

DeepA2: A Modular Framework for Deep Argument Analysis with Pretrained Neural Text2Text Language Models

Gregor Betz

Karlsruhe Institute of Technology
Karlsruhe, Germany
gregor.betz@kit.edu

Kyle Richardson

Allen Institute for AI
Seattle, WA, USA
kyler@allenai.org

Abstract

In this paper, we present and implement a multi-dimensional, modular framework for performing deep argument analysis (DeepA2) using current pre-trained language models (PTLMs). ArgumentAnalyst – a T5 model (Raffel et al., 2020) set up and trained within DeepA2 – reconstructs argumentative texts, which advance an informal argumentation, as valid arguments: It inserts, e.g., missing premises and conclusions, formalizes inferences, and coherently links the logical reconstruction to the source text. We create a synthetic corpus for deep argument analysis, and evaluate ArgumentAnalyst on this new dataset as well as on existing data, specifically EntailmentBank (Dalvi et al., 2021). Our empirical findings vindicate the overall framework and highlight the advantages of a modular design, in particular its ability to emulate established heuristics (such as hermeneutic cycles), to explore the model’s uncertainty, to cope with the plurality of correct solutions (underdetermination), and to exploit higher-order evidence.

[👤 Demo] [🧠 Model] [📁 Datasets]

1 Introduction

Argumentative text analysis is an interpretation method for clarifying arguments (Fisher, 2004). Being studied in argumentation theory, logic, or epistemology, it is widely taught and applied as a key critical thinking skill in, e.g., law (Alexy, 1989), the humanities (Bruce and Barbone, 2011), social sciences (Fairclough and Fairclough, 2012), policy advice (Hansson and Hirsch-Hadorn, 2016), or public debate (Beck et al., 2019). This paper presents a computational approach for *deep argument analysis*, i.e., for **reconstructing natural-language arguments** from a given text, as in the following example (adapted from Siegel, 2018):

source text \rightsquigarrow **reconstructed argument**

It is unethical to destroy human embryos. The most basic argument supporting this claim just stresses that it is wrong to intentionally kill innocent human beings.	(P1) It is impermissible to kill innocent human beings. (P2) The human embryo is an innocent human being. (C) THUS: It is impermissible to kill the human embryo.
---	---

The literature on argument reconstruction (cf. Feldman, 1998; Scholz, 2000; Lau, 2011; Powell and Kemp, 2014; Brun, 2014; Brun and Betz, 2016) characterizes deep argument analysis as:

- a complex task involving a variety of **sub-tasks**, such as identifying reasons and conclusions in a text, formalizing sentences, checking validity of an inference, logical streamlining, or explicating implicit premises.
- a non-conservative, **creative task** that goes beyond mere text annotation and essentially generates a new, more transparent text.
- an **iterative process** through which reconstructions are built and revised step-by-step, and the solution space is gradually explored.
- a hermeneutical task, guided by the **principle of charity**, which urges one to come up with an interpretation (reconstruction) as strong and plausible as possible.
- assuming a **normative background theory** about what constitutes a strong and plausible argument in the first place.
- being affected by **severe underdetermination**, both in terms of the process and the final outcome; in particular, there typically exist rival, yet equally legitimate reconstructions of one and the same text.

Given these special characteristics, *deep argument analysis* poses many challenges for machine models of natural language understanding. In this paper, we introduce a novel modular modeling approach for analysing complex argumentation that builds on recent pre-trained text2text transformers (Raffel et al., 2020). Our approach – DeepA2 (illustrated in Figure 1) – works by systematically

decomposing a complex reconstruction problem to smaller text2text sub-tasks (see Section 3), which allows for emulating the types of interpretation strategies and heuristics studied in argument theory. Referring to the different components of a comprehensive argumentative analysis, we may also define tailor-made metrics for assessing argument reconstructions. To demonstrate the benefits of our approach, we construct a new argumentation dataset (AAAC) that exhibits several complex *interpretive dimensions*, show how to map other existing datasets into our framework (Section 4), and train and evaluate our main model, referred to as **ArgumentAnalyst**, within DeepA2 (Section 5).

Our empirical results show:

1. ArgumentAnalyst generates – out-of-domain – semantically meaningful argument reconstructions, 70% of which are logically valid. By pooling alternative reconstructions, virtually every source text in the synthetic dataset can be reconstructed as a valid argument.

2. Modular generation chains which emulate iterative reconstruction strategies are highly successful: they yield, in particular, a more coherent interpretation of an argumentative text, exploit the text more thoroughly, and generally outperform one-step generation as soon as problems become difficult.

3. ArgumentAnalyst outperforms *EntailmentWriter* (Dalvi et al., 2021) on difficult *EntailmentBank* problems with respect to telling apart relevant premises from distractors.

4. ArgumentAnalyst generates reliable higher-order evidence (Christensen, 2010) which can be used for diagnosing logical fallacies – despite the fact that ArgumentAnalyst is maximally charitable and is trained to reconstruct any input whatsoever as a logically valid argument, even if the input argument, taken at face value, is painstakingly fallacious.

In concluding this paper, we sum-up and interpret these findings as general vindication of DeepA2’s modular, multi-angular design (Section 6).

2 Related Work

Taking **transformers as soft reasoners**, recent work, pioneered by Clark et al. (2020), has shown that pre-trained language models (PTLMs) possess basic deductive and abductive reasoning capabilities on diverse domains (Banerjee and Baral, 2020;

Betz et al., 2021; Bostrom et al., 2021), but are equally prone to fallacies and biases (Kassner and Schütze, 2020; Talmor et al., 2020). Besides drawing the correct conclusion, transformers are able to generate correct reasoning chains that justify an answer, which in turn further increases answer accuracy (Saha et al., 2020; Tafjord et al., 2020; Gontier et al., 2020; Saha et al., 2021; Dalvi et al., 2021).

Neural semantic parsing uses sequence models to *formalize* natural language sentences (Kamath and Das, 2019). Shin et al. (2021) show that PTLMs are zero-shot parsers, and that intermediate steps which rephrase and streamline the original input before parsing it to a formal language improve accuracy.

Argument mining is an active research field that studies computational methods for retrieving argumentative components from a text corpus (Wachsmuth et al., 2017; Moens, 2018; Potthast et al., 2019; Lawrence and Reed, 2020). Recently, work in this field has started to use PTLMs: Eindor et al. (2020) and Gretz et al. (2020) succeed in retrieving relevant pro- or con-arguments for a given topic from a large corpus with a fine-tuned BERT model (Devlin et al., 2019). Using BERT, Bar-Haim et al. (2020) map argumentative texts to key points that succinctly summarize the argument’s gist. Akiki and Potthast (2020) explore abstractive argument retrieval by means of text generation with GPT2 (Radford et al., 2019). Similarly, Syed et al. (2021) deploy BART (Lewis et al., 2019) to generate conclusions of argumentative texts on a challenging corpus compiled from Reddit and various online debate corpora. Rodrigues et al. (2020), revisiting the argument comprehension task (Haber- nal et al., 2014, 2018), demonstrate that identifying implicit premises – and deep argument analysis *a fortiori* – remains a hard, unsolved task. Recently, Chakrabarty et al. (2021) have shown that augmenting training data with discourse-aware common-sense knowledge improves the plausibility of automatically identified implicit premises. Such a knowledge-driven perspective is orthogonal to, and may eventually complement the logical approach adopted in this paper.

3 Framework

3.1 Problem Definition

Deep argument analysis of a given text seeks to answer the following **central question**: Can we

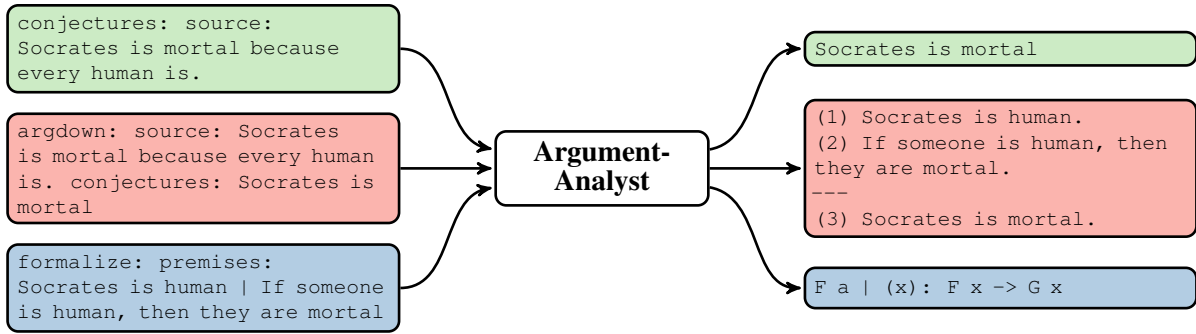


Figure 1: Example text-to-text tasks for deep argument analysis, defined by DeepA2.

make sense of the text as a presentation of a rational argument? And if so, what exactly is the argument; and how precisely is it related to the text?

In carrying out a deep argument analysis, one explicates, rephrases and rebuilds – even repairs – the text’s argument in one’s own words. That is why deep argument analysis is also referred to as *rational reconstruction* (cf. Leitgeb and Carus, 2021). The reconstructed argument forms, together with details about its logical properties and about its relation to the source text, a *comprehensive argumentative analysis* of a text. The latter can be seen as an interpretative hypothesis that is abductively inferred from a source text by means of an inference to the best explanation. Here is another example that illustrates how far a reconstruction may deviate from the original text that presents the argument (adapted from Brun and Betz, 2016):

source text	\rightsquigarrow	reconstructed argument
So, the researcher’s central dilemma exists in an especially acute form in psychology: either the animal is not like us, in which case there is no reason for performing the experiment; or else the animal is like us, in which case we ought not to perform on the animal an experiment that would be considered outrageous if performed on one of us.		(P1) If the animal is not like us, it is wrong to perform the experiment. (P2) If the animal is like us, it is wrong to perform the experiment. (C) THUS (with <i>classical dilemma</i>): It is wrong to perform the experiment.

A compelling argumentative analysis yields (i) a rational argument that is (ii) closely related to the source text. Deep argument analysis is, accordingly, guided by a **dual goal** (cf. Brun and Betz, 2016). An argument reconstruction should both be

- (i) **systematically correct**, i.e., the reconstructed argument itself is, e.g., transparent, deductively valid, non-circular, or doesn’t contain irrelevant premises; and
- (ii) **exegetically adequate**, i.e., the reconstructed

argument accounts for the original text, because, e.g., its premises merely reformulate parts of the text, or because its overall inferential structure can be traced within the source text.

The fact that there typically exists – regarding a specific text – a trade-off between these two goals is one major reason for the underdetermination of deep argument analysis and the plurality of legitimate reconstructions of a given text (cf. Brun and Betz, 2016).

Against this background, we may finally define the problem of

Deep artificial argument analysis: Describe, analyse and implement an effective computational system for deep argument analysis!

3.2 Multi-angular Data

The DeepA2 framework is built upon a *multi-angular* data structure (Tafjord and Clark, 2021) whose dimensions represent the essential components of a comprehensive argumentative analysis (see Section 3.1). Structured argumentative data is rendered as plain text (cf. Voigt, 2014). The different data dimensions, which are related as shown in Figure 2, are (with an illustrating example):

argument source text (S)

It is unethical to destroy human embryos. The basic argument supporting this claim just stresses that it is wrong to intentionally kill innocent human beings.

verbatim reason statements in source text (R)

it is wrong to intentionally kill innocent human beings (ref: (1))

verbatim conjectures in the source text (J)

It is unethical to destroy human embryos (ref: (3))

argument reconstruction (A)

- (1) It is impermissible to kill innocent human beings.
- (2) The human embryo is an innocent human being. – with hypothetical syllogism from (1) (2) –
- (3) It is impermissible to kill the human embryo.

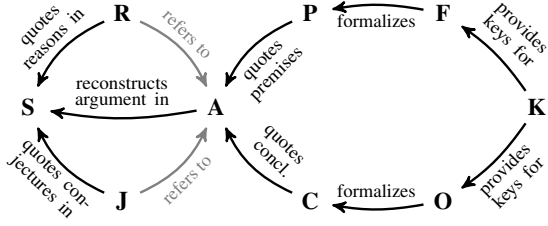


Figure 2: Relationships between dimensions of the multi-angular argumentative data.

premises of the reconstructed argument (P)
 It is impermissible to kill innocent human beings | The human embryo is an innocent human being
final conclusion of reconstr. argument (C)
 It is impermissible to kill the human embryo
formalizations of premises (F)
 $(x): Fx \rightarrow Gx \mid (x): Hx \rightarrow Fx$
formalization of conclusion (O)
 $(x): Hx \rightarrow Gx$
keys for the formalizations' constants (K)
 F: innocent human being | G: must not be killed | H: human embryo

Each record in a DeepA2 dataset contains a source text plus a legitimate comprehensive argumentative analysis, which is, given underdetermination, not necessarily the only compelling reconstruction of the text; moreover, a dataset *may* contain different records with one and the same source text analysed in several ways. So, for example, an alternative, equally legitimate argument reconstruction of the above source text (S) may read:

argument reconstruction (A)
 (1) If it is wrong to kill innocent human beings, then it is wrong to kill a human embryo.
 (2) It is wrong to kill innocent human beings.
 – with modus ponens from (1) (2) –
 (3) It is wrong to kill a human embryo.

Beyond this structural and functional characterization, DeepA2 is agnostic about the nature and origin of the argumentative data. Synthetically generated, automatically retrieved, manually created datasets as well as translations of other databases are all compatible with the framework and can be used side by side.

3.3 Generative Modes and Chains

Given DeepA2's multi-dimensional data structure described in the previous section, a **generative mode** maps data from some input dimensions to a target dimension. For example, the mode $S \rightsquigarrow A$ takes a source text (S) as input and outputs an argument reconstruction (A), the mode $RJ \rightsquigarrow A$ reconstructs the argument (A) given the verbatim reasons (R) and conjectures (J). All in all, we define and

investigate 21 different generative modes (see Appendix B). Every mode represents a task on which a text-to-text model can be trained.

By taking some mode's output as another mode's input, modes can be concatenated into **generative chains**. For example, the output of modes $S \rightsquigarrow R$ and $S \rightsquigarrow J$ (reasons and conjectures from source) can be fed into mode $RJ \rightsquigarrow A$ to reconstruct an argument. Such generative chains allow us to emulate different strategies (heuristics) for analysing a given argumentative text (see Appendix C for technical details).

Three generative chains which model distinct interpretative strategies, taking a source text (S) as sole input, are:

straight

$S \rightsquigarrow A$ $S \rightsquigarrow R$ $S \rightsquigarrow J$

hermeneutic cycle

$S \rightsquigarrow A$ $SA \rightsquigarrow R$ $SA \rightsquigarrow J$ $RJ \rightsquigarrow A$

logical streamlining

$S \rightsquigarrow A$ $A \rightsquigarrow P$ $A \rightsquigarrow C$ $C \rightsquigarrow O$ $CO \rightsquigarrow K$
 $OK \rightsquigarrow C$ $PC \rightsquigarrow A$ $SA \rightsquigarrow R$ $SA \rightsquigarrow J$

While the chain *straight*, where no output ever serves as input to another mode, represents a simple baseline, *hermeneutic cycle* and *logical streamlining* mimic prominent, equally-named methods in argument analysis (cf. [Bowell and Kemp, 2014](#); [Brun and Betz, 2016](#)). One goes through a hermeneutic cycle, generally speaking, if one revisits a text in view of its previous interpretation, as, for example, in steps $SA \rightsquigarrow R$ $SA \rightsquigarrow J$, where the source text (S) is re-interpreted (identifying reason statements and conjectures) given the previously reconstructed argument (A), so as to subsequently re-reconstruct the argument itself (step $RJ \rightsquigarrow A$). To logically streamline a reconstruction means to rephrase its conclusion or premises in order to make their logico-semantic structure more transparent. Such semantic clarification can be emulated by (i) formalizing a statement (e.g., $A \rightsquigarrow C$ $C \rightsquigarrow O$ $CO \rightsquigarrow K$) and (ii) using the keys (K) to retrieve the original statement from the generated logical formulas (such as in $OK \rightsquigarrow C$), from which the argument can be re-built (step $PC \rightsquigarrow A$).

For evaluation, we append to each generative chain the following sub-chain that formalizes the reconstructed argument:

formalization

$A \rightsquigarrow P$ $A \rightsquigarrow C$ $P \rightsquigarrow F$ $CPF \rightsquigarrow O$ $PFCO \rightsquigarrow K$

A generative chain can be construed as hypergraph on the dimensions of DeepA2’s multi-angular datasets, with each of its modes representing a directed hyper-edge. Summing up the number of input dimensions (except S) over all modes yields a simple graph centrality measure, which gauges a chain’s sophistication. Thus, *straight*, *hermeneutic cycle* and *logical streamlining* display a sophistication of 0, 4, and 11, respectively.

3.4 Metrics

As discussed in Section 3.1, an argument reconstruction should both be sound and make sense of the text to-be-interpreted. In line with the dual goal of argument analysis, we propose metrics both for the systematic correctness and for the exegetic adequacy of a given analysis. The following metrics measure the degree to which a given generated argument is *systematically correct*:

SYS-PP 1 if the argument is not a *petitio principii* (i.e., if no premise is identical with its final conclusion), 0 otherwise;

SYS-RP 1 if the argument has no *redundant premises* (i.e., if no premise occurs more than once), 0 otherwise;

SYS-RC 1 if the argument has no *redundant conclusions* (i.e., if no conclusion – intermediary or final – occurs more than once), 0 otherwise;

SYS-US 1 if all statements in the argument other than the final conclusion are explicitly *used in an inference*, 0 otherwise;

SYS-SCH ratio of sub-arguments which correctly instantiate the explicitly stated *inference scheme* (e.g., hypothetical syllogism);

SYS-VAL 1 if the argument is *globally valid* (i.e., if the final conclusion deductively follows from the premises), 0 otherwise;

All six systematic metrics can be computed automatically (SYS-SCH tries to parse the argument based on the inference schemes and templates used to construct the synthetic dataset in the first place; SYS-VAL passes the model-generated formalizations of premises and conclusion to a symbolic theorem prover (De Moura and Bjørner, 2008); and the remaining metrics check for string identity).

Whereas systematic metrics apply primarily to the generated argument (A), a reconstruction’s interpretative adequacy will also depend on how reasons (R) and conjectures (J) coherently link the argument’s components to the original text. As a first set of *exegetic metrics*, we thus propose

EXE-MEQ 1 if the reasons and conjectures are *mutually exclusive verbatim quotes* from the source text, 0 otherwise;

EXE-RSS semantic similarity (BLEURT, see Selam et al., 2020) of each reason statement and its counterpart premise in the reconstructed argument (if such exists, -1 otherwise);

EXE-JSS semantic similarity (see EXE-RSS) of each conjecture statement and its counterpart in the reconstructed argument (if such exists, -1 otherwise).

Each source text presents (more or less faithfully) an underlying target argument, which in turn marks some of the text’s statements as ‘target’ reasons, others as ‘target’ conjectures. The following two metrics assess the degree to which a comprehensive argumentative analysis correctly predicts (R, J) those target reasons and conjectures.

EXE-PPR predictive performance (F1-score) for identifying (target) reason statements in the source text;

EXE-PPJ predictive performance (F1-score) for identifying (target) conjecture statements in the source text.

An argument’s final conclusion may be implicit or explicit in a given text. The ability to fully exploit a text can be measured by verifying whether the reconstructed argument’s final conclusion is implicit (= prediction) if and only if the target argument’s one is.

EXE-TE text exploitation, as measured by ability (F1-score) to reconstruct arguments with explicit final conclusions (prediction) if and only if the target final conclusions are explicit.

3.5 Models

Any text-to-text language model is compatible with the proposed DeepA2 framework. We refer to models used within the framework as **ArgumentAnalyst**. In this study, we train and evaluate the transformer model T5 (Raffel et al., 2020) with 770M parameters as implemented by (Wolf et al., 2020).

3.6 Limitations

In the DeepA2 framework, arguments are reconstructed from relatively short and isolated texts, disregarding both the broader context of the argument and domain-specific background knowledge. This limits the framework, as presented here, in

important ways: Implicit premises that are explicated in an argument reconstruction can neither be checked for plausibility nor for agreement with the author’s broader convictions. In addition, the framework cannot assess an argument’s dialectic function in a wider debate. It seems worthwhile to explore according extensions of the framework in future research.

4 Datasets

For the experiments reported below, we synthetically create two artificial argument analysis corpora that comply with the DeepA2 framework (see also Appendix A): AAAC01 and AAAC02. In addition, we translate the synthetic *RuleTaker* (Clark et al., 2020) and the manually compiled *EntailmentBank* (Dalvi et al., 2021) datasets into our framework.

In argument analysis, one proceeds *from* a source text *to* its reconstruction. Creating the synthetic corpora, we reverse-engineer this process:

Step 1. We sample, first of all, a possibly complex argument (A) from a set of valid inference schemes. In doing so, we use a multi-step templating strategy (inspired by Betz et al., 2021) to translate symbolic forms into natural language schemes (which were generated by local domain experts) and to substitute natural language terms for placeholders. Premises (P), conclusion (C) and their formalization (F, O, K) are side-products of such a construction of an argument.

Step 2. Given the fully explicit argument (A), we compose a text (S) that presents the argument in a more or less transparent and faithful way. Such text creation involves: rendering the argument tree as a linear story, leaving out premises or conclusions (implicit premises and conclusions), inserting irrelevant material (distractors), using templates that obfuscate the logical form of a sentence, limiting the use of premise and conclusion indicators (such as “therefore”), applying rule-based and automatic paraphrasing. In composing the argumentative text (S), we may record its reasons (R) and conjectures (J).

Given the synthetic and controlled nature of our dataset, which involved eliciting rule templates from a group of local domain experts, all data is assumed to be correct by *construction*. As an additional check of correctness on the logic of our examples, we ran a symbolic theorem prover (De Moura and Bjørner, 2008) over the argument formalizations to verify their validity. To ensure the fluency

of the underlying language templates, all templates were hand verified by the authors.

Our two datasets AAAC01 and AAAC02 differ in the following ways:

1. predicates and names are sampled from different, disjunct domains (texts are about, e.g., allergies and family relations versus, e.g., badminton and cooking) to test a model’s robustness to lexical diversity (Rozen et al., 2019);
2. similarly, AAAC01 applies automatic paraphrasing (Alisetti, 2021) to the final source text whereas AAAC02 doesn’t;
3. AAAC02 allows for imprecise renditions of logical formulas, while AAAC01 sticks to plain formulations to test robustness to variations in description of rules.

Each dataset contains diverse texts and arguments. Broadly speaking, data records may differ in terms of properties of the argument (step 1 above) and properties of the argument’s presentation (step 2). Along these two dimensions, we define five homogeneous subsets of the data:

simple inference: arguments with a single inference step that neither involves negation nor compositional predicates;

complex inference: arguments with four inference steps that heavily rely on syntactically intricate schemes (e.g., transposition, or de Morgan);

plain presentation: all premises and conclusions are explicit in the source text which, in addition, contains no distractors;

mutilated presentation: at least two premises and one conclusion are implicit, while the text contains two distractors and explicitly states the final conclusion;

C&M: the argument’s inference is complex, plus the text contains at least two distractors.

The *RuleTaker* and *EntailmentBank* datasets contain multi-hop inference trees (A). To import these into the DeepA2 framework, we create source texts (S) for the given arguments by means of simple templates (such as “{theory} All this entails: {hypothesis}”) and record reasons (R) and conjectures (J) on the fly. Unlike AAAC and *EntailmentBank*, *RuleTaker* (as updated in Tafjord et al., 2020) contains an equal share of arguments for which (i) the conclusion follows from the premises, (ii) the conclusion contradicts the premises, (iii) the conclusion is independent of the premises.

5 Experiments and Results

As first and main experiment we train our base model (see Section 3.5) on the AAAC01 corpus, and evaluate the resulting ArgumentAnalyst model out-of-domain on AAAC02. ArgumentAnalyst undergoes multi-task training on 21 generative modes, which are interpreted as sequence-to-sequence tasks (the training set-up is further described in Appendix B).

The evaluation of ArgumentAnalyst on AAAC02 proceeds in two steps: (1.) prediction: produces output in accordance with 16 different generative chains (Appendix C); (2.) metrics application: assesses the quality of the generated output by means of the systematic and exegetic metrics of the DeepA2 framework (see Section 3.4).

Table 1 reports the ability of ArgumentAnalyst to generate systematically correct and exegetically adequate argument reconstructions. We obtain similar global results with the three chains *straight*, *hermeneutic cycle*, and *logical streamlining*, whose generated reconstructions mainly differ in terms of internal coherence (EXE-RSS, EXE-JSS) and text exploitation (EXE-TE). However, the different generative chains complement each other, as shown by *pooling*, which does not only outperform individual chains, but nearly attains oracle performance.

Moreover, ArgumentAnalyst produces much better reconstructions of simple inferences and plain presentations – compared to complex inferences and mutilated presentations, i.e., difficult problems (cf. Table 5 in App. D). In addition, within one and the same subset, substantial differences show up between the three generative chains. Globally speaking, *hermeneutic cycle* outperforms the other two chains for difficult problems.

Is ArgumentAnalyst capable of reliable self-evaluation? We have **validated the logic metric** (SYS-VAL), which passes on a self-generated formalization of the reconstructed argument to a theorem prover, in three ways: First of all, ArgumentAnalyst correctly recognizes *target* arguments as valid (with accuracy 92.7%), which has been verified by running the formalization subchain on target data. Secondly, virtually every generated argument with all-correct scheme instantiations (i.e., SYS-SCH = 1) is also – and correctly – recognized as logically valid. Thirdly, a manual analysis (**human-in-the-loop**) of 100 generated arguments with incorrect scheme instantiation (i.e., SYS-SCH < 1) reveals a high rate of false negatives: roughly one

half of all inferences that are not automatically identified as an instantiation of the given scheme actually do correctly instantiate it. The accordingly *adjusted* global ratio of correct scheme instantiations (Table 1) equals roughly 0.65 (rather than 0.31–0.33), which is consistent with the ratio of logically valid arguments being 0.72–0.73.

Do reconstructed arguments exhibit basic semantic flaws? Regarding the full dataset, ArgumentAnalyst produces nearly **flawless argument reconstructions**, committing basic errors (petitio, redundancy, unused statements) only very rarely (Table 1). And even for very difficult problems, two thirds of all generated arguments display no basic flaw whatsoever (Table 5, SYS-PP & SYS-RP & SYS-RC & SYS-US).

Are reconstructed arguments logically valid? Roughly 70% of all arguments generated by one of the three chains are logically valid (Table 1). More importantly, though, for virtually every source text in the dataset, there is at least one chain (out of 16) which reconstructs the text as a valid argument (*pooling*). Given that logical validity can be automatically assessed, the *pooled* system may thus **guarantee to yield a valid reconstruction**. Concerning different problem types (Table 5), *hermeneutic cycle* clearly outperforms the other chains as soon as the problem gets difficult. Additional analysis shows that ArgumentAnalyst can also **cope with underdetermination**, as 68% of all generated arguments whose final conclusion differs (BLEU \leq .8) from the target argument’s one – i.e., arguments that are not reconstructed as expected given the target data – are still logically valid.

Are the generated interpretations internally coherent? The generative chain *hermeneutic cycle* yields comprehensive argument reconstructions where premises (**P**) and conclusions (**C**) fit much better to detected reasons (**R**) and conjectures (**J**) than *straight* or *logical streamlining* (EXE-RSS, EXE-JSS). This holds globally (Table 1), as well as for easy, and for difficult problems (Table 5). Note that the *oracle* baseline for metrics EXE-RSS, EXE-JSS is well below 1, which reflects the fact that source texts may present arguments in highly mutilated ways; it is nearly attained by *pooling* the 16 different generative chains (Table 1).

Can ArgumentAnalyst detect reasons and conjectures, and fully exploit the text? The evaluation demonstrates that reason/conjecture detection on AAAC02 is a relatively easy task (EXE-PPR, EXE-PPJ).

chain	systematic metrics (SYS-*)						exegetic metrics (EXE-*)					
	PP	RP	RC	US	SCH	VAL	MEQ	RSS	JSS	PPR	PPJ	TE
straight	.95	.97	.96	.96	.33	.73	.80	-.08	-.10	.93	.93	.63
herm. cy.	.95	.98	.95	.93	.31	.72	.82	.16	.12	.93	.92	.71
logic. str.	.95	.97	.96	.95	.32	.72	.82	.11	.00	.93	.92	.69
pooling	1.0	1.0	1.0	1.0	.73	1.0	1.0	.26	.29	.96	.96	.97
oracle	1.0	1.0	1.0	1.0	1.0	1.0	1.0	.30	.37	1.0	1.0	1.0

Table 1: Performance of ArgumentAnalyst on the AAAC02 data as measured by systematic and exegetic metrics. Rows display results for three illustrative generative chains (*straight*, *hermeneutic cycle*, *logical streamlining*), for the item-wise best performing generative chain out of all 16 chains (*pooling*), and for oracle performance (*oracle*), which one obtains by applying the metrics to the target data itself.

steps	ArgAn _{EB}		ArgAn _{AAAC,EB}		EntWr
	straight	herm. cycle	straight	herm. cycle	
1	.863	.866	.816	.871	.951
2	.798	.815	.813	.826	.886
3	.812	.815	.826	.806	.858
4	.757	.791	.820	.822	.838
≥ 5	.795	.811	.786	.773	.742
any	.819	.830	.816	.834	.879

Table 2: Predictive performance of ArgumentAnalyst (ArgAn_{EB}, ArgAn_{AAAC,EB}) and EntailmentWriter (EntWr) for identifying reason statements in an input text (metric SYS-PPR) on the *EntailmentBank task2* dataset.

In contrast, fully exploiting a text (i.e., generating an argument with implicit final conclusion if and only if the underlying target argument has an implicit final conclusion, EXE-TE) is seemingly more challenging (Table 1). Again, *hermeneutic cycle* achieves best text exploitation, performing, however, clearly below *oracle* baseline – which may simply reflect the degree of underdetermination in the AAAC02 corpus.

In a second experiment we train two models on the imported *EntailmentBank* (*task1* and *task2*) dataset (see Section 4), namely: (1.) our base model (T5), which yields ArgumentAnalyst_{EB}; (2.) the ArgumentAnalyst model pretrained on AAAC02 (resulting in an intermediary pre-training set-up similar to Phang et al., 2018; Geva et al., 2020), which yields ArgumentAnalyst_{AAAC,EB}.

Since the *EntailmentBank* data doesn’t contain formalizations, we can only train on 14 modes, which are interpreted as sequence-to-sequence tasks (see Appendix B). We evaluate the models on *task2* of *EntailmentBank* only, which contains problems with a relatively large number of distractors, and proceed in two steps as before: prediction (with 11 different generative chains) and metrics

application. Dalvi et al. (2021) report the ability of *EntailmentWriter* (a fine-tuned T5-11b model) to correctly distinguish relevant premises of an argument from distractors in terms of a F1-score, which corresponds to our metric EXE-PPR. That’s why the sole focus in this second experiment is on EXE-PPR.

Table 2 describes the ability of ArgumentAnalyst models to correctly tell apart relevant premises from mere distractors in the *EntailmentBank task2* dataset for two generative chains (*straight*, which directly outputs reason statements, and *hermeneutic cycle*, which tries to reconstruct the argument first and uses both source text and argument to identify reasons), and compares this with the performance of *EntailmentWriter* (scores from Dalvi et al., 2021). The results, shown separately for arguments with a specific number of inference steps, let us draw three conclusions:

First, *ArgumentAnalyst* outperforms *EntailmentWriter* on difficult problems with more than 4 inference steps / sub-arguments.

Second, using the sophisticated chain *hermeneutic cycle* improves predictive performance compared to the simple *straight* chain.

Third, the chain *hermeneutic cycle* (unlike *straight*) generally benefits from intermediary pre-training on AAAC – caveat: not so for arguments with more than 4 steps. This latter observation might be due to the fact that the AAAC02 corpus, by construction, doesn’t contain arguments with more than 4 steps, so that pre-training biases the model towards shorter arguments.

In a third experiment we explore the following hypothesis:

Informative higher-order evidence. The degree to which ArgumentAnalyst struggles in reconstructing a given argument (presented in the source text) as logically valid is a reliable in-

indicator for whether the original argument is fallacious or not.

To test this hypothesis, we apply ArgumentAnalyst (trained on AAAC02, see above) to the *RuleTaker* data as imported into the DeepA2 framework (see Section 4): ArgumentAnalyst produces – by means of 13 generative chains – comprehensive reconstructions, to which the systematic and exegetic metrics are applied. *RuleTaker* contains an equal share of arguments whose conclusions follow from (label=valid), contradict (label=contradiction), or are independent of (label=neutral) the corresponding premises. Now, informative higher-order evidence would allow us to correctly predict these labels. And this is exactly what we observe: First, if reconstructions of one and the same source text which are independently generated with different chains agree (disagree), then the original argument tends to be valid (invalid). Second, by training simple classifiers on our argumentative metrics and further properties of the reconstructions, we robustly achieve a predictive accuracy 10% above the random baseline. While this is far below the SOTA results of tailor-made RuleTaker (Clark et al., 2020) and ProofWriter (Tafjord et al., 2020) models on this data, our findings nonetheless confirm the above hypothesis.

6 Conclusion

In this paper, we have presented and implemented a multi-angular, modular framework for deep argument analysis (DeepA2). It allows for defining a large variety of generative modes by combining different dimensions of the data. These modes, in turn, can be concatenated into complex generative chains. ArgumentAnalyst – a text-to-text model set up and trained within the DeepA2 framework – yields plausible reconstructions of argumentative texts. Our empirical findings vindicate the overall framework and highlight the following **advantages of a multi-angular, modular design** in general: First of all, modular chains may emulate established, well-proven, typically piece-meal, scholarly techniques for text analysis (heuristics), which hence may provide **normative, methodological guidance** in setting up NLP systems. Secondly, by defining and implementing different modular chains, and investigating the plurality of generated solutions, one can systematically **explore the system’s uncertainty as well as the tasks’s underdetermination**. Thirdly, monitoring the sys-

tem during modular computation yields diagnostically useful information (e.g., intermediary results) which not only describes the model’s performance on the given problem, but which additionally allows us – as **higher-order evidence** – to characterize (e.g., classify) the original problem in the first place. Fourthly, breaking down a complex task into sub-tasks with intermediary results that can be further processed and re-combined helps to **overcome input size limitations** of neural language models. Fifthly, modular generation with meaningful modes allows users to follow the system, comprehend generated solutions, verify sub-steps and detect errors – the NLP system becomes a **transparent, explainable AI** (Miller, 2019). Finally, modular NLP systems as described by DeepA2 may be connected to a user-interface which promises **fine-grained interactive control** of modular generations and seamless cognitive cooperation of AI and human experts in analysing texts.

Acknowledgments

We’re indebted to Christian Voigt for his critical and constructive feedback throughout the DeepA2 project.

References

- Christopher Akiki and Martin Potthast. 2020. Exploring argument retrieval with transformers notebook for the touche lab on argument retrieval at clef 2020.
- Robert Alexy. 1989. *A theory of legal argumentation: the theory of rational discourse as theory of legal justification*. Clarendon Press, Oxford.
- Sai Vamsi Aliseti. 2021. Paraphrase generator with t5. <https://github.com/Vamsi995/Paraphrase-Generator>.
- Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162.
- Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and N. Slonim. 2020. From arguments to key points: Towards automatic argument summarization. In *ACL*.
- Jordan Beck, Bikalpa Neupane, and John M. Carroll. 2019. *Managing conflict in online debate communities*. *First Monday*, 24(7).
- Gregor Betz, Christian Voigt, and Kyle Richardson. 2021. *Critical thinking for language models*. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*. Association for Computational Linguistics.

- Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible operations for natural language deduction. *ArXiv*, abs/2104.08825.
- Tracey Bowell and Gary Kemp. 2014. *Critical Thinking: A Concise Guide*, 4th edition edition. Routledge, London.
- Michael Bruce and Steven Barbone. 2011. *Just the arguments: 100 of the most important arguments in Western philosophy*. Wiley-Blackwell, Chichester, West Sussex, U.K.
- Georg Brun. 2014. Reconstructing arguments: Formalization and reflective equilibrium. *Logical Analysis and History of Philosophy*, 17:94–129.
- Georg Brun and Gregor Betz. 2016. Analysing practical argumentation. In Sven Ove Hansson and Gertrude Hirsch-Hadorn, editors, *The Argumentative Turn in Policy Analysis. Reasoning about Uncertainty*, pages 39–77. Springer, Cham.
- Tuhin Chakrabarty, Aadit Trivedi, and Smaranda Muresan. 2021. Implicit premise generation with discourse-aware commonsense knowledge models. In *EMNLP*.
- David Christensen. 2010. Higher-order evidence. *Philosophy and Phenomenological Research*, 81(1):185–215.
- Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, pages 3882–3890.
- Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.
- Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer.
- J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- L. Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, Yonatan Bilu, R. Aharonov, and N. Slonim. 2020. Corpus wide argument mining - a working solution. *ArXiv*, abs/1911.10763.
- Isabela Fairclough and Norman Fairclough. 2012. *Political Discourse Analysis*. Routledge, London.
- Richard Feldman. 1998. *Reason and Argument*, 2nd edition. Pearson, Prentice hall.
- Alec Fisher. 2004. *The Logic of Real Arguments*, 2nd ed edition. Cambridge University Press, New York.
- Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *ACL*.
- Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Christopher Pal. 2020. [Measuring systematic generalization in neural proof generation with transformers](#).
- Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, R. Aharonov, and N. Slonim. 2020. A large-scale dataset for argument quality ranking: Construction and analysis. *ArXiv*, abs/1911.11408.
- Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation mining on the web from information seeking perspective. In Elena Cabrio, Serena Villata, and Adam Wyner, editors, *ArgNLP 2014. Frontiers and Connections between Argumentation Theory and Natural Language Processing. Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*. CEUR-WS.org.
- Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *NAACL-HLT*.
- Sven Ove Hansson and Gertrude Hirsch-Hadorn, editors. 2016. *The Argumentative Turn in Policy Analysis. Reasoning about Uncertainty*. Springer, Cham.
- Aishwarya Kamath and R. Das. 2019. A survey on semantic parsing. *ArXiv*, abs/1812.00978.
- Nora Kassner and Hinrich Schütze. 2020. [Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.
- Joe Y. F. Lau. 2011. *An Introduction to Critical Thinking and Creativity: Think More, Think Better*. Wiley, Hoboken, N.J.
- John Lawrence and Chris Reed. 2020. [Argument Mining: A Survey](#). *Computational Linguistics*, 45(4):765–818.
- Hannes Leitgeb and André Carus. 2021. Rudolf Carnap. Supplement D: Methodology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Summer 2021 edition. Metaphysics Research Lab, Stanford University.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. [Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#).

- Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38.
- Marie-Francine Moens. 2018. [Argumentation mining: How can a machine acquire common sense and world knowledge?](#) *Argument & Computation*, 9:1–14.
- Jason Phang, Thibault Févry, and Samuel R. Bowman. 2018. [Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks.](#) *CoRR*, abs/1811.01088.
- Martin Potthast, Lukas Gienapp, F. Euchner, Nick Heilenkötter, Nicolas Weidmann, Henning Wachsmuth, Benno Stein, and Matthias Hagen. 2019. Argument search: Assessing argument relevance. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners.](#) *Preprint*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.
- J. Rodrigues, Ruben Branco, J. Silva, and A. Branco. 2020. Reproduction and revival of the argument reasoning comprehension task. In *LREC*.
- Ohad Rozen, Vered Shwartz, Roei Aharoni, and Ido Dagan. 2019. [Diversify your datasets: Analyzing generalization via controlled variance in adversarial datasets.](#)
- Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. [Prover: Proof generation for interpretable reasoning over rules.](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 122–136. Association for Computational Linguistics.
- Swarnadeep Saha, Prateek Yadav, and M. Bansal. 2021. [multiprover: Generating multiple proofs for improved interpretability in rule reasoning.](#) *ArXiv*, abs/2106.01354.
- Oliver R. Scholz. 2000. Was es heißt, eine argumentation zu verstehen? - zur konstitutiven rolle von präsumtionen. In Geert-Lueke Lueken, editor, *Formen der Argumentation*, pages 161–176. Leipziger Universitätsverlag, Leipzig.
- Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. [Bleurt: Learning robust metrics for text generation.](#) *arXiv preprint arXiv:2004.04696*.
- Richard Shin, C. H. Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, D. Klein, J. Eisner, and Benjamin Van Durme. 2021. [Constrained language models yield few-shot semantic parsers.](#) *ArXiv*, abs/2104.08768.
- Andrew Siegel. 2018. Ethics of Stem Cell Research. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2018 edition. Metaphysics Research Lab, Stanford University.
- Shahbaz Syed, Khalid Al-Khatib, Milad Alshomary, Henning Wachsmuth, and Martin Potthast. 2021. [Generating informative conclusions for argumentative texts.](#) In *FINDINGS*.
- Oyvind Tafjord and Peter Clark. 2021. [General-purpose question-answering with macaw.](#)
- Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. [Proofwriter: Generating implications, proofs, and abductive statements over natural language.](#) *arXiv preprint arXiv:2012.13048*.
- Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. [olmpics - on what language model pre-training captures.](#) *Trans. Assoc. Comput. Linguistics*, 8:743–758.
- Christian Voigt. 2014. Argdown and the stacked masonry layout: Two user interfaces for non-expert users. In *Computational Models of Argument*, pages 483–484, Amsterdam et al. IOS Press.
- Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. [Building an argument search engine for the web.](#) In *ArgMining@EMNLP*.
- 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. 2020. [Transformers: State-of-the-art natural language processing.](#) In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Synthetic Argument Data

The AAAC datasets used in this study are publicly available via Huggingface’s Hub – <https://huggingface.co/datasets/debatelab/aaac> – where the construction of the datasets is documented meticulously.

A synthetically generated AAAC record, which nicely illustrates the underdetermination of argument reconstruction, with two implicit premises,

one distracting statement and a simple (one-step) argument (formatted as presented to the model):

source: It is not the case that Tracy is not an admirer of Fullerton and Tracy has seen La Habra. Plus, if someone loves Chico, then they haven't visited Monterey, owing to the fact that loving Laguna Beach is sufficient for not having visited Monterey.

reasons: loving Laguna Beach is sufficient for not having visited Monterey (ref: (2))

conjectures: if someone loves Chico, then they haven't visited Monterey (ref: (4))

argdown:

(1) If someone is an admirer of Chico, then they are an admirer of Laguna Beach or a visitor of Stockton.

(2) If someone admires Laguna Beach, then they haven't visited Monterey.

(3) If someone has visited Stockton, then they haven't visited Monterey.

- with generalized dilemma (neg variant) from (1) (2) (3)

(4) If someone admires Chico, then they haven't visited Monterey.

premises: If someone is an admirer of Chico, then they are an admirer of Laguna Beach or a visitor of Stockton. (ref: (1)) | If someone admires Laguna Beach, then they haven't visited Monterey. (ref: (2)) | If someone has visited Stockton, then they haven't visited Monterey. (ref: (3))

conclusion: If someone admires Chico, then they haven't visited Monterey. (ref: (4))

premises_form: (x): Fx -> (G x v H x) (ref: (1)) | (x): G x -> not I x (ref: (2)) | (x): H x -> not I x (ref: (3))

conclusion_form: (x): F x -> not I x (ref: (4))

keys: F: admirer of Chico | G: admirer of Laguna Beach | H: visitor of Stockton | I: visitor of Monterey

B Training Set-up

By interpreting a generative mode as a sequence-to-sequence task, we may translate a multi-angular DeepA2 dataset (e.g., AAAC01) into a multi-task sequence-to-sequence format, on which a sequence-to-sequence model can be trained. For each record in the multi-angular DeepA2 dataset, we randomly sample 14 modes in accordance with the weights provided in Table 3 and add, for each mode, a corresponding sequence-to-sequence record to the training data. This results, for AAAC01, in a sequence-to-sequence training dataset with $14 \times 16,000$ records.

Our models (base model T5-large with 770M parameters, and pretrained ArgumentAnalyst) are

mode	$w_1 w_2$	mode	$w_1 w_2$	mode	$w_1 w_2$
S \rightsquigarrow A	1. 1.	S \rightsquigarrow R	1. 1.	P \rightsquigarrow F	.7 -
SR \rightsquigarrow A	1. 1.	SJ \rightsquigarrow R	1. 1.	PCO \rightsquigarrow F	.7 -
SJ \rightsquigarrow A	1. 1.	SA \rightsquigarrow R	1. 1.	C \rightsquigarrow O	.7 -
SRJ \rightsquigarrow A	1. 1.	S \rightsquigarrow J	1. 1.	CPF \rightsquigarrow O	.7 -
RJ \rightsquigarrow A	1. 1.	SR \rightsquigarrow J	1. 1.	PF \rightsquigarrow K	.7 -
PC \rightsquigarrow A	1. 1.	SA \rightsquigarrow J	1. 1.	CO \rightsquigarrow K	.7 -
A \rightsquigarrow P	.2 .2	A \rightsquigarrow C	.2 .2	PFCO \rightsquigarrow K	.7 -
FK \rightsquigarrow P	.7 -	OK \rightsquigarrow C	.7 -		

Table 3: 21 generative modes with corresponding weights in AAAC (w_1) and *EntailmentBank* (w_2) training data.

trained with batch-size 2 and learning rate 0.00001. For AAAC01, eval loss starts to increase at epoch 8; with *EntailmentBank* data, eval loss increases from epoch 2 onwards.

C Iterative Prediction with Generative Chains

Generative chains are implemented with a dynamic dictionary (9 keys, corresp. to the dimensions of DeepA2 data), which is initialized with the source text, provides input for the generative modes, and is updated after each generative step with the mode's generated output. Output is generated with beam search decoding and beam width 2.

Table 4 displays all generative chains we resort to in this study, all of which are used in the *first experiment*. The *second experiment* makes use of chains 1–11. The *third experiment* deploys chains 1–13.

D Additional Results

Table 5 assesses ArgumentAnalyst's reconstructions on specific subsets of the AAAC02 dataset (defined in Section 4) for three representative generative chains.

Table 6 details the performance of ArgumentAnalyst on the entire AAAC02 dataset as measured by tailor-made argumentative metrics. Table 7 shows the corresponding performance on out-of-sample eval data AAAC01.

Distinguishing four mutually exclusive subsets of AAAC02, Tables 8–11 detail the the quality of ArgumentAnalyst's reconstruction for easy and difficult problems. Tables 12–15 present the corresponding out-of-sample performance on the equally partitioned AAAC01 dataset (eval split).

#	mode sequence	len.	soph.
1	S~A S~R S~J	3	0
2	S~J S~R SJ~A	3	1
3	S~J S~R SR~A	3	1
4	S~J S~R RJ~A	3	2
5	S~J SJ~R RJ~A	3	3
6	S~J SJ~R SRJ~A	3	3
7	S~R SR~J RJ~A	3	3
8	S~R SR~J SRJ~A	3	3
9	S~A SA~R SA~J RJ~A	4	4
10	S~A SA~R SA~J SRJ~A	4	4
11	S~A SA~R SA~J SRJ~A SA~R SA~J SRJ~A	7	8
12	S~A A~P A~C P~F PF~K FK~P PC~A SA~R SA~J	9	11
13	S~A A~P A~C C~O CO~K OK~C PC~A SA~R SA~J	9	11
14	S~A A~P A~C C~O CO~K OK~C PC~A A~P A~C P~F PF~K FK~P PC~A SA~R SA~J	15	20
15	S~A A~P A~C P~F CPF~O PFCO~K FK~P OK~C PC~A SA~R SA~J	11	18
16	S~A A~P A~C P~F CPF~O PCO~F PFCO~K FK~P OK~C PC~A SA~R SA~J	12	21

Table 4: 16 generative chains (without final formalization sub-sequences) evaluated in this study. The illustrative chains highlighted in the main paper are #1 (straight), #9 (hermeneutic cycle), and #13 (logical streamlining).

chain	inference		presentation		C&M
	simple N=1274	compl. N=180	plain N=330	mutil. N=114	
SYS-PP & SYS-RP & SYS-RC & SYS-US					
straight	.95	.72	.98	.61	.69
herm. c.	.94	.68	.96	.67	.61
log. str.	.95	.68	.98	.64	.61
SYS-VAL					
straight	.84	.48	.88	.40	.34
herm. c.	.83	.56	.84	.49	.50
log. str.	.82	.47	.86	.46	.37
EXE-RSS					
straight	.03	-.25	.05	-.31	-.30
herm. c.	.20	.08	.15	.08	.11
log. str.	.17	-.01	.13	.01	-.06
EXE-JSS					
straight	.06	-.32	.10	-.37	-.37
herm. c.	.23	-.06	.21	-.03	-.21
log. str.	.13	-.26	.07	-.26	-.40

Table 5: Performance of ArgumentAnalyst on specific subsets (columns) of the AAAC02 data as measured by selected systematic and exegetic metrics (sub-tables). Rows display results for three illustrative generative chains (*straight*, *hermeneutic cycle*, *logical streamlining*).

chain	systematic metrics (SYS-*)						exegetic metrics (EXE-*)					
	PP	RP	RC	US	SCH	VAL	MEQ	RSS	JSS	PPR	PPJ	TE
#1	0.95	0.97	0.96	0.96	0.33	0.73	0.80	-0.08	-0.10	0.93	0.93	0.63
#2	0.95	0.97	0.94	0.94	0.33	0.71	0.80	-0.09	0.04	0.93	0.93	0.67
#3	0.95	0.98	0.95	0.93	0.31	0.70	0.80	0.10	-0.11	0.93	0.93	0.62
#4	0.94	0.97	0.94	0.92	0.30	0.70	0.80	0.12	-0.00	0.93	0.93	0.66
#5	0.94	0.97	0.95	0.91	0.30	0.70	0.83	0.13	0.05	0.94	0.93	0.69
#6	0.94	0.97	0.95	0.93	0.31	0.70	0.83	0.10	0.03	0.94	0.93	0.67
#7	0.93	0.97	0.95	0.92	0.29	0.70	0.83	0.13	0.05	0.93	0.92	0.68
#8	0.94	0.97	0.95	0.93	0.30	0.69	0.83	0.10	0.02	0.93	0.92	0.67
#9	0.95	0.98	0.95	0.93	0.31	0.72	0.82	0.16	0.12	0.93	0.92	0.71
#10	0.96	0.98	0.96	0.94	0.32	0.71	0.82	0.14	0.09	0.93	0.92	0.69
#11	0.96	0.98	0.96	0.93	0.32	0.71	0.82	0.15	0.11	0.93	0.92	0.71
#12	0.93	0.95	0.94	0.94	0.32	0.71	0.81	-0.17	-0.08	0.93	0.92	0.68
#13	0.95	0.97	0.96	0.95	0.32	0.72	0.82	0.11	-0.00	0.93	0.92	0.69
#14	0.93	0.95	0.94	0.94	0.32	0.70	0.81	-0.18	-0.14	0.93	0.92	0.66
#15	0.92	0.96	0.94	0.95	0.33	0.71	0.81	-0.20	-0.19	0.93	0.92	0.65
#16	0.92	0.96	0.94	0.94	0.33	0.72	0.81	-0.20	-0.19	0.93	0.92	0.65

Table 6: Performance of ArgumentAnalyst for systematic and exegetic metrics on the entire OOD eval data (AAAC02). Rows display mean results for each of the 16 generative chains.

chain	systematic metrics (SYS-*)						exegetic metrics (EXE-*)					
	PP	RP	RC	US	SCH	VAL	MEQ	RSS	JSS	PPR	PPJ	TE
#1	0.97	0.98	0.97	0.98	0.61	0.87	0.78	0.08	0.13	0.95	0.95	0.64
#2	0.97	0.98	0.96	0.97	0.60	0.87	0.78	0.09	0.24	0.95	0.95	0.68
#3	0.96	0.98	0.96	0.97	0.58	0.86	0.78	0.26	0.12	0.95	0.95	0.64
#4	0.95	0.98	0.95	0.96	0.57	0.85	0.78	0.26	0.20	0.95	0.95	0.67
#5	0.96	0.98	0.95	0.96	0.57	0.84	0.80	0.27	0.27	0.96	0.95	0.70
#6	0.97	0.98	0.96	0.96	0.58	0.84	0.80	0.26	0.24	0.96	0.95	0.69
#7	0.95	0.98	0.96	0.96	0.57	0.86	0.79	0.27	0.26	0.95	0.94	0.71
#8	0.96	0.98	0.96	0.96	0.57	0.85	0.79	0.26	0.25	0.95	0.94	0.70
#9	0.97	0.99	0.97	0.97	0.59	0.88	0.79	0.31	0.36	0.96	0.95	0.78
#10	0.97	0.99	0.97	0.97	0.60	0.87	0.79	0.30	0.34	0.96	0.95	0.77
#11	0.97	0.99	0.97	0.97	0.60	0.87	0.79	0.31	0.35	0.96	0.95	0.77
#12	0.95	0.97	0.95	0.96	0.54	0.84	0.79	0.17	0.25	0.96	0.94	0.75
#13	0.97	0.99	0.97	0.97	0.61	0.87	0.79	0.29	0.32	0.96	0.95	0.76
#14	0.95	0.97	0.95	0.96	0.54	0.84	0.79	0.16	0.24	0.96	0.94	0.74
#15	0.94	0.97	0.95	0.96	0.54	0.85	0.79	0.15	0.18	0.96	0.95	0.73
#16	0.94	0.97	0.95	0.95	0.54	0.85	0.79	0.15	0.19	0.96	0.95	0.73

Table 7: Performance of ArgumentAnalyst for systematic and exegetic metrics on the entire OOS eval data (AAAC01). Rows display mean results for each of the 16 generative chains.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
SYS-PP & SYS-RP & SYS-RC & SYS-US					
#1	0.95	0.72	0.98	0.61	0.69
#2	0.93	0.66	0.96	0.59	0.60
#3	0.92	0.69	0.96	0.68	0.73
#4	0.92	0.66	0.95	0.69	0.60
#5	0.92	0.68	0.95	0.59	0.61
#6	0.93	0.66	0.97	0.68	0.59
#7	0.92	0.67	0.96	0.62	0.64
#8	0.92	0.66	0.95	0.64	0.66
#9	0.94	0.68	0.96	0.67	0.61
#10	0.94	0.73	0.98	0.68	0.77
#11	0.94	0.69	0.98	0.66	0.73
#12	0.93	0.60	0.95	0.57	0.50
#13	0.95	0.68	0.98	0.64	0.61
#14	0.92	0.57	0.93	0.58	0.49
#15	0.92	0.66	0.95	0.59	0.56
#16	0.92	0.64	0.95	0.56	0.60

Table 8: Performance of ArgumentAnalyst for selected systematic metric (SYS-PP & SYS-RP & SYS-RC & SYS-US) on specific subsets (columns) of the OOD eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
EXE-RSS					
#1	0.03	-0.25	0.05	-0.31	-0.30
#2	0.02	-0.27	0.07	-0.33	-0.31
#3	0.15	-0.03	0.12	-0.01	-0.06
#4	0.16	0.01	0.12	-0.01	0.04
#5	0.18	0.04	0.13	0.04	0.06
#6	0.17	-0.04	0.12	-0.02	-0.09
#7	0.18	0.05	0.14	0.03	0.08
#8	0.16	-0.02	0.12	-0.02	-0.07
#9	0.20	0.08	0.15	0.08	0.11
#10	0.19	0.04	0.15	0.05	-0.01
#11	0.21	0.04	0.15	0.07	-0.03
#12	-0.14	-0.20	-0.12	-0.23	-0.25
#13	0.17	-0.01	0.13	0.01	-0.06
#14	-0.17	-0.22	-0.16	-0.23	-0.26
#15	-0.19	-0.23	-0.24	-0.24	-0.23
#16	-0.19	-0.23	-0.24	-0.25	-0.24

Table 10: Performance of ArgumentAnalyst for selected exegetic metrics (EXE-RSS) on specific subsets (columns) of the OOD eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
SYS-VAL					
#1	0.84	0.48	0.88	0.40	0.34
#2	0.82	0.54	0.84	0.47	0.46
#3	0.82	0.44	0.87	0.39	0.36
#4	0.81	0.48	0.83	0.44	0.43
#5	0.82	0.44	0.85	0.45	0.37
#6	0.81	0.46	0.85	0.42	0.41
#7	0.83	0.44	0.82	0.46	0.49
#8	0.80	0.44	0.83	0.40	0.40
#9	0.83	0.56	0.84	0.49	0.50
#10	0.82	0.50	0.85	0.46	0.43
#11	0.82	0.48	0.84	0.46	0.41
#12	0.81	0.47	0.84	0.42	0.37
#13	0.82	0.47	0.86	0.46	0.37
#14	0.80	0.48	0.82	0.41	0.40
#15	0.82	0.45	0.84	0.50	0.33
#16	0.83	0.52	0.85	0.46	0.43

Table 9: Performance of ArgumentAnalyst for selected systematic metric (SYS-VAL) on specific subsets (columns) of the OOD eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
EXE-JSS					
#1	0.06	-0.32	0.10	-0.37	-0.37
#2	0.16	-0.17	0.19	-0.12	-0.26
#3	0.02	-0.32	0.03	-0.42	-0.33
#4	0.12	-0.17	0.13	-0.14	-0.19
#5	0.15	-0.11	0.15	-0.08	-0.18
#6	0.16	-0.14	0.15	-0.22	-0.22
#7	0.16	-0.11	0.16	-0.10	-0.18
#8	0.15	-0.18	0.14	-0.19	-0.27
#9	0.23	-0.06	0.21	-0.03	-0.21
#10	0.23	-0.12	0.21	-0.15	-0.27
#11	0.25	-0.13	0.20	-0.11	-0.27
#12	0.06	-0.36	0.04	-0.28	-0.47
#13	0.13	-0.26	0.07	-0.26	-0.40
#14	-0.02	-0.39	-0.07	-0.31	-0.48
#15	-0.08	-0.41	-0.16	-0.36	-0.49
#16	-0.08	-0.37	-0.15	-0.35	-0.45

Table 11: Performance of ArgumentAnalyst for selected exegetic metric (EXE-JSS) on specific subsets (columns) of the OOD eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
	SYS-PP & SYS-RP & SYS-RC & SYS-US				
#1	0.98	0.78	1.00	0.75	0.76
#2	0.97	0.77	0.99	0.70	0.73
#3	0.95	0.79	0.96	0.77	0.74
#4	0.95	0.76	0.96	0.69	0.73
#5	0.97	0.75	0.98	0.66	0.74
#6	0.96	0.77	0.98	0.73	0.78
#7	0.96	0.73	0.96	0.71	0.72
#8	0.97	0.75	0.97	0.73	0.74
#9	0.98	0.80	0.99	0.80	0.70
#10	0.98	0.78	0.99	0.80	0.73
#11	0.98	0.78	0.99	0.80	0.71
#12	0.97	0.71	0.97	0.70	0.67
#13	0.98	0.81	0.99	0.76	0.78
#14	0.96	0.73	0.96	0.70	0.69
#15	0.97	0.72	0.96	0.70	0.68
#16	0.97	0.72	0.96	0.68	0.68

Table 12: Performance of ArgumentAnalyst for selected systematic metric (SYS-PP & SYS-RP & SYS-RC & SYS-US) on specific subsets (columns) of the OOS eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
	EXE-RSS				
#1	0.19	-0.16	0.11	-0.07	-0.18
#2	0.21	-0.13	0.10	-0.05	-0.15
#3	0.30	0.11	0.17	0.22	0.06
#4	0.29	0.16	0.16	0.24	0.16
#5	0.32	0.18	0.19	0.23	0.18
#6	0.31	0.11	0.18	0.19	0.07
#7	0.30	0.15	0.17	0.25	0.16
#8	0.30	0.12	0.17	0.24	0.08
#9	0.33	0.23	0.19	0.30	0.23
#10	0.33	0.20	0.19	0.27	0.16
#11	0.33	0.21	0.19	0.28	0.16
#12	0.20	0.06	0.11	0.16	0.04
#13	0.33	0.12	0.19	0.26	0.07
#14	0.20	0.06	0.10	0.16	0.03
#15	0.18	0.04	0.07	0.14	0.00
#16	0.18	0.04	0.07	0.11	0.02

Table 14: Performance of ArgumentAnalyst for selected exegetic metrics (EXE-RSS) on specific subsets (columns) of the OOS eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
	SYS-VAL				
#1	0.97	0.68	0.96	0.74	0.74
#2	0.97	0.68	0.97	0.73	0.71
#3	0.94	0.70	0.94	0.72	0.71
#4	0.95	0.65	0.94	0.68	0.71
#5	0.96	0.59	0.95	0.65	0.62
#6	0.95	0.62	0.96	0.69	0.63
#7	0.94	0.66	0.94	0.66	0.71
#8	0.95	0.67	0.95	0.69	0.69
#9	0.97	0.65	0.97	0.72	0.69
#10	0.97	0.67	0.97	0.68	0.72
#11	0.97	0.70	0.97	0.68	0.74
#12	0.95	0.63	0.95	0.72	0.70
#13	0.97	0.68	0.95	0.73	0.73
#14	0.95	0.63	0.94	0.72	0.69
#15	0.95	0.65	0.94	0.75	0.71
#16	0.95	0.65	0.95	0.73	0.71

Table 13: Performance of ArgumentAnalyst for selected systematic metric (SYS-VAL) on specific subsets (columns) of the OOS eval data.

chain	<i>inference</i>		<i>presentation</i>		C&M
	simple	complex	plain	mutilat.	
	EXE-JSS				
#1	0.35	-0.14	0.36	-0.09	-0.13
#2	0.40	0.02	0.39	0.10	0.02
#3	0.30	-0.15	0.29	-0.08	-0.15
#4	0.36	0.03	0.33	0.08	-0.02
#5	0.41	0.15	0.39	0.17	0.11
#6	0.40	0.04	0.38	0.10	-0.01
#7	0.39	0.12	0.37	0.15	0.06
#8	0.39	0.08	0.38	0.10	-0.02
#9	0.47	0.16	0.42	0.31	0.13
#10	0.47	0.11	0.42	0.26	0.02
#11	0.47	0.11	0.42	0.26	0.02
#12	0.40	-0.01	0.35	0.14	-0.08
#13	0.45	0.03	0.36	0.21	-0.01
#14	0.38	-0.00	0.30	0.15	-0.05
#15	0.30	-0.04	0.22	0.07	-0.07
#16	0.30	-0.03	0.22	0.11	-0.06

Table 15: Performance of ArgumentAnalyst for selected exegetic metric (EXE-JSS) on specific subsets (columns) of the OOS eval data.