Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Medieninformatik

Exploiting Argumentation Knowledge Graphs for Argument Generation

Bachelor's Thesis

Trautner Lukas

1. Referee: Prof. Dr. Benno Stein
2. Referee: jun.-Prof. Dr. Jan Ehlers

Submission date: March 25, 2020

Declaration

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

Weimar, March 25, 2020

.....
Trautner Lukas

Abstract

This thesis presents a new approach for generating arguments based on knowledge encoded in an argumentation knowledge graph. First, using a manually annotated argumentation corpus, we construct a new argumentation graph according to a model tailored for argument generation. The graph is completed by identifying its implicit knowledge. Second, we develop a language generation model for transforming knowledge from that graph into a natural language argument. Our approach is evaluated by four semi-expert annotators. The results show a high quality of our generated arguments, particularly, in their fluency and informativeness.

Contents

Chapter 1

Introduction

Argumentation is a fundamental type of communication. People use it in their daily lives for various purposes such as convincing others of their views or forming opinions on controversial issues.

The main building block of argumentation is an argument, which typically comprises one conclusion and one or more premises. The conclusion usually states a claim in the discussed issue, and the premise justifies this claim by providing evidence or reasoning.

The process of coming up with new arguments is not straightforward. In many cases, it is a challenge for people to find good arguments that support their views (or counter them) or help to form their opinions. Developing an automatic method for providing people with arguments, hence, is desirable and highly useful.

Over recent years, many studies in the research area of computational argumentation, which deals with the automated processing of argumentation, have been done. However, most of these studies address tasks related to analyzing existing arguments and argumentation in various aspects. The task of generating new arguments has been widely ignored by researchers.

Overall, the existing approaches for argument generation have focused on generating counterarguments by utilizing sources of argumentative texts (Hua and Wang [2018] and Hua et al. [2019]), while the integration of knowledge graphs into the argument generation process remain to be studied. The thesis at hand contributes here: it aims to exploit an argumentation knowledge graph for argument generation. Our motivation for using knowledge graphs is the proven effectiveness of integrating them in text generation Koncel-Kedziorski et al. [2019].

The thesis contributions are as follows: (1) Based on Al-Khatib et al. [2020], which introduced a model for argumentation knowledge graphs, we propose a new (modified) model that is more usable for argument generation.

Based on this modified model, we populate an argumentation knowledge graph, using knowledge identified manually by Al-Khatib et al. [2020], and complete the graph by exploiting its structure and deriving new (implicit) knowledge. (2) We develop an argument generation model, that is capable of generating arguments from the knowledge in an argumentation graph. More specifically, we train the GPT-2 natural language generation model (Radford et al. [2019]) on a combination of three corpora, two comprise argumentative texts and one comprise (mainly) non-argumentative text. The training process includes creating a textual representation of the knowledge encoded in the three used corpora.

We evaluate our argument generation approach with the help of four human annotators. The annotators were provided with 100 pairs, each contains knowledge and an argument generated from the knowledge. The generated arguments were assessed regarding their fluency, informativeness, and relevance to the given knowledge. The evaluation results show that our approach is able to generate high-quality arguments, outperforming previous argument generation approaches (Hua and Wang [2018]).

The remainder of this thesis is structured as follows. Chapter 2 provides a brief overview of the research area of computational argumentation, reports some research studies on graphs for computational argumentation purposes, and describes some existing approaches for argument generation. Then, our construction of the argumentation knowledge graph is introduced in chapter 3, in which we describe the argumentation graph model by Al-Khatib et al. [2020], explain our modifications to that model, and discuss the process of populating and completing such a graph. The process of developing our argument generation model is described in chapter 4. Finally, chapter 5 concludes by recapping the work done in this thesis and presenting improvements to our work, as well as the possible direction to pursue in the future.

Chapter 2

Background and Related Work

Argument generation from an argumentation knowledge graph is one of the computational argumentation tasks. In this chapter, we briefly overview computational argumentation in section 2.1. In section 2.2, we review some approaches that employed graphs for computational argumentation tasks. Finally, in section 2.3, we discusses several research studies in the area of argument generation.

2.1 Computational Argumentation

Recently, the research area of computational argumentation, which is concerned with the automated processing of argumentation, has emerged. Computational argumentation comprises two main sub-areas, (1) the analysis of argumentation and (2) the synthesis of argumentation.

Many research studies considered various aspects of analyzing argumentation. This includes tasks such as identifying arguments in text (Al-Khatib et al. [2016]), segmenting arguments into argumentative discourse units (Ajjour et al. [2017]), assessing the quality of an argument (Wachsmuth et al. [2017a]), mining the argumentation structure (Gemechu and Reed [2019]), and argument retrieval (Wachsmuth et al. [2017b]).

The synthesis sub-area of computational argumentation has received less attention compared to the analysis one. Argument synthesis includes several tasks that aim for generating new arguments. Few approaches were proposed for argument synthesis, which we describe in section 2.3.

2.2 Argumentation Graphs

Graphs have not been utilized for argument generation, but they have been employed for different computational argumentation tasks. For example, Toledo-Ronen et al. [2016] developed an *Expert Stance Graph* (ESG) with the goal of supporting stance classification, argument mining, and several other tasks. Their ESG comprises information about the stance of many experts towards specific controversial issues. The graph is modelled as a directed-bipartite-graph consisting of two types of nodes, the first type corresponds to experts, and the second to controversial issues. The stance of an expert towards an issue is represented by a directed edge from an expert node to a controversial issue node. The graph is populated based on Wikipedia data regarding controversial issues and experts.

Also, Gemechu and Reed [2019] introduced an approach regarding 'Decompositional Argument Mining'. In this approach, they represent the structure of an argument as a graph by encoding information about the relationships between the segments of an argument (e.g., whether one segment supports or attacks another segment). The graph is constructed in four steps: (1) the argument is segmented into target concepts (C) and aspects (A). The opinion on concepts (OC) and opinion on aspects (AC) are identified (i.e., the negative or positive attitudes towards concepts and aspects). (2) The similarities between C and A are computed to connect segments of an argument. (3) To see whether they contradict or agree, relations between OC and AC are identified. (4) The segments of the arguments are linked, based on the previously identified similarities and relations, to construct the final graph.

Eide [2019] compiled Swedish parliamentary data into a semantic graph, storing information about members of the parliament. The metadata from speeches with metadata about the members of the parliament are combined to construct a graph. The graph covers the presence of parliament's members in the parliament, their participation in debates and commissions, party affiliation and other biographical information. The goal of constructing the graph is to employ it in a named entity recognition and resolution system, in order to improve the effectiveness in argument mining in Swedish parliamentary debates.

2.3 Argument Generation

In the following, we cover some existing approaches for argument generation.

Hua and Wang [2018] tackled the problem of automatic counterargument generation. Given an argument on a controversial issue, the goal is to generate an argument with a different stance on that issue. An argument generation

pipeline is developed with two main components: *evidence retrieval* and *argument construction*. The *evidence retrieval* component, first, retrieves a list of relevant articles from Wikipedia, by constructing a set of queries based on an argument's keywords, and then reranks sentences from the retrieved articles by their potential to support the argument that will be generated in the *argument construction* component. The *argument construction* includes two steps: in the first one, a generation model encodes the statement and the evidence with a shared encoder in a sequence. In the second step, two separate decoders are employed, one for generating intermediate representations of key phrases followed by a another one for producing the final argument. The model is trained on argument-counterargument pairs taken from the subreddit /r/changemyview. The results of the experiments show that their model is capable of generating new counterarguments, with a better scores in both automatic and human evaluation compared to standard sequence-to-sequence text generation models.

Hua et al. [2019] extended the previous work of Hua and Wang [2018]. In specific, a new counterargument-generation model is proposed. The new model replaced the *evidence retrieval* component with an *argument retrieval* component, which is responsible for retrieving passages relevant to the input argument from Wikipedia and news media, reranking those passages according to their stance towards the input argument, and extracting key phrases from the ranked passages. A counterargument is then generated in two subsequent steps: first, a text planning decoder produces sentence-level representations, encoding the key phrases and the language style of an argument to be generated. These intermediate representation are then fed to a content realization decoder, which produces the final counterargument. With several experiments, the model achieves better results compared to the previous work.

Chapter 3

Argumentation Knowledge Graph Construction

This chapters gives insights into the process of constructing an argumentation knowledge graph. Section 3.1 introduces a way of modeling argumentative knowledge as a graph. We base our model on a model for argumentative knowledge, proposed by Al-Khatib et al. [2020], to which we add some modifications. Section 3.2 demonstrates the population of a new argumentation knowledge graph based on the proposed model. First, we introduce the corpus that we used as the base of our graph population. Then we describe the process of transforming the knowledge present in this corpus into a knowledge graph. The resulting graph contains knowledge that is not explicitly present in the graph but can be derived from it. To add this implicit knowledge explicitly to the graph, we identify those, so called implicit relations, and add them to the graph in section 3.3. Finally section 3.4 concludes this chapter by providing statistics of the resulting argumentation knowledge graph.

3.1 Argumentative Knowledge

Al-Khatib et al. [2020] introduced a model for argumentative knowledge, that captures certain types of relations between concepts, encapsulated in arguments. The relations encode the effect type that one concept could have on another concept. For example, concept A has a positive or a negative effect on concept B; positive has to be understood as 'A promotes/causes/leads to B' and negative as 'A suppresses/prevents/stops B'.

Original Model

Al-Khatib et al. model argumentative knowledge as an unweighted directed graph with the following components:

- **Concept Instances** correspond to nodes in the graph. A concept instance is a phrase expressing an entity, event or an abstract principle or idea. If available, concept instances are grounded in concepts from a knowledge base (e.g. Wikipedia).
- **Effect Relations** are represented by directed edges in the graph. An effect relation is given if a source concept instance affects a target concept instance, either positively or negatively.
- **Concept Consequences** are modeled as attributes of nodes. A concept instance may be considered, in general as a good or bad consequence.
- **Concept Groundings** are modeled as attributes of nodes. A concept instance is grounded by mapping it to one or more concepts in a knowledge base. By doing so, concept instances that represent the same concept can be identified.

The resulting graph structure can be seen in figure 3.1 taken from Al-Khatib et al. [2020].

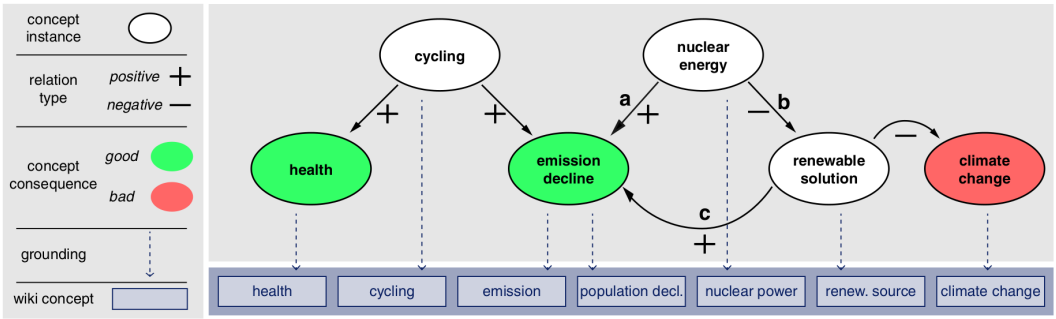

Figure 3.1: Exemplary instance of the proposed argumentation knowledge graph. - Al-Khatib et al. [2020]

Modified Model

Here, we explain our modified version of the graph model of Al-Khatib et al.. The modification aims to enhance the utilization of the graph for argument generation.

Instead of modeling the consequence of a concept instance as an attribute of a node, we choose to model it as an additional type of relation and therefore introduce new edge- and node-types. The motivation for this change is that we later on in the argument generation process base on the consequences of a concept instance to generate arguments with a specific stance (see chapter 4).

These modifications lead to the following components of the knowledge-graph:

- **Concept Instances** same as in Al-Khatib et al. [2020]
- **Effect Relations** same as in Al-Khatib et al. [2020]
- **Concept Groundings** same as in Al-Khatib et al. [2020]
- **Consequences** are modeled as an additional type of node in the graph, representing concept instances for which a target concept instance in an effect relation is either 'good' or 'bad' for. In contrast to the concept instances involved in the effect relation, concept consequences are not grounded into a knowledge base.
- **Consequence Relations** are modeled as edges in the graph. A consequence relation has as the source node a concept instance and as the target node a consequence. Consequence relations have one of two types, either 'good', if a concept instance has 'good' consequences, or 'bad', if the consequences of a concept instance are considered as 'bad'.

Additionally each node and relation in the graph has been indexed by a unique ID, allowing for easier identification of elements in the graph. Also, each edge in the graph has its source text (the claim, which a relation was extracted from) as an attribute.

The graph structure resulting from these modifications can be seen in figure 3.2

3.2 Argumentation Knowledge Graph Population

Based on the new model discussed in the previous section, this section describes the population of the graph with argumentative knowledge. We, first, introduce the corpus that is utilized for populating the graph and then describe the population steps, in which the knowledge present in this corpus was mapped into our proposed model for an argumentative graph.

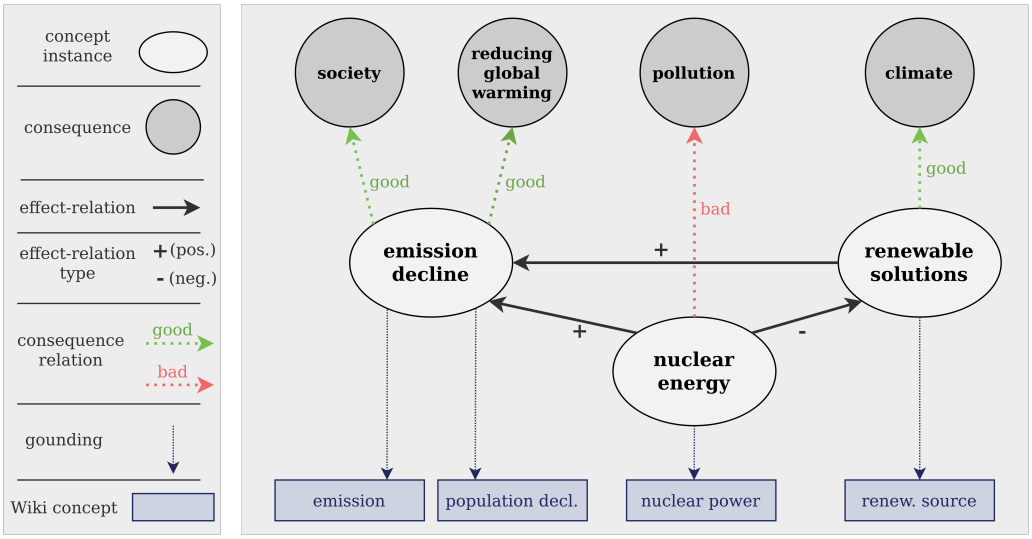

Figure 3.2: Exemplary instance of the modified argumentation knowledge graph.

3.2.1 Argumentation Corpus

Al-Khatib et al. [2020] developed a corpus that comprises in total 16429 manual annotations of 4740 claims, crawled from the online debate portal *debatepedia.org*, each claim has been annotated by five annotators. The annotators were asked to identify knowledge encapsulated in the claims. First, the annotators had to decide whether a claim contains an effect relation, if yes, they had to decide which concepts are involved in the effect relation as well as which type of relation (positive or negative) is there. For each of the two concepts, the annotators were provided by a list of Wikipedia articles, from which they had to pick the articles relevant to the respective concept. Finally, the annotators had to decide if the consequence of the target concept in the effect-relation is 'neutral', 'good', or 'bad'. If they considered the consequences to be 'good' or 'bad', they were asked to list some concepts that the target concept in the effect relation is 'good' or 'bad' for. An example of an annotated claim can be seen in figure 3.3 taken from Al-Khatib et al. [2020].

3.2.2 Aggregation of the Annotations

The five annotations for each claim were aggregated, as follows:

- **Step 1:** A claim is considered to contain an effect relation, if the majority of the five annotators labeled the claim as containing an effect relation. Claims in which no effect-relation was identified are ignored in

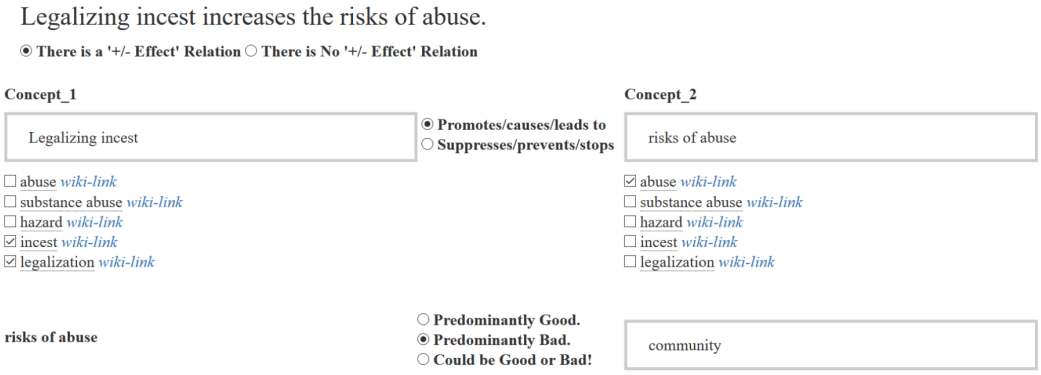

Figure 3.3: The figure shows the annotation interface used by Al-Khatib et al. [2020] for the annotation task. 'Concept_1' and 'Concept_2' correspond to the source and target concept instances of the effect relation respectively. - Al-Khatib et al. [2020]

the following steps, because they contain no knowledge relevant for the graph.

- **Step 2:** The source and target concept instances of the effect relation are aggregated using the longest common mention of a concept instance, appearing in at least the majority of the annotations, that labeled the claim as containing an effect relation.
- **Step 3:** For the groundings of the source and target concept instances of the effect relation we only used the groundings, annotated by at least the majority of the annotators, labeling the claim as containing an effect relation.
- **Step 4:** The type of the effect relation is computed by using the label that the majority of the annotators, that labeled the claim as containing an effect relation, agreed on, or 'NoAgreement', if no agreement was found.
- **Step 5:** For the type of the consequence relation, we used the label that the majority of the annotators, that identified the presence of an effect relation in the claim, agreed on, or 'NoAgreement', if no agreement was found.
- **Step 6:** If in the previous step, the type of the consequence relation was identified as 'good' or 'bad', all the concepts listed by the annotators, that labeled the consequence relation as the respective type, were used as consequences.

An example of the aggregation of the five annotations of a claim can be seen in figures 3.4 and 3.5.

	Relation	Concept 1	Groundings 1	Effect	Concept 2	Groundings 2	Consequence Relation	Consequences
0	Relation	legalizing drugs	['legalization', 'drug']	positive	state can regulate the sale	['state (polity)', 'drug', 'regulation']	good	['citizens', ' public health']
1	NoRelation							
2	Relation	legalizing drugs	['state (polity)', 'legalization']	positive	regulation	['regulation']	neutral	
3	Relation	legalizing drugs	['legalization', 'drug']	positive	state can regulate the sale	['state (polity)', 'regulation']	good	['state', ' government', ' public safety']
4	Relation	legalizing drugs	['legalization', 'drug', 'recreational drug use']	positive	regulate the sale	['state (polity)', 'regulation']	good	['taxes', ' budgets', ' sales']

Figure 3.4: An example of the five annotations of the claim: *By legalizing drugs, the state can regulate the sale.* Concept 1 and Concept 2 correspond to the source and target node of the effect relation respectively. The same goes for Groundings 1 and Groundings 2.

	Relation	Concept 1	Groundings 1	Effect	Concept 2	Groundings 2	Consequence Relation	Consequences
0	Relation	legalizing drugs	['legalization', 'drug']	positive	state can regulate the sale	['state (polity)', 'regulation']	good	['citizens', ' public health', ' state', ' government', ' public safety', ' taxes', ' budgets', ' sales']

Figure 3.5: The resulting aggregation of the annotations from figure 3.4.

3.3 Argumentation Knowledge Graph Completion

At the current state, the graph contains only relations that were explicitly identified by the annotators in the claims. However, the graph contains relations between concepts that are not directly connected to each other in the graph.

For example, let's consider the following example from Al-Khatib et al. [2020]. Suppose we have these three statements:

(a) Nuclear energy leads to emission decline.

(b) Nuclear energy undermines renewable solutions.

(c) Renewable solutions tackle climate change and help to decline emission.

Those statements are mapped to the following relations in the graph:

(a) Nuclear energy has a positive effect on emission decline.

(b) Nuclear energy has a negative effect on renewable solutions.

(c) Renewable solutions have a positive effect on emission decline.

We can observe that, that nuclear emission has a positive effect on emission decline (through (a)), but on the other hand, nuclear energy also has a negative effect on emission decline (through (b) and (c)).

Such relations, that require following a path from a source concept ('nuclear energy') to a target concept ('emission decline') through one or more other concepts ('renewable solutions'), can be directly added to the graph. Hence, no need for recomputing them every time they are required in the argument generating process.

We call those kind of relations, which require to be derived from the graph through processing paths, 'implicit relations', and we call the task of identifying these relations 'graph completion'.

Similar to the explicit relations, we also distinguish two types of implicit relations:

- **Implicit Effect Relations** are relations derived from a path between a source and target concept instance, consisting only of effect-relations.
- **Implicit Consequence Relations** derived from paths between a source and a target concept where the last relation in that path is a consequence relation.

To uncover all the implicit relations present in the graph and add them as edges to the graph, we considered all pairs of concepts where there exists a path from a source concept to a target concept consisting of two or more relations.

Figure 3.6: The same excerpt from the graph, one before the completion process (at the top of the page) and one after it (at the bottom). Red nodes correspond to concept instances and grey nodes to consequences. Green edges model positive/good effect-/consequence-relations. Red edges correspond to negative/bad effect-/consequence-relations.

For the paths with a length of only two relations, the implicit relations were added to the graph simply by considering the types of the two relations and adding a new edge from the starting node of the path to the ending node of the path with the resulting type for the edge as seen in table 3.1.

Table 3.1: Values for the resulting implicit relations for implicit effect- and consequence-relations.

Implicit Effect Relations		
First Relation	Second Relation	Resulting Relation
positive	positive	positive
positive	negative	negative
negative	positive	negative
negative	negative	positive
Implicit Consequence Relations		
First Relation	Second Relation	Resulting Relation
positive	good	good
positive	bad	bad
negative	good	bad
negative	bad	good

For paths spanning three relations or more, we first consider only the first two relations of that path and add the implicit relation as previously described, if it is not already present in the graph. Then, we use this implicit relation with the next relation in the path and also discover the implicit relation for those two relations. This process is repeated until the last node in the original path has been reached.

Each edge that was added to the graph by this procedure has the path it derived from as an additional attribute.

3.4 Graph Statistics

As a result of the initial population of our graph with knowledge from manually annotated claims, in section 3.2, and the completion of the graph with implicit consequence- and effect-relations in section 3.3, we now give information regarding the structure and size of the resulting graph.

Compared to other knowledge graphs, consisting of millions of nodes and edges, our graph is comparably small, consisting of 5016 edges and 17229 edges. Table 3.2 shows the number of appearances of each type of node from the in total 5016 nodes, 2720 of them represent concept instances and 2296 represent consequences.

The completion of the graph roughly doubled the amount of relations present in it. Table 3.3 shows the distribution of occurrences of the different types of relations. The aggregation of annotations, resulted in 8442 consequence- and effect-relations between the nodes. The completion of the graph, by computing implicit relations, resulted in additional 8787 relations.

Figure 3.7 shows the top 30 nodes with the most incoming and outgoing edges, giving an overview over the concepts most commonly covered in the graph.

Table 3.2: Distribution of the two types of nodes in the graph.

	#Nodes
Concept Instances	2720
Consequences	2296
$\sum$	**5016**

Table 3.3: Overview over the distribution of edges in the graph. 'Explicit' denotes the edges stemming from the annotated claims. 'Implicit' denotes the edges added in the completion process.

Edge Type	Edge Value	Explicit	Implicit	$\sum$
Effect	positive	1282	38	1320
Effect	negative	388	25	413
Consequence	good	3214	1962	7176
Consequence	bad	3558	4762	8320
$\sum$		8442	8787	**17229**

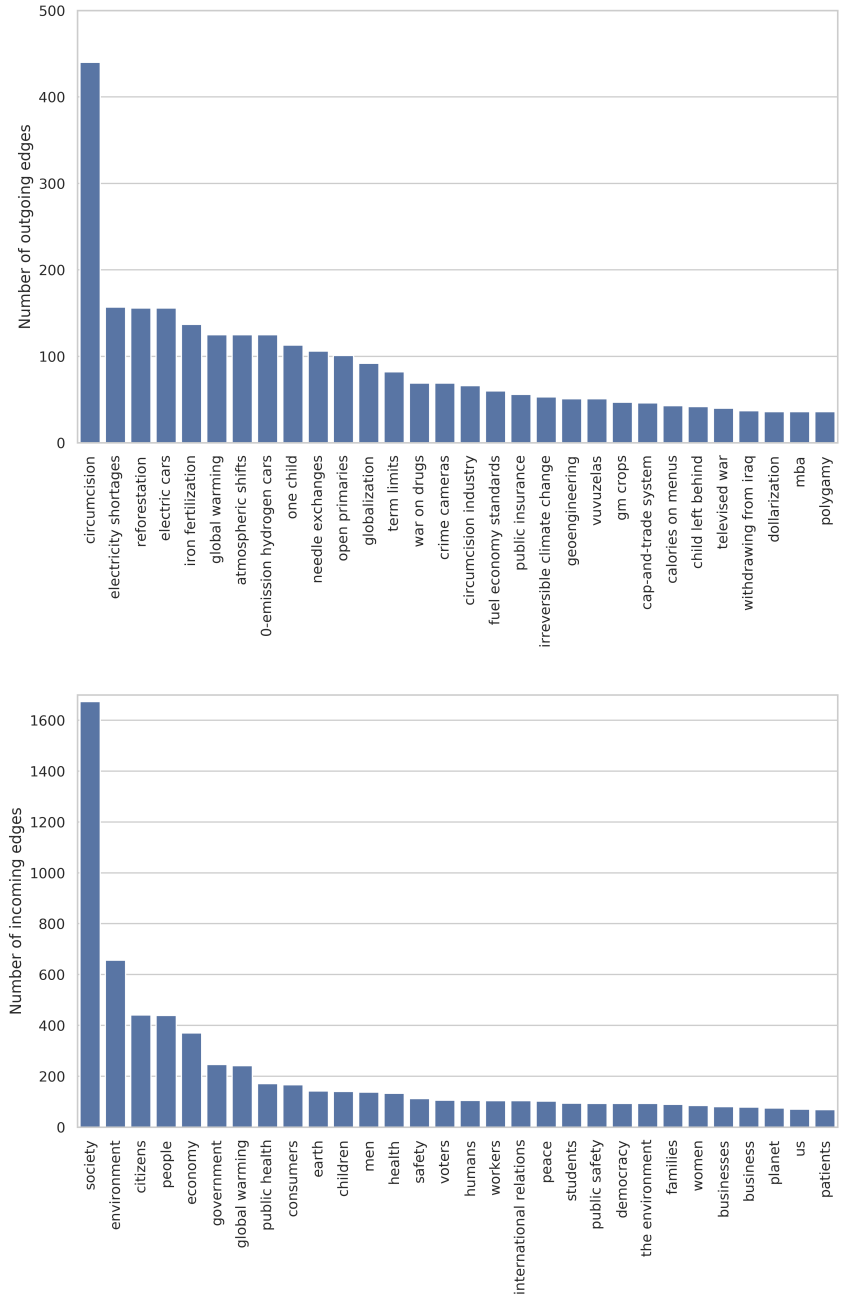

Figure 3.7: Overview of the nodes with the most outgoing (at top) and incoming edges (at bottom).

Chapter 4

Argument Generation

This chapter describes our approach for generating arguments based on the knowledge graph we described in chapter 3. The approach includes taking one or more simple paths from the knowledge graph and transforming them into a natural-language argument. In the following, we first describe the transformation of knowledge to text, and then discuss the connection between our argument generation approach and argument web search engines such as `args.me`.

4.1 Knowledge to Text

This section discusses how to transform the knowledge, we from the graph, into an argument. From the various approaches for text generation, we employ a neural text generation model. The reason for that is the recent advances in the development of neural text generation models and their proved effectiveness.

4.1.1 Approach

Different neural language generation model are capable of generating text from structured data such as knowledge in graphs. For example, Koncel-Kedziorski et al. [2019] trained a model on graph-structured knowledge extracted from scientific abstracts to generate new abstracts. Also, Song et al. [2018a] employed a graph-to-sequence model to generate text from abstract meaning representations (Banarescu et al. [2013]). However, training these approaches requires human written text as well as the knowledge encoded in the text. To apply such approaches for generating arguments, we need an appropriate corpus that consists of argumentative text with graph-structured annotations of the knowledge encoded in it. The lack of appropriately annotated data makes applying those approaches not possible in this thesis. Even though the proposed argumentation knowledge graph comprises graph-structured annotations of claims,

using this corpus as the training set for an argument generation model would lead to short claim-like generated arguments that lack reasoning or evidence. Our goal is to generate arguments, that not only include a claim but also back it up with reasoning or evidence according to the input knowledge retrieved from our graph.

The construction of a corpus that meets the requirements of the current knowledge-to-text generation models is out of this thesis' scope. Therefore, instead of employing a knowledge-to-text generation model, we employ a text-to-text generation model and train it to generate new arguments from a textual representation of the graph-structured knowledge. The training process is described in detail in the following.

4.1.2 GPT-2

Radford et al. [2019] introduced a new large-scale neural language model, called GPT-2. This model is capable of generating coherent text and achieving state-of-the-art results in many language generation tasks such as question answering, machine translation, and text summarization. GPT-2 uses the transformer architecture proposed by Vaswani et al. [2017], and was trained on a dataset of 8 million web pages with the goal of predicting the next word, given all the previous words within some text[1].

Even though GPT-2 is not able to process a graph structure as input data, we decided to leverage its powerful language generation capabilities to generate arguments. We did so by using a textual representation of the knowledge, in the proposed argumentation knowledge graph, as an input for the model.

GPT-2 developed four pre-trained models available to the public. Those models vary in their number of parameters, spanning a range from 124 Million to 1.5 Billion parameters[2]. We chose to perform our experiments using the largest pre-trained model with 1.5 Billion parameters.

4.1.3 Training Corpus

The pretrained GPT-2 models were trained on web-text from various different sources, including argumentative as well as non-argumentative text. To make GPT-2 viable for argument generation, we chose to fine-tune it with texts derived from three corpora: two contain mainly argumentative texts and one contains mostly non-argumentative texts. The later includes texts related to the topics present in the argumentation knowledge graph, hence, it can

[1] https://openai.com/blog/better-language-models/
[2] https://openai.com/blog/gpt-2-1-5b-release/

supplement the text-generation model with additional information on these topics. The three corpora are described in detail in the following.

Args.me Corpus

The *args.me corpus*, developed by Ajjour et al. [2019], is the corpus underlying the argument search engine *args.me*[3], proposed by Wachsmuth et al. [2017b]. It comprises arguments from four different online debating portals, *debate.org*[4], *debatewise.org*[5], *debatepedia.org*[6] and *idebate.org*[7] up to May 2019. Each argument in this corpus consists of a conclusion and a premise. We only consider the premise of an argument in the training (aka fine-tune) process, because the conclusions are usually short spans of texts and in many cases only state an assertion without giving an explanation or a reason for it. We only consider arguments from three of the four debate portals and ignore those in *debate.org*, because they are, compared to arguments from the other debate portals, of a rather low quality. In total, we end up with 28705 arguments from three considered debate portals.

r/Changemyview Corpus

Reddit is a social news aggregation, web content rating, and discussion website [8]. It is structured into different sub-reddits, concerning different topics or issues. Users can submit content to reddit, which can get up- or downvoted by other users. One particular interesting subreddit for our goal is *r/changemyview*[9], which allows users to post their views on a controversial issue (original posts) and allows other users to reply to those original posts to change their views. The user who posts the original post can award the replies that changed his view with a, so called, delta. Content from this subreddit is of a high quality, due to the good moderation in Reddit, and has been used in research in Wei et al. [2016] and Hidey et al. [2017], for example.

Tan et al. [2016] compiled the content of *r/changemyview* into a corpus, including 20.626 posts with their replies, covering a timespan from 2013 to 2015. For our experiments, we only considered content from top-level posts and skipped the replies. The content of a post was split into paragraphs, based on its structure, this resulted in 28705 paragraphs from *r/changemyview*.

[3] www.args.me
[4] https://www.debate.org
[5] https://debatewise.org/
[6] http://www.debatepedia.org
[7] https://www.idebate.org
[8] https://en.wikipedia.org/wiki/Reddit
[9] https://www.reddit.com/r/changemyview/

Wikipedia Corpus

Wikipedia.org[10] is the largest online encyclopedia with over six million articles and therefore is a valuable source for information about numerous topics. The Wikimedia[11] foundation makes Wikipedia articles available in the form of Wikipedia-dumps. In comparison to the two previously mentioned corpora, Wikipedia contains mostly non-argumentative text, consisting of facts about concepts. Due to the massive size of Wikipedia, we only selected a subset of articles for training our model. In particular, we selected the articles that are relevant to the concepts present in our graph. We did so using the groundings of concepts, that directly link a concept to one or more Wikipedia articles. After selecting relevant articles, we also split them into paragraphs based on their structure, which resulted in 81872 paragraphs from 2050 articles.

Statistics of the resulting corpus can be seen in table 4.1.

Table 4.1: Overview over the resulting corpus. '#Paragraphs' denotes the number of paragraphs from each corpus. '#Documents' denotes the number of document the paragraphs stem from. A document is either an argument, a post or an article depending on the corpus.

	#Paragraphs	#Documents
Args.me	33,864	33,864
r/changemyview	28,705	20,626
Wikipedia	81,872	2,050

4.1.4 Textual Representation of Knowledge

With the goal of training a model that is capable of generating arguments from knowledge, we need to provide the model with knowledge during the training process. For that purpose, we employed a heuristic approach of using an entity-linking system to detect concepts present in a span of text, and a list of verbs that potentially indicate effect-relations between concepts. In this way, the model is provided with textual representations of the knowledge in the training data. This representation is similar to the structure of knowledge in our graph.

[10] https://en.wikipedia.org
[11] https://wikimediafoundation.org/

Knowledge Extraction

This step includes the identification of the concepts and the presence of an effect relation in a text.

Concept Identification DBpedia Spotlight[12] (Daiber et al. [2013]) is a entity-linking system, that links entities present in text to Wikipedia concepts. We used this tool for identifying concepts in arguments. We implemented our own instance of it using the code provided by its developers[13].

Effect relation Identification For the detection of potential positive or negative effect-relations, we compiled a list of 3440 verb-indicators from two sources (Choi and Wiebe [2014] and Rashkin et al. [2015]), that may indicate the presence of an effect-relation.

Paragraph Preprocessing We then detect relations in text as follows. We first query DBpedia Spotlight to identify the concepts in a paragraph. Then, we identify all the appearances of a verb-indicator in this paragraph. If a sentence in a paragraph contains the pattern 'concept' 'verb' 'concept', we consider it as a relation. Paragraphs where no relation was identified are removed from the corpus. The resulting corpus can be seen in table 4.2.

Table 4.2: Size of the training corpus after and before the relation extraction process.

	#Paragraphs (with relations)	#Paragraphs (Total)
Args.me	11,744	33,864
r/changemyview	10,254	28,705
Wikipedia	46,958	81,872
$\sum$	**68,956**	144,441

4.1.5 Training

For training our argument generation model, we encoded the from the training corpus extracted relations as text (see figure 4.1). Two special tokens were

[12]https://www.dbpedia-spotlight.org/
[13]https://github.com/dbpedia-spotlight/dbpedia-spotlight-model

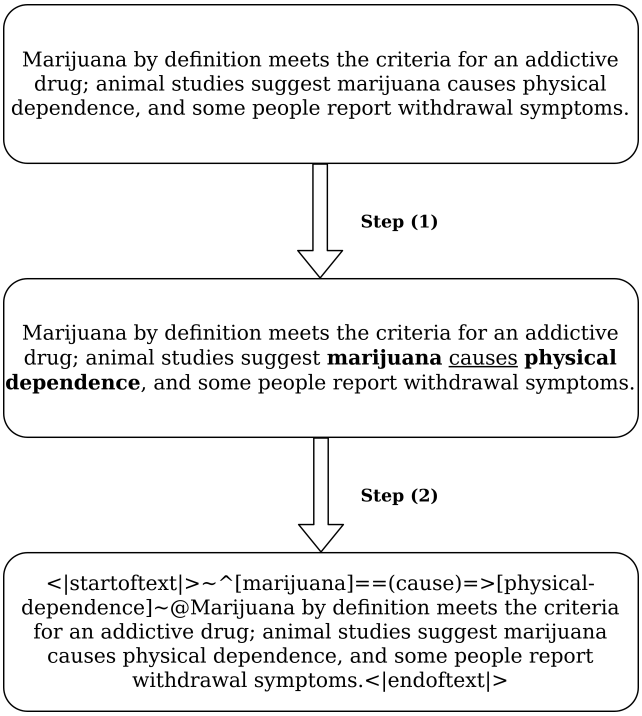

Figure 4.1: Overview over the process of preparing the training corpus. **(1)** For each instance (paragraph) in the training corpus we first heuristically identify knowledge, similar to the knowledge in our graph, by detecting concepts (highlighted as bold text) and relations between the concepts (underlined text), in it. **(2)** The identified knowledge in the training instance is encoded as text and added as a prefix to the training instance, finally the special tokens delimiting the start and end of the training instance are added.

added, '<|startoftext|>' and '<|endoftext|>', to delimit the start and the end of a training instance respectively. An example of one training instance can be seen in figure 4.1.

We trained our model using the GPT-2 implementation in github[14], which allows for easy finetuning of pre-trained GPT-2 models. We finetune the pre-trained 1.5 billion parameter model for 1000 epochs on our training corpus.

4.1.6 Evaluation

Evaluation Setup

To evaluate the quality of our model, we randomly sampled 100 paths from the argumentation knowledge graph and generated arguments for them. The

[14]https://github.com/minimaxir/gpt-2-simple

argument generation process can be seen in figure 4.2. We considered two different cases. In the first case, a path sampled from the graph, including knowledge that we have before the completion step (see Chapter 3), is used as input for the model (explicit knowledge). In the second case, a path that is solely deduced based on the graph completion is used as an input (implicit knowledge).

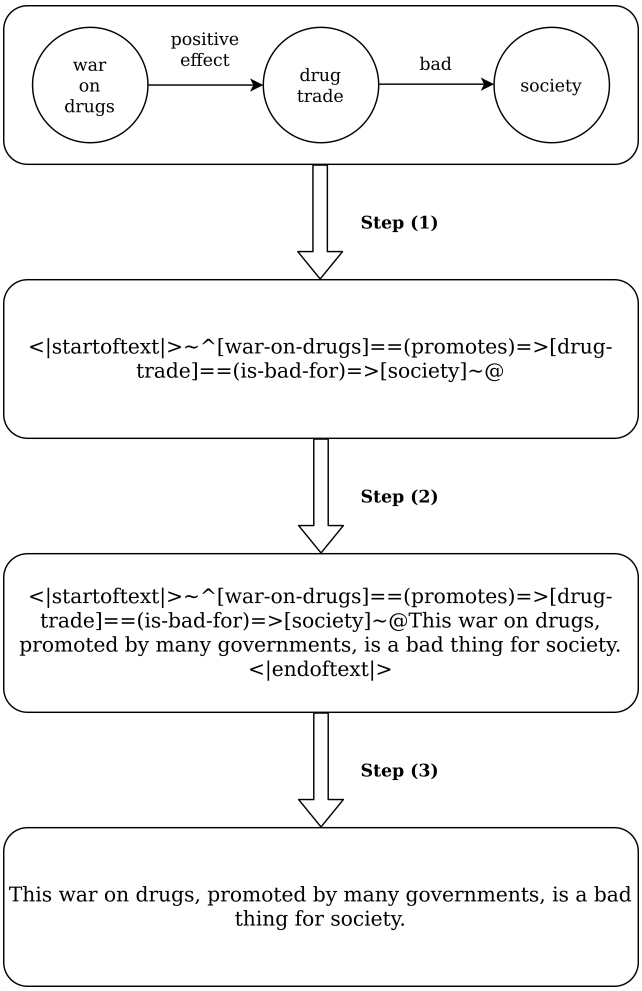

Figure 4.2: Overview over the argument generation process. **(1)** A path from the graph is transformed into its text representation. **(2)** We use the text representation as input for the argument generation model, which then generates an argument. **(3)** The encoded input knowledge and the special tokens are stripped from the model output, leaving only the final generated argument.

We asked four annotators to assess 100 pairs, each includes a knowledge path and an argument, for three different aspects, similar to Hua and Wang [2018]. The three aspects are:

- **Fluency** describes, whether a generated argument consists of grammatically correct English text.
- **Informativeness** describes, whether an argument contains useful information and not generic statements, like 'thank you' or 'it is a good day', nonsense statements, like 'the sun is close to the beginning of the conference', or duplicated statements.
- **Relevance** describes, whether an argument is relevant to the given input knowledge. An argument is more relevant to the knowledge if it covers the concepts and relations, present in the input knowledge.

The annotators were asked to assess each of the aspects on a scale from 1 to 5, corresponding to the worst and best possible score respectively.

Evaluation Results

In the following, we present the results of the annotation task. The agreement of the annotators can be seen in table 4.3. Table 4.4 shows the results of the annotation task and table 4.5 compares it to the results obtained in Hua and Wang [2018].

Table 4.3: The agreement between the four annotators is shown. **Percent (x)** denotes the percentage of samples, where at least **x** annotators rated a sample with the same score in the respective criteria. **Alpha** denotes Krippendorff's alpha agreement, that the annotators achieved in **F**luency, **I**nformativeness and **R**elevance.

	All Samples			**Explicit Knowledge**			**Implicit Knowledge**		
	F	**I**	**R**	**F**	**I**	**R**	**F**	**I**	**R**
Percent (2)	0.93	0.82	0.90	0.92	0.84	0.90	0.94	0.80	0.90
Percent (3)	0.27	0.10	0.27	0.28	0.12	0.26	0.26	0.08	0.28
Percent (4)	0.03	0.00	0.09	0.02	0.00	0.08	0.04	0.00	0.10
Alpha	0.11	-0.09	0.29	0.06	0.11	0.25	0.16	-0.06	0.31

In table 4.3, we see that the overall agreement between the annotators is rather low. We observed that two annotators asses the arguments very differently, one was generally more inclined to give high scores for arguments, and the other one gave low scores.

Table 4.4: The results of the annotation task are shown, for **F**luency, **I**nformativeness and **R**elevance. **Std** denotes the standard deviation, **25%**,**50%** and **75%** denote the respective quantiles. #**Samples** denotes the number of samples in each category.

	All Samples			**Explicit Knowledge**			**Implicit Knowledge**		
	F	**I**	**R**	**F**	**I**	**R**	**F**	**I**	**R**
Mean	**3.71**	**3.22**	**2.52**	**3.67**	**3.24**	**2.59**	**3.75**	**3.195**	**2.45**
Std	0.59	0.55	0.93	0.50	0.50	0.85	0.68	0.68	1.01
25%	3.25	2.75	2.00	3.25	3.00	2.00	3.50	2.75	1.75
50%	3.75	3.25	2.25	3.63	3.25	2.5	3.75	3.13	2.25
75%	4.00	3.50	3.00	4.00	3.50	3.00	4.19	3.50	2.75
#**Samples**	100			50			50		

Table 4.5: The results of our argument generation model are compared to the counterargument generation model from Hua and Wang [2018].

	Fluency	**Informativeness**	**Relevance**
Hua and Wang [2018]	2.5 (±0.59)	1.6 (±0.8)	1.8 (±0.8)
Our model	3.7 (±0.8)	3.2 (±0.6)	2.5 (±0.9)

Tables 4.4 and 4.5 show that the overall quality of our argument generation approach is high; It outperforms the model proposed by Hua and Wang [2018], achieving higher scores in all aspects. We accredit the high scores for 'Fluency' (3.71) and 'Informativeness' (3.67) to the high language-generation effectiveness of the GPT-2 model. For relevance, even though we apply a simple method for generating arguments from a 'text-representation of knowledge', we achieved a score of 2.5, which is considered to be high giving the high difficulty of the task.

To give an impression of the generated arguments figures 4.3 and 4.4 present the five best- and worst-rated generated arguments, respectively. We checked for all the generated arguments, if they appear in the training corpus to make sure that they are not copied from there. None of the generated arguments were present in the training corpus.

financial crisis is (generally bad) for society
In the United States, the financial crisis is a period of economic and social turmoil characterized by the collapse of the housing market and resulting financial crisis is bad for society, particularly in terms of jobs, the national budget, and other social ills. The economic crisis has been the most severe since the Great Depression, and has been referred to by some economists as the Great Depression of the 21st Century.
torture is (generally bad) for society
In the US, a large proportion of the population is against torture. Most countries recognize that torturers are not productive members of society and are therefore not worth supporting. Even if torturers are not a threat to society, they are a threat to the state. By using torture, they are able to get information that could be used to incriminate members of society and to prevent them from carrying out their duties. It is the use of torture that is the most destructive of the individual and society, and the most dangerous of the public. The use of torture is particularly harmful in countries where torturers have a degree of impunity.
investments in green energy (promotes/causes/leads to) reducing us foreign oil dependence, and reducing us foreign oil dependence is (generally good) for citizens
By investing in green energy, governments can promote a green energy policy and reduce oil dependence by cutting fuel prices, lowering environmental standards and promoting energy conservation. In addition, green energy investments can be good for the environment and improve energy security, and health. The United States, for example, is a key player in the global oil industry, having the largest oil production per capita in the world. By investing in green energy, the U.S. can help lower emissions and improve energy security while also reducing the price of fossil fuels, which is an important goal of energy policy.
cloning is (generally bad) for future generations
There are also many more examples of cloning that is bad for the future generations of humankind. In the case of cloning, it is bad for future generations of humankind, because it cannot be done in a legal and ethical framework that is in place. In the case of the use of cloning, it is a form of theft, because the clones are created without any protection for the future generations of humankind. Because the future is not in place, we are not able to protect the future generation of those who are harmed by cloning, and we are not able to protect future generations of those who are harmed by the cloning. Therefore, cloning without protection for future generations of humankind creates a situation that is worse than theft.
war on drugs (promotes/causes/leads to) drug trade, and drug trade is (generally bad) for society
This war on drugs, promoted by many governments, is a bad thing for society.

Figure 4.3: The five best rated generated arguments are shown. Bold text denotes the knowledge-path, that was used for the generation of the respective argument.

civil unions is (generally bad) for humanity
The current human rights situation in the world is bad for humanity. It is not a case of a country being pro-human rights while it is at war with another country. A country which is at war with another country is at war with its own. It is not a case of countries being pro-human rights while at war with each other. A country is at war with its own, as well as the other country.<|state that is at war with itself is at war with itself.<|other country that is at war with itself.<|other country that is at war with itself is at war with itself.<|other country that is at war with itself is at war with itself<|other country that is at war with itself is at war with itself<|other country that is at war with itself is at war with itself<|other country that is at war with itself is at war with itself<
civil unions is (generally bad) for gays and the straight people they marry
As for the civil same-sex marriage, the law is bad for gays and lesbians and for straight people, particularly for those who prefer to marry people of the opposite sex. Many marriage laws in the world, including civil unions, are not only against same-sex marriage, but against same-sex marriage. In countries like Australia, Belgium, Belgium, Denmark, Italy, Luxembourg, Malta, Malta, Luxembourg, Israel, Italy, Israel, Ireland, Italy, Italy, Luxembourg, Malta, Luxembourg, Malta, Malta, Italy, Liechtenstein, Malta, Luxembourg, Malta, Malta, Israel, Liechtenstein, Luxembourg, Malta, Italy, Malta, Liechtenstein, Malta, Liechtenstein, Malta, Liechtenstein, Malta, Liechtenstein, Luxembourg, Liechtenstein, Malta, Liechtenstein, Malta, Liechtenstein, Malta, Liechtenstein, Malta, Liechtenstein, Malta, Liech
eu membership is (generally good) for government
The EU is a single EU member state. All EU countries are members of the European Union. The EU is a single European Union organisation, that is composed of member states that are all members of the EU. The EU is a member of the Organization of Economic Co-operation and Development , the OECD , the Eurostat, the Eurotrade Organization , the Eurofiscal, the Eurofederal and the Eurodif . EU membership is defined as the membership of one country in the European Union.', 'knowledge': 'eu membership is (generally good) for government
cloning (promotes/causes/leads to) creation of children for predetermined, and creation of children for predetermined is (generally bad) for future generations
The fact that human beings have been cloned by human beings is not a reason to pass laws that could lead to the creation of clones. In the case of cloning, because a person has been born and lives in a certain place, a person has a certain level of human potential and hence, human cloning is a human act that is promoted by human beings in a certain place. It is a human act that is bad for humanity as a whole, and therefore not against human beings. The reason why a person has to be treated with care is precisely because a certain level of human potential exists in him, and there is no reason to treat a person with care. Therefore, it is the act of cloning that is harmful for humanity as a whole. It is not because a person has been born and has a certain level of human potential that would lead to the creation of clones, but because his human potential is so great that he would be treated with care. In other words, there can be no
hi-tech crime cameras is (generally bad) for prospective victims of crime
In some cases, a low-tech crime camera is a means of capturing images from crime scenes that can be avoided by use of better technology and proper safety procedures. These cameras may be used to help with the identification of suspects, to help with the investigation of crimes, to help with the capture of people in the field, and to aid in law enforcement investigations and data collection. Low-tech cameras may be used to help with data analysis and to help with forensic investigations. Low-tech crime cameras are often used for a particular purpose, but not all, such cameras are created equally. Some low-tech crime cameras can only capture video from certain angles, such as the angle of the sun or moon. These angle cameras may also have a fixed focus, such as with a tripod or focuser. The angle of the sun or moon may also have a variable focusing system, such as with a variable diaphragm. The fixed focus of a diaphragm may be used to

Figure 4.4: The five worst rated generated arguments are shown. Bold text denotes the knowledge-path, that was used for the generation of the respective argument.

4.2 Connection to Argument Search Engines

Several visionary end-user applications may require the capability of generating arguments that satisfy the information need of a user. A user expects to be presented with generated arguments that take a certain stance towards a certain topic or issue, according to his/her need. In the context of an argument search engine, for example, it is important to tackle the challenge of retrieving knowledge from the argumentation graph that is relevant to the user inputs, and generating arguments that satisfy their needs.

Argument search engines, like `args.me` (Wachsmuth et al. [2017b]), allow users to inform themselves about controversial issues, by presenting them arguments with opposing stances, based on a user-defined query. An example is shown in figure 4.5. The presented arguments are retrieved from the index of the search engine, which consists of arguments extracted out of online debating portals.

An argument generation model can be incorporated into an argument search engine, by dynamically generating new arguments, based on a users query.

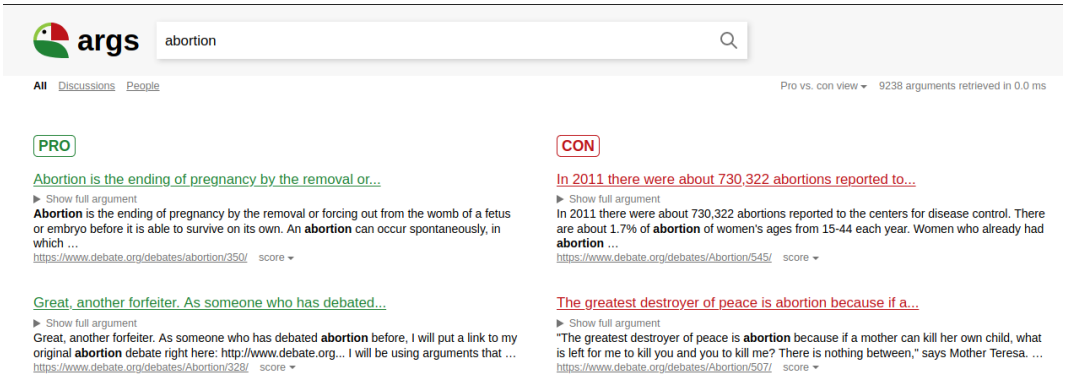

Figure 4.5: An impression of `args.me` with the example query *abortion*.

In the following we introduce three settings to utilize our approach to support argument search engines.

4.2.1 Settings

Here, we will discuss how to consider user needs in an argument search engine when retrieving the knowledge from the graph. We will consider the cases when a user inputs a 'topic', a 'topic' and a 'stance' towards it, and an 'argument' while requesting a counter-argument for it.

Topic

In the setting of having a topic as a query, the goal is to generate an argument that is relevant to that topic. For that it is necessary to retrieve only relevant knowledge-paths from the graph and generate arguments based on them. The requirement for such paths is that they start with a concept relevant to the queried topic. Reliably identifying which concepts are relevant to which topic is a difficult task itself. A starting point for that task is to retrieve only paths, that have the queried topic as their starting node.

For example: Given the query *climate change*, we consider only those paths for argument generation, that have as the starting node the concept *climate change* and generate arguments, based on them, even though other concepts that would also be relevant to that query, like *global warming*, would not be retrieved by that simple approach.

Topic and Stance

Here, we extend the previous setting by introducing an additional constraint: the stance of the argument towards the topic. To generate an argument with an appropriate stance towards the topic (e.g 'pro' or 'con'), we consider only the knowledge that represents that stance.

To do so, after identifying candidate paths, like in the previous settings, that can be used to generate an argument related to a certain topic, we need to select only the paths that are representing the desired stance from the candidate paths. We identify the stance of a candidate path by considering the consequence relations. In specific, we consider a consequence relation with the type 'good' to correspond to the stance 'pro' and 'bad' to correspond to 'con'. The reason is that we assume people to take the stance 'pro' (or 'con') towards some concept, because they perceive the consequences of that concept as good (or 'bad').

This makes determining the stance of paths spanning only one consequence relation trivial. We only need to select those paths from the candidates that have the desired consequence relation.

For paths consisting of more than one relation, we can determine the stance of the path by exploiting the implicit consequence relation resulting from it, which we already added to the graph in chapter 3.

Counterargument

The previous setting can also be extended for generating counterarguments, either by (1) generating a counterargument for an argument with a known

stance towards a known topic or (2) generating an argument-counterargument pair by identifying two paths in the graph with opposite stances.

Chapter 5

Conclusion and Future Work

In section 5.1, we conclude this thesis by describing its main focus. In section 5.2, we discuss the limitation of our work and and the improvements that we plan to pursue in the future.

5.1 Conclusion

In this thesis, we propose the first approach to generate arguments from an argumentation knowledge graph. Within this approach, we populated a new argumentation knowledge graph based on a new model for argumentative knowledge adopted from Al-Khatib et al. [2020] and extended it with knowledge implicitly present in the graph.

Based on the constructed graph, we developed an argument generation model that is capable of transforming knowledge paths from the graph into a natural-language argument. We implement our model based on on the GPT-2 language generation model (Radford et al. [2019]) and trained it on a combination of texts from argumentative and non-argumentative sources, in which relations, similar to the relations present in our graph, were heuristically extracted.

We evaluated our argument generation model with the help of human annotators. The results of the evaluation experiments show that our argument generation model can generate arguments with higher quality than previous argument generation models (Hua and Wang [2018]).

Finally we discussed how end-user applications, like an argument search engine can benefit from a graph-based argument generation approach.

5.2 Future Work

The area of argument generation has been scarcely touched. As far as we know, this thesis is the first that tackles the concrete task of generating arguments from an argumentation knowledge graph, where we developed a new approach that is shown to be effective. Yet, there are many potential improvements to our work, which we discuss in the following.

Even though the relatively small constructed argumentation knowledge graph in this thesis was exploited successfully for generating arguments, we believe that constructing a large-scale argumentation knowledge graph will lead to substantial improvements in argument generation. In general, the graph we used in our research is very small in comparison to traditional knowledge graphs. For example, the Conceptnet (Speer et al. [2016]) knowledge graph comprises millions of nodes and edges, whereas our graph contains only a few thousands of nodes and edges, and restricted to a very limited number of relations.

The manually annotated corpus that is used to construct our graph contains texts from a single source (debatepedia.org) and is limited to the topics mentioned in that source, which in turn, limits our ability to generate arguments for many topics. Also, our knowledge graph is based on rather old texts, that have been updated last time in 2011 [1]. So, recent controversial issues (e.g. Donald Trump, Coronavirus) are not present in that graph, making it not possible to generate arguments on such issues. Incorporating up-to-date sources would solve that problem.

The results of the evaluation of our knowledge-to-text model have shown, that, even though our model achieves overall better results than other argument generation models, there is still room for improvements, especially in how the generated argument represent the input knowledge. We think that incorporating a language-generation model that actually generates text from graph-structured input such as (Koncel-Kedziorski et al. [2019], Song et al. [2018b], or Velickovic et al. [2017]), instead of a text representation of that input (like in our model), will improve the results in that regard. Even though we were aware of such models we decided not to apply such an approach, because of the absence of appropriate training data.

The evaluation of our model showed good results overall, but we had the problem of the low agreement between the annotators regarding the evaluation aspects. In the future, we will enhance the process of evaluating the generated arguments. We also plan to try an automatic evaluation. Even though there are several automatic evaluation metrics, such as BLEU (Papineni et al. [2002]) and METEOR (Banerjee and Lavie [2005]), which have been applied in the

[1] http://www.debatepedia.org/en/index.php/Past_Debate_Digest_topics

past for measuring the quality of text generation approaches. We were not able to apply any of them because of the lack of gold-standard arguments that we can be compared to generated arguments.

This thesis focuses mostly on developing a model for generating natural-language arguments from knowledge, consisting of only one path for one argument. In the future, we hope to be able to generate more complex argumentation from knowledge consisting of multiple paths or even small sub-graphs covering a topic. In section 4.2, we briefly described how knowledge that is relevant for three settings in argument search engines can be identified. This is a very promising direction that we want to follow. In the thesis, we only introduced a simple method for identifying the appropriate knowledge for each setting. In the future, we want to develop more advanced ways of matching topics and stances to knowledge in the graph. Also introducing a ranking-component, that ranks the knowledge, if multiple relevant knowledge-paths are retrieved.

Bibliography

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. Unit Segmentation of Argumentative Texts. In *4th Workshop on Argument Mining (ArgMining 2017) at EMNLP*, pages 118–128. Association for Computational Linguistics, September 2017. doi: 10.18653/v1/W17-5115. URL `http://www.aclweb.org/anthology/W16-2803`. 2.1

Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. Data Acquisition for Argument Search: The args.me corpus. In *42nd German Conference on Artificial Intelligence (KI 2019)*. Springer, September 2019. 4.1.3

Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. Cross-Domain Mining of Argumentative Text through Distant Supervision. In *12th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*, pages 1395–1404. Association for Computational Linguistics, June 2016. doi: 10.18653/v1/N16-1165. URL `https://aclweb.org/anthology`. 2.1

Khalid Al-Khatib, Yufang Hou, Henning Wachsmuth, Charles Jochim, Francesca Bonin, and Benno Stein. End-to-End Argumentation Knowledge Graph Construction. In *34th AAAI Conference on Artificial Intelligence (AAAI 2020)*. AAAI, February 2020. 1, 3, 3.1, 3.1, 3.1, 3.1, 3.1, 3.2.1, 3.3, 3.3, 5.1

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W13-2322`. 4.1.1

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0909`. 5.2

Yoonjung Choi and Janyce Wiebe. +/-EffectWordNet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1181–1191, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1125. URL `https://www.aclweb.org/anthology/D14-1125`. 4.1.4

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013. 4.1.4

Stian Rødven Eide. The Swedish PoliGraph: A semantic graph for argument mining of Swedish parliamentary data. In *Proceedings of the 6th Workshop on Argument Mining*, pages 52–57, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4506. URL `https://www.aclweb.org/anthology/W19-4506`. 2.2

Debela Gemechu and Chris Reed. Decompositional argument mining: A general purpose approach for argument graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 516–526, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1049. URL `https://www.aclweb.org/anthology/P19-1049`. 2.1, 2.2

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5102. URL `https://www.aclweb.org/anthology/W17-5102`. 4.1.3

Xinyu Hua and Lu Wang. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–230, Melbourne, Australia, July 2018. Association for Computational

Linguistics. doi: 10.18653/v1/P18-1021. URL https://www.aclweb.org/anthology/P18-1021. 1, 2.3, 4.1.6, 4.1.6, 4.5, 4.1.6, 5.1

Xinyu Hua, Zhe Hu, and Lu Wang. Argument generation with retrieval, planning, and realization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2661–2672, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1255. URL https://www.aclweb.org/anthology/P19-1255. 1, 2.3

Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text Generation from Knowledge Graphs with Graph Transformers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2284–2293, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1238. URL https://www.aclweb.org/anthology/N19-1238. 1, 4.1.1, 5.2

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040. 5.2

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1, 4.1.2, 5.1

Hannah Rashkin, Sameer Singh, and Yejin Choi. Connotation frames: Typed relations of implied sentiment in predicate-argument structure. *CoRR*, abs/1506.02739, 2015. URL http://arxiv.org/abs/1506.02739. 4.1.4

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for amr-to-text generation. *CoRR*, abs/1805.02473, 2018a. URL http://arxiv.org/abs/1805.02473. 4.1.1

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for amr-to-text generation. *CoRR*, abs/1805.02473, 2018b. URL http://arxiv.org/abs/1805.02473. 5.2

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016. URL http://arxiv.org/abs/1612.03975. 5.2

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of WWW*, 2016. 4.1.3

Orith Toledo-Ronen, Roy Bar-Haim, and Noam Slonim. Expert stance graphs for computational argumentation. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 119–123, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2814. URL `https://www.aclweb.org/anthology/W16-2814`. 2.2

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL `http://arxiv.org/abs/1706.03762`. 4.1.2

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *ArXiv*, abs/1710.10903, 2017. 5.2

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. Computational Argumentation Quality Assessment in Natural Language. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 176–187, April 2017a. URL `http://aclweb.org/anthology/E17-1017`. 2.1

Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. Building an Argument Search Engine for the Web. In *4th Workshop on Argument Mining (ArgMining 2017) at EMNLP*, pages 49–59. Association for Computational Linguistics, September 2017b. 2.1, 4.1.3, 4.2

Zhongyu Wei, Yang Liu, and Yi Li. Is this post persuasive? ranking argumentative comments in online forum. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 195–200, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2032. URL `https://www.aclweb.org/anthology/P16-2032`. 4.1.3