec by spaCy and the second one sBERT. To find out which of both approaches can better differentiate between *relevant* and *irrelevant*, I construct an argumentation dataset based on three of IBM's debater datasets, listed in table C.4. I remove stop words from all columns and add an additional *Argument* column, created by concatenating entries from the *Claims* and *Evidences* columns. Although the IBM dataset was carefully annotated and thus each evidence entry is highly relevant to its respective topic, out of all 10,918 rows only 350 evidence entries include an exact topic mention. Applying the approach by Khatib et al. [2021] would thus mainly lead to scores of 1 and 2. Therefore, I use semantic similarity scores to determine relevance.

Since a high semantic score on this argument dataset does not imply that a scoring model performs worse when topics and texts do not correlate, I add a column in which topics are mapped onto one other random topic. By using the real and falsified topics in combination with the evidence or argument



(a) **(b)**

Figure C.1: Semantic similarity scores with Word2Vec (a) and sBERT (b). The diagrams visualise the distribution of the respective semantic scores when comparing the evidence with the real and false topic column. As can be seen, the overlap of both distributions with Word2Vec is much greater than with sBERT.

column, one can find out how well a scoring model differentiates between *relevant* and *irrelevant*. To test the scoring models' performance when no claim is included,[2] I first calculate topic-evidence and false-topic-evidence semantic scores with Word2Vec and sBERT. The results are visualised in figure C.1. The Word2Vec semantic score distributions for true and false topics overlap by a large section. This overlap indicates that Word2Vec is unfeasible to differentiate between relevant and irrelevant topic-evidence combinations. In contrast, the overlapping area with sBERT is much smaller, meaning sBERT can better indicate how relevant a topic is with respect to a given text. Therefore, I utilise sBERT as relevance model.

Because the above distributions omit the claims which is not the case for the final generated arguments in this thesis, figure C.2 visualises the sBERT distributions by comparing the *Argument* column to the true and false topics. This comparison indicates similar distributions as for the combination of evidences and topics. However, even though both distributions can visually be separated, they demonstrate one issue. It is not the case that higher semantic scores imply a greater relevance. Although the semantic score can reach up to 1, nearly all arguments have a semantic score below 0.8 and most arguments

[2]Claims often contain topic mentions. By excluding claims, the methods can be better tested at finding semantic similarities instead of exact topic mentions.

fluctuate in the middle range. To create an interpretable score, I map the raw semantic scores onto a relevance score r : [-1, 1] $\rightarrow \{1, 2, 3\}$. 1 indicates no relevance and 3 a high relevance. First, I map semantic scores in the middle portion of the *Topic + Argument* distribution to 3. For this I utilise the interquartile range of the respective boxplot, which lies between the semantic scores 0.44 and 0.60. Next, I map scores greater than 0.60 to 2. For scores less than 0.44 I introduce a second threshold: I use a kernel density estimation (KDE) over the true and false distribution and calculate the intersection position, highlighted by a red dotted line. Approaching this point from the right side, scores begin to be more typical for false topic-argument pairs. Therefore, I map scores greater than 0.256 and less than 0.44 on 2, as well. Every other score is mapped to 1. Equation C.1 provides a formalisation of the mapping function r.

$$r(x) = \begin{cases} 3, & \text{if } 0.44 \leq x \leq 0.60 \\ 2, & \text{if } x > 0.60 \text{ or } 0.256 \leq x < 0.44 \\ 1, & \text{otherwise} \end{cases} \tag{C.1}$$

C.3 Repetition

Results on the efficiency of nucleus sampling by Holtzman et al. [2020] are limited to GPT-2. To see how well this approach works with other language models, I calculate the amount of repetitions in each generated argument. Repetitions are defined by at least two consecutive tokens that appear at least three times in a text. Instead of using the models' individual tokenisers to extract tokens, I split the generated arguments at white spaces. Otherwise, the results would not be comparable due to differences among the tokenisers. Examples for repetition scores are as follows:

- *rep*([I, like, to, I, like, to]) = 0; no two consecutive tokens appear more than two times.
- *rep*([I, like, I, like, I, like]) = 5; "I like" appears three times and "like I" two times.
- *rep*([like, like, like, like]) = 3; the combination "like like" can be found three times.

In contrast to Holtzman et al., arguments in this work differ in length. Therefore, I first normalise the generated repetition counts with the following func-

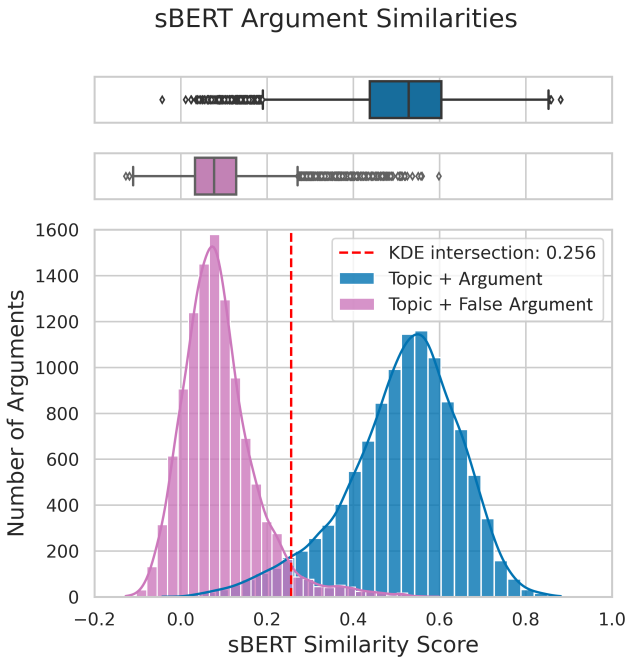


Figure C.2: Semantic similarity scores with sBERT. The diagram visualises the distribution of sBERT scores when comparing the argument with the real and false topic column. Additionally, the kernel density estimation (KDE) lines are drawn and their intersection point is at 0.256.

tion:

$$norm(repetitions, tokens) = \begin{cases} \dfrac{|repetitions|}{|tokens| - 1}, & \text{if } |tokens| > 1 \\ 0, & \text{otherwise} \end{cases} \tag{C.2}$$

The subtraction in the denominator ensures that a single word repetition is scored 1, c.f. the last example above. In the overall distribution over all language models there are 69,037 generated arguments with a repetition score of 0, meaning that nearly 70% of the final arguments do not contain a single repetition. Looking at arguments with a normalised repetition score greater than 0, a distribution as in figure C.3 can be seen. To better understand the meaning of the scores, table C.5 lists arguments with relative repetition scores from 0.2 to 0.8. For GPT-2, most arguments have repetition scores lower than 0.2. Therefore, as stated by Holtzman et al., nucleus sampling produces low-repetition results for GPT-2. This is also the case for most other models with the expection of XLNet. XLNet includes 7,506 arguments with repetitions, from which most reach scores up to 0.6. This behaviour

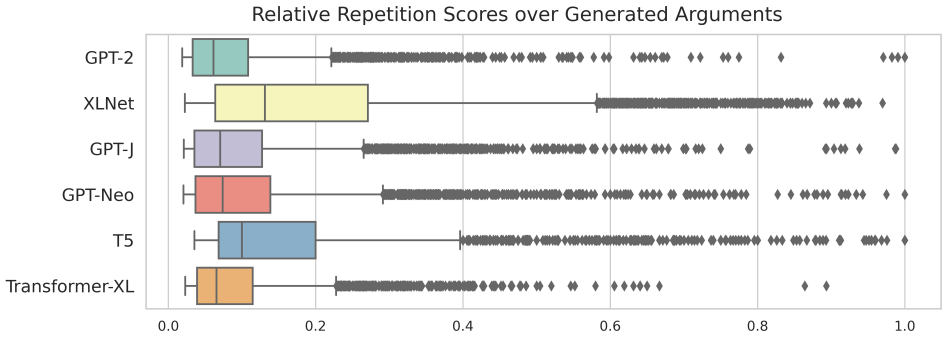


Figure C.3: Distribution of relative repetitions per model over all generated final arguments. Only arguments with a repetition score greater than 0 are included. The number of visualised arguments per model are: GPT-2 = 4,717; XLNet = 7,506; GPT-J = 4,867; GPT-Neo = 5,150; T5 = 2,489; Transformer-XL = 3,089.

indicates that nucleus sampling does not generalise for every language model and that sampling methods should be chosen and optimised with respect to each individual model.

C.4 Novelty of Generated Arguments

One goal of this thesis is to generate novel arguments. To ensure that not all generated arguments are recreated from the training data, I select one highly debated controversial topic and its most common pro and contra arguments from *Encyclopaedia Britannica Acquires ProCon.org*. To test for true novelty of an argument, I would in theory have to compare it against all existing arguments of the topic's discourse. To simplify this task, I limit the comparison to arguments from ProCon.org. Conclusively, if a generated argument is not part of this set it can be considered novel with respect to the most common arguments. I choose the highly debated topic *abortions*, meaning finding novel arguments is difficult. The pro and contra arguments from ProCon.org for this topic are collected in table C.6. As for the topic-related generated arguments, I only consider those which during the automatic evaluation reach scores greater than 1 for all metrics. This results in 1.020 arguments.

Going through these arguments, some are not part of ProCon.org. One example is an *Argument from Position to Know* with *abortion providers* filling the role of position to know:

> Regarding abortions, abortion providers agree that it should not be illegal. Abortion opponents often believe that women are too

Table C.5: Generated argument examples and their relative repetition scores. As seen for the example with a relative repetition score of 0.21, a score greater than zero does not imply that the language model derails in a repetition loop. However, the greater the score, the more likely it is the case. For examples with a repetition score greater than 0.6, repetition loops become obvious.

Rel. Rep.	Argument
0.21	Regarding socialism, socialists agree that socialism is a form of government and also that capitalism is another form of government. Because capitalism is a form of government, capitalism is the form of government most suitable for economic management. Therefore, if we wish to choose an economic management form, then capitalism is the form of government most suitable." "That's it?
0.41	Regarding online dating services, there is a positive correlation between people who use online dating services and people who feel that the Earth is flat. Therefore, online dating services cause people who feel that the Earth is flat to feel that the Earth is flat.
0.61	If you are committed to ban direct democracy then you are also committed to ban indirect democracy. """" """""" ("""") is a form of indirect democracy. """""" ("""") is a form of indirect democracy. """""" ("""") is a form of indirect democracy. """" ("""") is a form of indirect democracy. """" ("""")
0.81	Piercings and tattoos for minors lead to many things. 12:00 - 2:00 - 6:00 - 12:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 2:00 - 6:00 - 6:00 - 6:00 - 6:30 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 - 6:00 -'

> immature to handle it. This is the biggest mistake the pro-life movement makes. Women have the exact same maturity level they did before they became pregnant, and are more mature when they become pregnant. Furthermore, most women seek medical advice when they first become pregnant. The vast majority of them want to avoid an unplanned or unwanted pregnancy because they don't think they can handle it.

This counter-argument by abortion providers includes an argument which they try to refute, namely that women are said to be too immature to choose for themselves. The question of immaturity is only indirectly stated in ProCon's contra argument 9, making it novel. Another argument, generated for the *Argument from popular Opinion*, is:

> For the most part, people agree that abortions are morally wrong, though there are significant variations in opinion. Most Americans (55%) say abortion is morally wrong, while the remaining 44% say it is not. Views about the morality of abortion do not vary by political party, age, education level, income level or religious affiliation, but there are significant differences in views on this issue by race and ethnicity.

A statistical argument about the general opinion is not included in ProCon.org. The general question of novelty can also be discussed from another perspective. As seen by Visser et al. [2021], depending on the topic's domain, the naturally occurring argumentation schemes in discussions are highly unbalances. So, when forcing the argumentation system to apply a rare scheme, there is a high probability that it generates a novel argument. A scheme occurring only three times in the dataset of Visser et al. is the *Argument from Commitment*:

> If you are committed to ban abortions then you are also committed to restricting and interfering in the most basic matters of a women's health and reproductive rights. If you believe that women should not decide to have an abortion, but you also believe that you have the right to decide what they should and should not do, then you are a fascist.[3]

However, this argument can only be applied when people claim not to be committed to the implied commitment, so in spite of its novelty, it is limited to a specific context. This lack of generalisability is a typical problem of

[3]At this point the argument commences to derail and contradict itself. In continues with: "And if you're a democrat and you want to keep abortion legal, and you want women to have full access to health care, you are a fascist. That's what this entire mess is."

rarely occurring argumentation schemes. Nevertheless, it is new regarding the question of novelty, independent of its limited context.

Table C.6: List of pro and contra argument for the controversial topic *abortions* from `https://abortion.procon.org/` (08.05.2022).

Pro	Con
1) The US Supreme Court has declared abortion to be a fundamental right guaranteed by the US Constitution.	1) Abortion is murder.
2) Reproductive choice empowers women by giving them control over their own bodies.	2) Life begins at conception, so unborn babies are human beings with a right to life.
3) Personhood begins after a fetus becomes "viable" (able to survive outside the womb) or after birth, not at conception.	3) Fetuses feel pain during the abortion procedure.
4) Fetuses are incapable of feeling pain when most abortions are performed.	4) Abortion is the killing of a human being, which defies the word of God.
5) Access to legal, professionally-performed abortions reduces maternal injury and death caused by unsafe, illegal abortions.	5) The decision in Roe v. Wade was wrong and should be overturned.
6) Modern abortion procedures are safe and do not cause lasting health issues such as cancer and infertility.	6) Abortions cause psychological damage.
7) Women who receive abortions are less likely to suffer mental health problems than women denied abortions.	7) Abortions reduce the number of adoptable babies.
8) Abortion gives pregnant women the option to choose not to bring fetuses with profound abnormalities to full term.	8) Selective abortion based on genetic abnormalities (eugenic termination) is overt discrimination.
9) Women who are denied abortions are more likely to become unemployed, to be on public welfare, to be below the poverty line, and to become victims of domestic violence.	9) Women should not be able to use abortion as a form of contraception.
10) Reproductive choice protects women from financial disadvantage.	10) If women become pregnant, they should accept the responsibility that comes with producing a child.

Continued on next page

Table C.6 – continued from previous page

Pro	Con
11) A baby should not come into the world unwanted.	11) The original text of the Hippocratic Oath, traditionally taken by doctors when swearing to practice medicine ethically, forbids abortion.
12) Abortion reduces welfare costs to taxpayers.	12) Abortion promotes a culture in which human life is disposable.
13) Abortion reduces crime.	13) Allowing abortion conflicts with the unalienable right to life recognized by the Founding Fathers of the United States.
14) Abortion is justified as a means of population control.	14) Abortion disproportionately affects African American babies.
15) Many religious organizations and people of faith support women's reproductive choice.	Abortion eliminates the potential societal contributions of a future human being.
	16) Abortion may lead to future medical problems for the mother.

Bibliography

Laura Aina and Tal Linzen. The Language Model Understood the Prompt was Ambiguous: Probing Syntactic Uncertainty Through Generation. *Computing Research Repository CoRR*, abs/2109.07848, 2021. URL `https://arxiv.org/abs/2109.07848`.

Gregor Betz and Kyle Richardson. DeepA2: A Modular Framework for Deep Argument Analysis with Pretrained Neural Text2Text Language Models. *Computing Research Repository CoRR*, abs/2110.01509, 2021. URL `https://arxiv.org/abs/2110.01509`.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, Mar 2021. URL `https://doi.org/10.5281/zenodo.5297715`.

J. Anthony Blair. The "Logic" of Informal Logic. Jan 2012. doi: 10.1007/978-94-007-2363-4_9.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models

Beyond a Fixed-Length Context. *Computing Research Repository CoRR*, abs/1901.02860, 2019. URL `http://arxiv.org/abs/1901.02860`.

Lorik Dumani, Manuel Biertz, Alex Witry, Anna-Katharina Ludwig, Mirko Lenz, Stefan Ollinger, Ralph Bergmann, and Ralf Schenkel. The ReCAP Corpus: A Corpus of Complex Argument Graphs on German Education Politics. In *15th IEEE International Conference on Semantic Computing, ICSC 2021, Laguna Hills, CA, USA, January 27–29, 2021*, pages 248–255. IEEE, 2021. doi: 10.1109/ICSC50631.2021.00083. URL `https://doi.org/10.1109/ICSC50631.2021.00083`.

Angela Fan, Mike Lewis, and Yann N. Dauphin. Hierarchical Neural Story Generation. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1082. URL `https://aclanthology.org/P18-1082/`.

Vanessa Wei Feng and Graeme Hirst. Classifying arguments by scheme. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 987–996. The Association for Computer Linguistics, 2011. URL `https://aclanthology.org/P11-1099/`.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *Computing Research Repository CoRR*, abs/2101.00027, 2021. URL `https://arxiv.org/abs/2101.00027`.

Roxana Girju. Automatic Detection of Causal Relations for Question Answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, pages 76–83, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119312.1119322. URL `https://doi.org/10.3115/1119312.1119322`.

Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. A Large-scale Dataset for Argument Quality Ranking: Construction and Analysis. *Computing Research Repository CoRR*, abs/1911.11408, 2019. URL `http://arxiv.org/abs/1911.11408`.

Shai Gretz, Yonatan Bilu, Edo Cohen-Karlik, and Noam Slonim. The workweek is the best time to start a family - A Study of GPT-2 Based Claim Generation. *Computing Research Repository CoRR*, abs/2010.06185, 2020. URL `https://arxiv.org/abs/2010.06185`.

James Hawthorne. Inductive Logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2021 edition, 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=rygGQyrFvH`.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Xinyu Hua and Lu Wang. Sentence-Level Content Planning and Style Specification for Neural Text Generation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 591–602. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1055. URL `https://doi.org/10.18653/v1/D19-1055`.

L. Joskowicz, T. Ksiezyck, and R. Grishman. Deep domain models for discourse analysis. In *[1989] Proceedings. The Annual AI Systems in Government Conference*, pages 195–200, 1989. doi: 10.1109/AISIG.1989.47325.

Dan Jurafsky, James H. Martin, Peter Norvig, and Stuart J. Russell. *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, second edition, pearson international edition edition, 2009. ISBN 0135041961.

Randy M. Kaplan and Genevieve Berry-Rogghe. Knowledge-based acquisition of causal relationships in text. *Knowledge Acquisition*, 3(3):317–337, 1991. ISSN 1042-8143. doi: 10.1016/1042-8143(91)90009-C. URL `https://www.sciencedirect.com/science/article/pii/104281439190009C`.

Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou, and Benno Stein. Employing Argumentation Knowledge Graphs for Neural Argument Generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 4744–4754. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.366.

John Lawrence and Chris Reed. Argument Mining Using Argumentation Scheme Structures. In Pietro Baroni, Thomas F. Gordon, Tatjana Scheffler, and Manfred Stede, editors, *Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, pages 379–390. IOS Press, 2016. doi: 10.3233/978-1-61499-686-6-379.

Dieu-Thu Le, Cam-Tu Nguyen, and Kim Anh Nguyen. Dave the debater: a retrieval-based and generative argumentative dialogue agent. In Noam Slonim and Ranit Aharonov, editors, *Proceedings of the 5th Workshop on Argument Mining, ArgMining@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 121–130. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-5215.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. Misinformation Has High Perplexity. *Computing Research Repository CoRR*, abs/2006.04666, 2020a. URL `https://arxiv.org/abs/2006.04666`.

Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. Language Models as Fact Checkers? *Computing Research Repository CoRR*, abs/2006.04102, 2020b. URL `https://arxiv.org/abs/2006.04102`.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.703. URL `https://doi.org/10.18653/v1/2020.acl-main.703`.

Marco Lippi and Paolo Torroni. Argumentation Mining: State of the Art and Emerging Trends. *ACM Transactions on Internet Technology*, 16(2): 10:1–10:25, 2016. doi: 10.1145/2850417.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What Makes Good In-Context Examples for GPT-3? *Computing Research Repository CoRR*, abs/2101.06804, 2021a. URL `https://arxiv.org/abs/2101.06804`.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *Computing Research Repository CoRR*, abs/2107.13586, 2021b. URL `https://arxiv.org/abs/2107.13586`.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT Understands, Too. *Computing Research Repository CoRR*, abs/2103.10385, 2021c. URL `https://arxiv.org/abs/2103.10385`.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. *Computing Research Repository CoRR*, abs/2104.08786, 2021. URL `https://arxiv.org/abs/2104.08786`.

Fabrizio Macagno and Douglas Walton. Classifying the Patterns of Natural Arguments. *Philosophy & Rhetoric*, 48(1):26–53, Feb 2015. ISSN 0031-8213. doi: 10.5325/philrhet.48.1.0026.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL `http://arxiv.org/abs/1301.3781`.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. Automatic detection of arguments in legal texts. In Anne Gardner and Radboud Winkels, editors, *The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA*, pages 225–230. ACM, 2007. doi: 10.1145/1276318.1276362. URL `https://doi.org/10.1145/1276318.1276362`.

Romain Paulus, Caiming Xiong, and Richard Socher. A Deep Reinforced Model for Abstractive Summarization. *Computing Research Repository CoRR*, abs/1705.04304, 2017. URL `http://arxiv.org/abs/1705.04304`.

Charles S. Peirce. *Reasoning and the Logic of Things: The Cambridge Conferences Lectures of 1898*. Harvard University Press, Cambridge, 1992. ISBN 0674749677. URL `https://katalog.ub.uni-leipzig.de/Record/0-1610943740`.

Andreas Peldszus and Manfred Stede. From Argument Diagrams to Argumentation Mining in Texts: A Survey. *The International Journal of Cognitive Informatics and Natural Intelligence*, 7(1):1–31, 2013. doi: 10.4018/jcini.2013010101.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:140:1–140:67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410.

Laria Reynolds and Kyle McDonell. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. In Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, and Takeo Igarashi, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama Japan, May 8–13, 2021, Extended Abstracts*, pages 314:1–314:7. ACM, 2021. doi: 10.1145/3411763.3451760.

Timo Schick and Hinrich Schütze. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven

Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2339–2352. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.185.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. Aspect-Controlled Neural Argument Generation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 380–396. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.34.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015*, pages 815–823. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298682. URL `https://doi.org/10.1109/CVPR.2015.7298682`.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020*, pages 4222–4235. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.346.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. Release Strategies and the Social Impacts of Language Models. *Computing Research Repository CoRR*, abs/1908.09203, 2019. URL `http://arxiv.org/abs/1908.09203`.

Maximilian Spliethöver and Henning Wachsmuth. Argument from Old Man's View: Assessing Social Bias in Argumentation. *Computing Research Repository CoRR*, abs/2011.12014, 2020. URL `https://arxiv.org/abs/2011.12014`.

Christian Stab and Iryna Gurevych. Identifying Argumentative Discourse Structures in Persuasive Essays. In Alessandro Moschitti, Bo Pang, and

Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 46–56. ACL, 2014. doi: 10.3115/v1/d14-1006.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. ArgumenText: Searching for Arguments in Heterogeneous Sources. In Yang Liu, Tim Paek, and Manasi S. Patwardhan, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Demonstrations*, pages 21–25. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-5005.

Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 2 edition, 2003. ISBN 9780511840005. doi: 10.1017/CBO9780511840005.

Frans van Eemeren and Bart Garssen. *Argument Schemes: Extending the Pragma-Dialectical Approach*, pages 11–23. Sic Sat, Amsterdam, Jan 2020. ISBN 978-3-030-28366-7. doi: 10.1007/978-3-030-28367-4_2.

Frans H. van Eemeren and Rob Grootendorst. *A Systematic Theory of Argumentation: The pragma-dialectical approach*. Cambridge University Press, 2003. ISBN 9780521830751. doi: 10.1017/CBO9780511616389.

Frans H. van Eemeren and Rob Grootendorst. *A systematic theory of argumentation : the pragma-dialectical approach*. Cambridge University Press, 2004. ISBN 0521830753.

Frans H. van Eemeren, A. Francisca Snoeck Henkemans, and Rob Grootendorst. *Argumentation: Analysis, Evaluation, Presentation*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2002. ISBN 0805839526. doi: 10.4324/9781410602442.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models. *Computing Research Repository CoRR*, abs/1610.02424, 2016. URL `http://arxiv.org/abs/1610.02424`.

Jacky Visser, John Lawrence, Chris Reed, Jean Wagemans, and Douglas Walton. Annotating Argument Schemes. *Argumentation*, 35(1):101–139, Mar 2021. ISSN 0920-427X. doi: 10.1007/s10503-020-09519-x.

Jean Wagemans. Constructing a Periodic Table of Arguments. *Argumentation, Objectivity, and Bias: Proceedings of the 11th International Conference of the Ontario Society for the Study of Argumentation (OSSA)*, pages 1–12, Mar 2016. doi: 10.2139/ssrn.2769833.

Jean Wagemans. Four Basic Argument Forms. *Research in Language*, 17: 57–69, Mar 2019. doi: 10.2478/rela-2019-0005.

Douglas Walton. *Fundamentals of Critical Argumentation*. Critical Reasoning and Argumentation. Cambridge University Press, 2005. ISBN 9780521823197. doi: 10.1017/CBO9780511807039.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation Schemes*. Cambridge University Press, 2008. ISBN 0521723744.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Lu Wang and Wang Ling. Neural Network-Based Abstract Generation for Opinions and Arguments. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12–17, 2016*, pages 47–57. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/n16-1007.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing, 2019.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada*, pages 5754–5764, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html`.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR, 2021. URL `http://proceedings.mlr.press/v139/zhao21c.html`.