A Neuro-Symbolic Cognitive System for Intuitive Argumentation

Vasanth Sarathy VSARATHY@SIFT.NET
Mark Burstein BURSTEIN@SIFT.NET
Scott Friedman FRIEDMAN@SIFT.NET
Robert Bobrow RBOBROW@SIFT.NET
Ugur Kuter UKUTER@SIFT.NET
Smart Information Flow Technologies (SIFT), Lexington, MA, USA

Abstract

Making and evaluating arguments is an important cognitive capability that plays a vital role in human cooperation by allowing us to communicate beliefs and persuade others. In this paper, we ask how can machines comprehend arguments. Research in computational argumentation has a rich history of addressing questions about the abstract dynamics of argumentation and how argument structures can be mined from text. However, what is missing is an integrated cognitive system that provides a computational account for how we might intuitively make sense of arguments. Work in cognitive psychology has suggested that when presented with a novel argument, humans interpret the language and produce acceptability/coherence judgments intuitively. In this paper, we introduce SKEPTIC, a computational implementation of the process of intuitive argumentation, combining modern deep neural networks and natural language processing techniques with established cognitive systems principles. We describe an architecture and an algorithm for extracting graph-like argument structures from raw unstructured text as well as retrieving implicit assumptions that support the argument. Being able to extract arguments and their implicit assumptions will be able to help us detect misinformation, reduce polarization, understand social and cultural rationales, and assist with critical thinking and persuasive writing.

1. Introduction

Consider the following argument:

> "*Cloning will be beneficial for many people who are in need of organ transplants. Cloned organs will match perfectly to the blood group and tissue of patients since they can be raised from cloned stem cells of the patient. In addition, it shortens the healing process as it is very rare to find an appropriate organ donor, and because by using cloning in order to raise required organs the waiting time can be shortened tremendously.*" (Stab & Gurevych, 2016)

This is a reasoned argument about cloning. It is "reasoned" in the sense that reasons are provided to support a pro-cloning conclusion. Humans make arguments like this constantly to communicate beliefs and goals, to convince others, and to have others perform actions. Arguments facilitate communication by overcoming disagreements, thereby enabling increased cooperation and better allocation of duties and rights (Mercier, 2011). Informally, an argument is a representation of a conclusion or a claim together with reasons (premises) for drawing the conclusion. When we comprehend others' arguments we evaluate if the reasons support the conclusions. The ability to automatically detect and evaluate arguments has the potential to help detect misinformation, reduce polarization, understand social and cultural rationales, and assist with critical thinking and persuasive writing. The central research question we ask here is *how can a machine comprehend and evaluate arguments present in unstructured natural language*? Is this argument convincing? What assumptions is it making? We propose SKEPTIC, a cognitive system that comprehends arguments by extracting structured representations (e.g., Figure 1) showing relationships between the different spans of text, as well as generating novel textual content to supplement and rationalize the argument.

Toulmin's seminal work in argumentation research proposed a schema outlining various components of an argument and their relationships, which served as a foundation for many different computational approaches (Toulmin, 2003). Subsequently, logicians focused on abstract argument structures looking at how arguments strengthen and weaken each other (Dung, 1995). These approaches, while formally sound did not tackle the problem of extracting argument structures from raw and messy speech. More recently, there have been approaches aimed at mining argument structures from text (Stab & Gurevych, 2016; Peldszus, 2017). However, these approaches (a) do not provide an integrated cognitive systems account of argumentation, which includes not only mining arguments and working out the logic involved, but also their relationships with background knowledge, and implicit assumptions, and (b) do not engage in the process of rationalization and inquiry, which humans do naturally when making sense of arguments.

To address this gap, we turn to cognitive psychology and particularly the work of Mercier and colleagues who have developed the rubric of the Argumentative Theory of Reasoning, which ties reasoning to argumentation in humans(Mercier & Sperber, 2011) . Under this framework, they argue that argument production and evaluation are *intuitive inferential processes*. When presented with a novel argument, humans interpret the language and produce acceptability/coherence judgements intuitively, without consciously reasoning about argument validity. What reaches awareness is a fully-constructed argument structure that supports the already intuited judgements. Subsequent conscious/deliberate reasoning may be performed over these structures, to further strengthen the reasoning or assimilate updates - so evaluation can continue non-intuitively, as well. What is missing from Mercier et al.'s approach is a computational deeper diver into how exactly arguments are produced and evaluated with intuitive inference and reasoning. With SKEPTIC, we provide a first computational account of the process of intuitive argumentation, combining modern deep learning and natural language processing techniques with established cognitive systems principles.

In the rest of the paper, we will discuss our proposed SKEPTIC architecture discussing the theory and implementation behind each component. We will also provide some formal definitions and an algorithm for how SKEPTIC coordinates these components to generate an argument graph like the one in Figure 1 from raw unstructured text. In addition to the "cloning" example, we will introduce several shorter examples of arguments throughout the paper to illustrate various challenges and capabilities for effectively intuiting arguments.

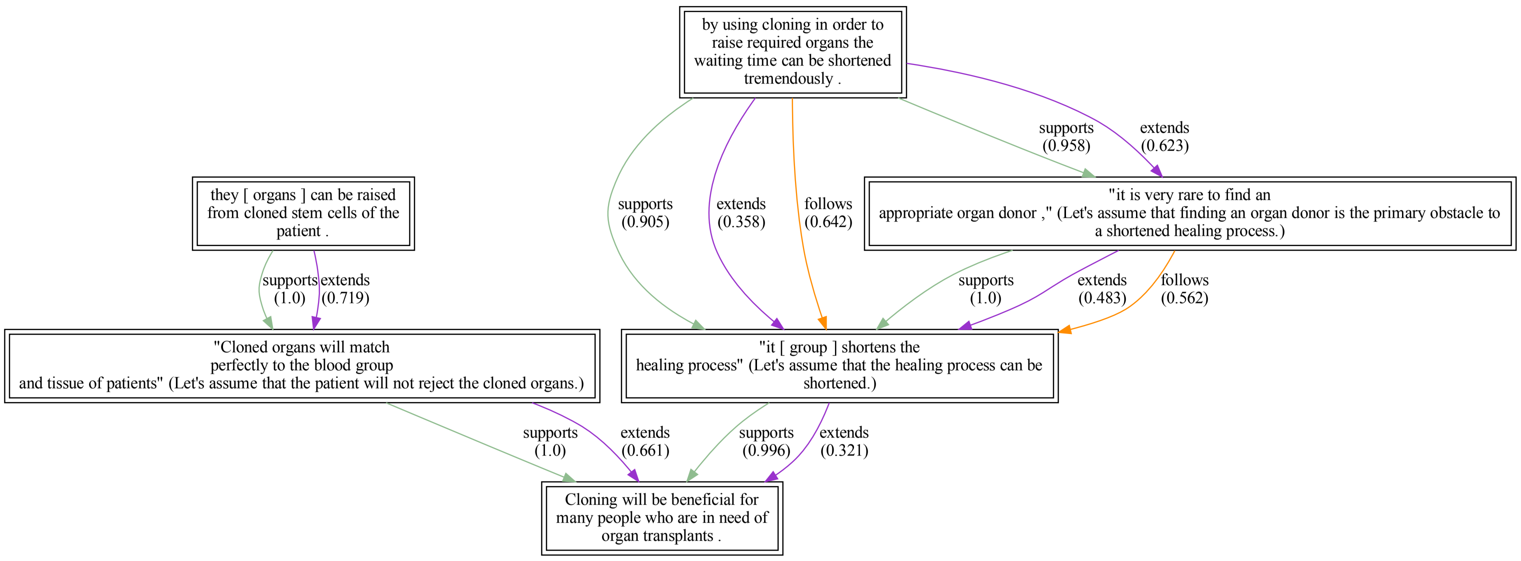

by using cloning in order to raise required organs the waiting time can be shortened tremendously .
they [organs] can be raised from cloned stem cells of the patient .
"it is very rare to find an appropriate organ donor ," (Let's assume that finding an organ donor is the primary obstacle to a shortened healing process.)
"Cloned organs will match perfectly to the blood group and tissue of patients" (Let's assume that the patient will not reject the cloned organs.)
"it [group] shortens the healing process" (Let's assume that the healing process can be shortened.)
Cloning will be beneficial for many people who are in need of organ transplants .
supports (0.958)
extends (0.623)
supports (0.905)
extends (0.358)
follows (0.642)
supports (1.0)
extends (0.719)
supports (1.0)
extends (0.483)
follows (0.562)
supports (1.0)
extends (0.661)
supports (0.996)
extends (0.321)

Figure 1. Argument Structure automatically generated by SKEPTIC for the unstructured natural language argument: "*Cloning will be beneficial for many people who are in need of organ transplants. Cloned organs will match perfectly to the blood group and tissue of patients since they can be raised from cloned stem cells of the patient. In addition, it shortens the healing process as it is very rare to find an appropriate organ donor, and because by using cloning in order to raise required organs the waiting time can be shortened tremendously.*"The nodes in the graph represent Argument Discourse Units (ADUs) that are spans of text in the original input as well implicit assumptions not in the original text, and the edges represent a variety of argument-relevant relations that help ensure that the argument can be rationalized. The parentheticals beginning with "let's assume that" are automatically generated text representing the implicit assumptions made in the argument. The words in the square brackets are coreference mentions, particularly useful for identifying anaphoric mentions.

2. SKEPTIC Architecture

We describe a cognitive system that takes unstructured text (written or transcribed from speech) as input and produces a rich structured graph-based representation capturing relevant relationships between the premises and the claims in the text. SKEPTIC uses pre-trained neural language models or so-called "Foundation Models" (Bommasani et al., 2021) (FMs) for both inference as well as for associative machinery to retrieve implicit knowledge to interpret and flesh out the natural language argument.[1] Foundation models for language (like BERT, T5 and GPT-3) are neural network language models trained to predict the next word (or a masked word) with very large datasets (GPT-3 has trained on 500 billion word tokens). Research has found state-of-the-art performance with foundation models on a variety of downstream tasks. Using natural language and graph-based representations as its *interlingua*, SKEPTIC features several components (Figure 2) whose operations are coordinated by a central Manager component, which in turn can be tied to dialog and vision systems of broader cognitive architectures.

[1] Here, we adopt the term "foundation model" as it captures not only the underlying technical approach of pre-training but also their emerging impact on the practice of AI research and development itself (Bommasani et al., 2021)

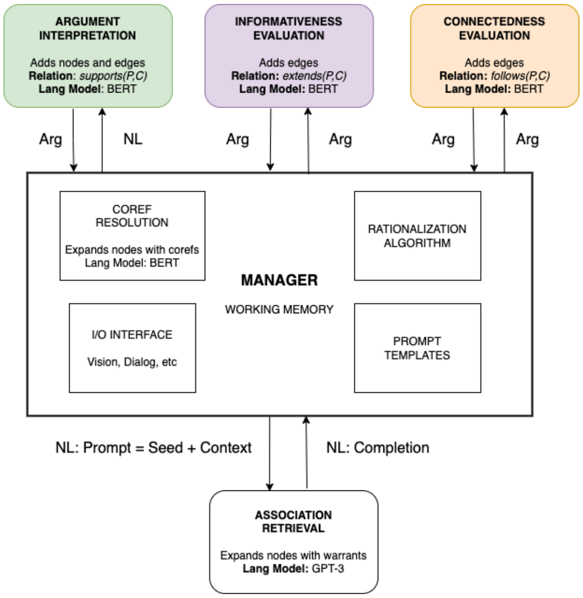


Figure 2. SKEPTIC Architecture: The Manager receives raw unstructured text and progressively rationalizes the argument presented in the text by (1) interpreting it (ARGUMENT INTERPRETATION: green) to generate graph-like structures representing premise-conclusion pairs connected by a *supports* edge, (2) evaluating whether the conclusion provides additional information above and beyond the premise, while still staying tethered to it (INFORMATIVENESS EVALUATION: purple) and adding a *extends* edge, and (3) evaluating whether the conclusion follows from (e.g., entailed by) the premise (CONNECTEDNESS EVALUATION: orange). If there are missing *extends* and *follows* edges, then the Manager directs the generation of implicit assumptions to provide additional information or additional rationale to strengthen the argument.

We distinguish intuitive processes from deliberative ones in line with the extensive literature on dual-process systems (Evans, 2003), and their analogies to current AI approaches. Neural net models, due to their lack of explicability and transparency, can be considered a type of intuitive processing, and symbolic systems can be considered deliberative. There is much debate to be had about exactly what it means for something to be intuitive for SKEPTIC, but because we (and thus SKEPTIC) lack the ability to understand and introspect on the vector embeddings present in the neural nets, we consider their operation to be intuitive processing. We claim that SKEPTIC is a cognitive architecture that intuitively (using neural nets) generates arguments. Note, we are not proposing an end-to-end neural model, in which any functional cognitive architecture is implicit in the layers of the neural net. Instead, we have an explicit cognitive architecture shown in Figure 2 with functional units and a meta algorithm executed by the Manager. What is implicit are the inner workings of these various functional components. This distinction is very important as we think there is very little scientific value in training an end-to-end system to fit a dataset. Instead, we are interested in the dual-purpose of a functional implementation, but also a possibly deeper understanding of how humans rationalize arguments. Splitting up the components and proposing an explicit cognitive architecture, we believe, is a necessary step in that process.

2.1 Argument Structure

Toulmin proposed that arguments are composed of **claims** (or **conclusions**) that are based on **grounds** (or **premises**), and supported by **warrants** comprising the reasoning that authorizes the inferential leap from the grounds to the claim(s) (Toulmin, 2003).[2] Others have elaborated this schema with specific types of relationships between the grounds and the claim. For example, Peldszus & Stede (2013) proposes several different types of support and attack relationships that work independently to allow premises (a.k.a, grounds) to either support or attack claims independently or in concert with other claims. Stab & Gurevych (2016) have argued that natural language argument often features simpler structures than what Peldszus (2017) has proposed. For the purposes of our current work, we follow Stab & Gurevych (2016) and focus on the simpler argument structure, one which informally involves a premise and a conclusion connected with a directional relation. In the next few sections, we discuss different components that add to this argument structure, either in the form named relationships between premises and conclusions or the nature and content of the premises and conclusions themselves.

For example, the argument *"The building is full of asbestos. We should tear it down"* is a simple argument that might naturally break down into two parts (a.k.a. ADU or Argument Discourse Units): (1) *"the building is full of asbestos"* and (2) *"we should tear it down."* We also know intuitively that the first ADU is the premise and the second is a conclusion supported by the premise. Figures 1 and 4 show example argument structures generated by SKEPTIC. In the next few sections, we will describe each of the components of the SKEPTIC architecture of Figure 2 that generated this parse.

[2] There are other aspects in Toulmin's model including bases, qualifiers and rebuttals, but we will table those types of relations for the time-being, focusing instead on the core grounds-claim + warrant aspect.

2.2 Argument Interpretation – *supports(P,C)*

Argument interpretation is a process that extracts a surface-level parse of the argument. SKEPTIC's argument interpretation produces a graph-based argument structure: nodes are spans of texts representing ADUs (Argument Discourse Units) with an attribute labeling of whether the ADU is a **premise** or a **claim**; edges are directed **supports** and **attacks** relations between ADUs. The **supports** and **attacks** relations allow the reader to infer that a certain argument *exists* in the text, as **supports** and **attacks** are the most basic relations capturing the dialectic notion that the listener (or reader in the case of written text) can infer that the speaker is making an argument. As Peldszus (2017) has noted, "[i]n the basic dialectical situation, the support relation is triggered by the opponent challenging the presented claim by asking for a reason." In a way, argument interpretation is at the core of what makes an argument a reasoned one. SKEPTIC's argument interpretation process extracts these **supports** and **attacks** relations from the linguistic form and style of human speech.

SKEPTIC uses the SpEAR (Span-based Entity, Attribute and Relation Extraction) neural transformer architecture (Friedman et al., 2021) to automatically (1) identify ADUs (i.e., textual spans) from the unstructured text, (2) label those ADUs with either **premise** or a **claim** attributes, and (3) connect ADUs with **supports** and **attacks** relations.

We train and test SKEPTIC's SpEAR-based argument interpretation using data from the Persuasive Essays Corpus (Stab & Gurevych, 2016) that contains 300 short essays (total of 1,506 claims and 3,832 premises) coupled with labeled graphs showing **supports** and **attacks** relations between spans of text. Prior corpora for training SpEAR included data that was typically short single-sentence inputs and causal relations that connected short phrases or single words to each other. The arguments in the Persuasive Essays Corpus (and other argument corpora), however, are typically longer both in terms of (1) length of the ADUs (i.e., extracted spans) and (2) the distance between related (via **supports** and **attacks**) spans in the input text. To address the challenge of span length, SpEAR instead identifies the first and last tokens of each ADU and it automatically joins them into a longer span, thereby transforming the single-entity detection into an entity-boundary-detection task. Since the Persuasive Essays Corpus' ADU spans are much longer than spans normally extracted by SpEAR and NER models, SpEAR equivalently learns to identify the boundary tokens of an ADU (**ADU-start** and **ADU-end** tokens) and relate them via a specialized **endOf** relation from each **ADU-start** to its corresponding **ADU-end**, and then it automatically compose an entire ADU. This boundary-learning strategy allows SpEAR to identify much longer spans than previously possible. Overall, the Argument Interpretation component (shown in green in Figure 2 extracts a graph-like structure with the **supports** and **attacks** relations (see green edges in Figures 1 and 4) between relevant ADUs.

Argument interpretation results are shown in Table 1 using a 90% training split and a 10% testing split over the Persuasive Essays Corpus, substituting different pre-trained transformer variants into SpEAR. All models reliably extract ADUs from text at 0.92 F1 score, and the uncased SciBERT model outperforms other variants on attribute prediction and relation prediction despite having fewer parameters than the large BERT variants. We believe this is because SciBERT's vocabulary is tuned to scientific language, which occurs within the Persuasive Essays Corpus.

	ADUs			**Attributes**			**Relations**		
Base Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
bert-base-uncased	89.94	**94.40**	92.11	74.47	77.75	76.08	71.85	73.39	72.61
bert-large-uncased	90.62	93.75	92.16	75.27	77.75	76.49	**73.39**	70.62	71.98
bert-large-cased	**90.73**	93.86	**92.27**	74.31	77.09	75.68	73.24	72.13	72.68
scibert-uncased	89.66	94.40	91.97	**75.62**	**79.96**	**77.73**	72.91	**73.64**	**73.27**

Table 1. Micro **P**recision, **R**ecall, and **F1** of ADU prediction, Attribute prediction (labeling **claim** and **premise** on ADUs), and Relation prediction (labeling **supports**, **attacks**, and **endOf** across ADUs) on a 90%/10% train/test split of the Persuasive Essays Corpus. Scores multiplied by 100 for alignment.

2.3 Informativeness Evaluation – *extends(P,C)*

Much of the current research on computational argument mining begins and ends with argument interpretation, namely the extraction of supports and attacks relationships from text, as discussed earlier. However, humans are good at intuitively identifying a bogus argument that might have the surface appearance of an argument but is not one. For example, argument interpretation alone would parse the argument: *"The sky is blue. Therefore, the sky is blue"* as two ADUs, each containing the phrase *"the sky is blue"* connected by a supports relation. While the structure of the argument and the connective *"therefore"* suggest the existence of an argument, we intuitively realize that there is no additional information (novelty) in the claim over and above the premise. In fact, these are simply restatements of each other. There is a sense that claims do more than restate a premise. But, on the flip side, claims must be tethered to, (have some bearing on) the premise. An argument like *"The sky is blue. Therefore, my dog is hungry."* is just as difficult to accept as the claims appear to have no bearing on the premise. Let's discuss each aspect – bearing on, and novelty – in turn.

2.3.1 "Pertinence" or "has bearing on"

The notion of has-bearing-on is a semantic relation that encapsulates relevance and similarity. However, it is potentially a broader notion covering the pragmatics of argumentation in which the speaker is attempting to convince the listener that there was a reason they presented this particular argument. We can ask why the speaker would have reached the conclusion given the premise. Why are they talking about hunger of their dog and the blueness of the sky? This notion is linked to the Gricean conversational maxim of "relevance", in which one tries to be relevant, and says things that are pertinent to the discussion. We might be able to establish that one ADU has bearing on another if they share some semantic similarity either at the lexical level or at a conceptual level (sunny and blueness of the sky). If we are unable to find such a similarity, we might need to extract and surface intermediate warrants and attempt to establish bearings between the original ADUs and warrants. This is a recursive mechanism (not unlike a toddler repeatedly asking why) and one that might continue until the system can determine the argument structure. SKEPTIC does engage in this recursive mechanism, which will be discussed later.

2.3.2 "Novelty"

By now, we hope you agree that claims must have some bearing on their premise. But what about our tautological argument presented earlier with identical premises and claims? These certainly have bearing on each other, but they lack novelty. It captures the desirable virtue of moving "forward" an argument by extending what was said – having a conclusion go beyond the premise. The idea is that here the conclusion is adding something even if that is just tying together concepts. Another way to think about novelty is in relation to bearing-on. While bearing-on aims to bring ADUs together (in some sense maximize similarity), novelty is attempting to refine and pull ADUs apart.

In this paper, we operationalize the notions of novelty and pertinence together with the relation **extends**. We implement this as a neural network for adjudicating semantic similarity. To do so, we used a sentence-transformer neural network model, which maps sentences and paragraphs to a 768 dimensional dense vector space and can be used for tasks like clustering or semantic search (Song et al., 2020). Specifically, computing semantic similarity between two texts with Sentence Transformers involves converting the two texts into individual vector representations, and then use a metric like cosine similarity to determine the similarity between the two vectors we're comparing. The output provides a numeric score between 0 and 1, where 0 suggests there is no similarity and 1 suggests there is large amount of overlap. To balance novelty with pertinence, we implement thresholds (e.g., 0.25 and 0.75) that allow us to filter out pairings that are too novel or too similar.

2.4 Connectedness Evaluation

Mercier et al. have noted that humans make judgments about arguments intuitively, including certain aspects of their quality. Thus far, we have focused on argument structure from the standpoint of surface level interpretation as well as from whether the argument holds together in terms of the conclusions tethered to but extending from the premise. Next, we will discuss how we can operationalize our intuitive judgements about whether the argument is logically/temporally/causally connected. That is, do the conclusions **follow** from the premise and what inferential role is played by premise in relation to the conclusion? This notion of "connectedness" serves as a starting point for evaluating the strength or believability of arguments. We operationalize this notion of connectedness by introducing the relation **follows**. We implement this in a neural network capable of evaluating textual entailment. To do so, we use a version of the large masked language model, RoBERTa-large fine-tuned to the task of natural language inference (Liu et al., 2019; Bhargava et al., 2021), specifically, the multi-genre natural language inference corpus (MNLI) (Williams et al., 2018). Given a premise sentence and a hypothesis sentence, the model predicts whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). It is worth noting here, that we use the term **follows** to reference the relation rather than "entails" because entailment here is not strict in the formal or semantic sense of textual entailment.

Some researchers refer to this idea of "connectedness" as "validity" however, we see validity as limited to deductive notions. For example, the following argument could be plausible, not in a deductive sense, but in an inductive sense: *"This swan is white. Therefore, all swans are white."* This is a reasoned argument, not necessarily one that you will entirely agree with, but it is a reasonable argument with some inferential validity. In fact, humans make these sorts of inductive as well as abductive arguments such as *"This swan is white. It must be the salts in the water."* The crucial aspect here is that many natural human arguments are "defeasible." If we find a non-white swan, then this argument fails plausibility. There is potentially a myriad of reasons that people find arguments persuasive, which in turn are dependent on other factors like the credibility of the writer, the number of pieces of evidence presented, the reader's background knowledge, and the state of mind of the reader, among other things. While we do not address all these factors, we believe this notion of "connectedness" and identifying the **follows** relation will serve as a foundation that closely ties together naturalist argumentation and computational capabilities.

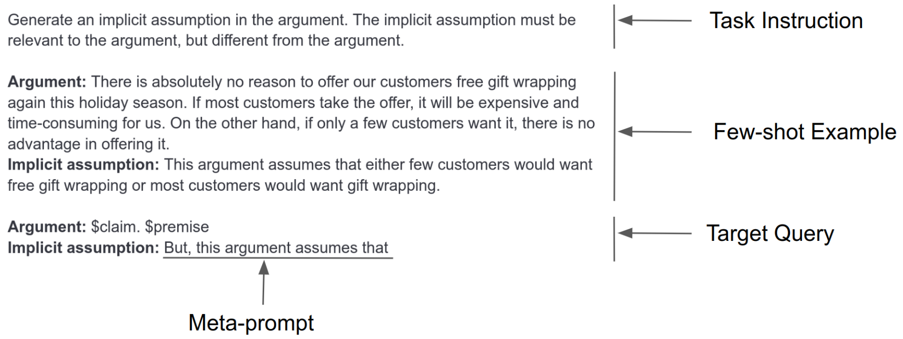
2.5 Associative Memory Retrieval

Intuitive judgments about structure, information and connectedness allow us to evaluate the general nature and quality of argument. However, humans are able to regularly rationalize even the most disconnected and structurally unsound arguments. Take for example, our previous example: "The sky is blue. Therefore my dog is thirsty." If we assume that the argument is being made by someone (say a trusted friend named "Rae") engaged in cooperative dialogue (in a Gricean sense) and we generally trust what she says, then we intuitively attempt to rationalize this argument. We do so by filling in any perceived gaps with our own associations derived from our prior experiences. For example, we might expand this argument in our heads as follows: *"Rae's dog is thirsty. That must be because the dog was playing outside in the heat. I know the dog likes to play outside when it is sunny. It is sunny and likely hot outside because the sky is blue. While playing outside, the dog probably got quite dehydrated from the heat."*.

There are infinitely many possible rationalizations, each depending on the hearer's own associative machinery and other contextual factors. A computational framework for argument must be able to account for this process of rationalizing and also provide a mechanism to extract relevant experiential associations to help us make sense of the argument. Here we propose using trained neural *Foundation Models* for language to serve as our associative memory and a **prompting** as our memory retrieval process. Broadly, FMs for language are deep neural language models[3] pretrained on a large amount of unlabelled text in a self-supervised setting. There are a variety of transformer-based language models including GPT-3 (Brown et al., 2020), BERT , BART , T5 , and PaLM , among others. FMs pre-trained on a large corpus of web data have been shown to contain different kinds of knowledge implicitly in their parameters without the need for any human supervision, including world knowledge, relational knowledge, commonsense knowledge, linguistic knowledge and actionable knowledge.

We selected GPT-3 for this paper because we believe that FMs can serve as a valuable associative memory, unlike other formally specified knowledge bases. For example, humans associate the word caterpillar with not only the fact that it's an insect, but also other characteristics from their own past experiences. GPT-3 makes such similar associations by linking *"caterpillar"* with not only static properties, but other concepts such as *"slime"*. Modifying the input with the word *"hungry*

[3] Language models are basically a machine learning model that is able to look at part of a sentence and predict the next word. Current state of the art language models are implemented as neural transformer architectures. Transformer is a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output .



Generate an implicit assumption in the argument. The implicit assumption must be relevant to the argument, but different from the argument.

Argument: There is absolutely no reason to offer our customers free gift wrapping again this holiday season. If most customers take the offer, it will be expensive and time-consuming for us. On the other hand, if only a few customers want it, there is no advantage in offering it.
Implicit assumption: This argument assumes that either few customers would want free gift wrapping or most customers would want gift wrapping.

Argument: $claim. $premise
Implicit assumption: But, this argument assumes that

Figure 3. GPT-3 prompt template used by SKEPTIC. The prompt is composed of four main parts: (1) the task instruction for providing the FM with the overall task description, (2) prompt prefix with zero or more examples, (3) target query with variables $claim and $premise for dynamically populating the premise-conclusion pair, and (4) a metaprompt to guide the FM towards generating completions that align with our intended output.

caterpillar", invokes associations like Eric Carle's children's book – A very Hungry Caterpillar, a natural association for humans with little children at home. Experiential memory of this sort is closer to human memory than other static knowledge bases such as ATOMIC, ConceptNet, Wordnet, Visual Genome, CSKG, and the like, which tend to not take context-based retrieval into account (Ilievski et al., 2021).

Here, we propose using GPT-3 to extract implicit warrants (implicit assumptions) from argument premise-conclusion pairs. The imagined argument of Rae's dog referenced earlier contains such implicit assumptions (albeit written manually) including rule-like structures such as *"the dog likes to play outside when it is sunny"*, events such as *"the dog was playing outside"* and default inferences such as *"...probably got quite dehydrated..."*. We extract such implicit assumptions automatically by **prompting** the FM (in this case, GPT-3) with a task instruction along with a premise-conclusion pair. Figure 3 shows an example prompt template used by SKEPTIC.

Prompting is the defacto technique for extracting knowledge from FMs like GPT-3. Prior to GPT-3, the standard approach to the evaluation and use of such models has involved fine-tuning on a portion of a task dataset. GPT-3 achieved state-of-the-art performance on a wide variety of tasks without fine tuning, using as few as zero examples (Reynolds & McDonell, 2021; Liu et al., 2021). Being a language model, GPT-3 is designed to predict the next word token in a sequence, which can then serve as an input to subsequent predictions in an autoregressive manner. This input is often called a prompt. The fast moving field of prompt programming involves techniques for generating prompts that appropriately match the user's intent and desired use case.

Prompt engineering and design is a nascent subfield and the relationships between the prompts and their completion are only now being learned and discovered. That said, some general principles and best practices are beginning to emerge in the literature. One practice is to give the model enough context on the task in the text input. Because FMs are trained on natural language in a self-supervised manner (without human labels), they largely mirror how humans (collectively) use language. So, prompts tend to generate completions closer to the user's intent when they are written in a natural style. Additional research has shown the importance of meta-prompting (Wei et al., 2022) as well. We have incorporated some of these best practices in how we prompt GPT-3 to generate warrants in SKEPTIC. Figure 3 shows our template for prompting in which GPT-3 will complete the sentence (or meta prompt) beginning with "But, this argument." The text leading up to the meta prompt forms the main body of the prompt and serve as task-specific context to steer the FM towards a completion that more closely aligns with the user's intent. In addition to designing the prompt text, FMs also enable users to tune certain hyperparameters including those for reducing the likelihood of sampling repetitive sequences of tokens (frequency and presence penalties) and control the randomness and creativity of the model's predictions (temperature).

2.6 Manager

The Manager coordinates the construction and evaluation of argument structures by running the rationalization algorithm that we will discuss next. Broadly, the manager obtains an initial argument parse from the Argument Interpretation component and then evaluates, using the Informativeness and Connectedness Components whether for each premise-conclusion pair (edge of the graph) the conclusion extends and follows from the premise. For those edges, where the conclusion does not either extend or follow, the Manager will direct the generation of implicit warrants by the Associative Memory Retrieval component. The manager explicitly includes the identified implicit warrants into the corresponding premise and re-evaluates for informativeness and connectedness. Prior to these evaluations however, when a new body of unstructured text is received, the Manager first performs coreference resolution – identifying text mentions that are referring to the same entities. Coreference resolution is necessary here as we will be splitting up the text into Argument Discourse Units (ADUs) – spans of text from the original text input – which may contain pronominal and other anaphoric referential expressions, which are semantically empty. By performing coreference resolution and surfacing the entity clusters, the semantics of these individual ADUs can be made explicit and therefore each premise-conclusion pair can be handled and evalulated independently. For coreference resolution we use Coreferee, a coreference resolution system developed under the spaCY library (Hudson, 2022; Honnibal et al., 2020).

2.6.1 Formal Definitions

Thus far, we have discussed the operation of various components of the proposed architecture. In doing so, we have referenced several terms – such as argument, span, ADU – rather informally. In this section we provide certain definitions for these terms more concretely, which in turn will be useful for the algorithm. Consider an unstructured **text document** D. D is comprised of a sequence of **words** $D = \langle w_0, w_1, \dots w_n \rangle$ that can be composed into contiguous **spans** of words $S = \langle w_i, \dots, w_j \rangle, 0 \leq i < j \leq n$. We can define an **argumentation discourse unit** (ADU) as a span S that is argumentatively relevant and have its own argumentative function or role R (Peldszus, 2017). Formally, $ADU = (S, R)$. For example, when $D =$ "The sky is blue. Therefore, my dog is thirsty", we might have a two ADUs, $ADU_1 =$ ("sky is blue",$premise$), $ADU_2 =$ ("my dog is thirsty", $conclusion$). Informally, an ADU's role could be a $premise$, $claim$, or any other type/property of a span of text together with the particular role (premise, claim, implication, presupposition) it plays

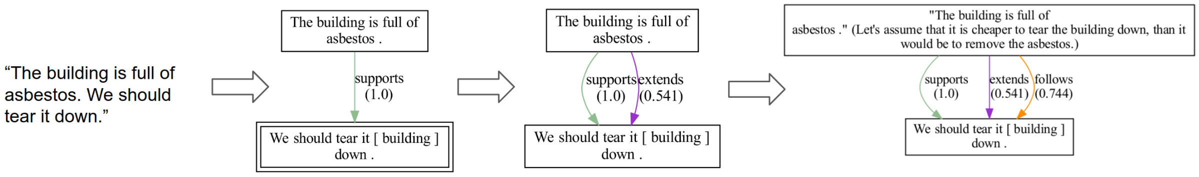

“The building is full of asbestos. We should tear it down.”
The building is full of asbestos .
supports (1.0)
We should tear it [building] down .
The building is full of asbestos .
supports (1.0)
extends (0.541)
We should tear it [building] down .
"The building is full of asbestos ." (Let's assume that it is cheaper to tear the building down, than it would be to remove the asbestos.)
supports (1.0)
extends (0.541)
follows (0.744)
We should tear it [building] down .

Figure 4. The Manager (Figure 2) uses Algorithm 1 to rationalize the unstructured text argument by first extracting an argument structure and then stretching the premises with implicit warrants in order to make the rationalization sturdier, that is, increase the number of relations between nodes from just **supports** relations to also include **extends** and **follows** relations

in the argument. We define **argument** A as a *directed labeled multigraph* $A = (V, M)$, where nodes V are ADUs and the multiset $M = (E, m)$ represents relations between ADUs. Roughly, the edges E (and associated labels m) captures the idea that ADUs can be linked together in different ways, each way contributing to the “goodness” of the argument. In our running examples, we have had three types of labels **supports**, **extends** and **follows**.

Given an argument A, we propose a measure of $sturdiness$ that captures the idea that an argument is stronger if the links between the premises and conclusions are supported by several types of edge labels. That is, a premise-conclusion pair that only contains a **supports** edge is weaker than one that contains both a **supports** edge and a **extends** edge. As we will see in the next section describing the algorithm, this measure of sturdiness will allow us to expand and rationalize those edges that are weaker.

Intuitively, the sturdiness scoring takes each vertex (ADU) of the argument and determines for each in-edge and each out-edge what is the fraction of types that exist against a total of three types. The per-edge sturdiness score is then an average of the in-edge sturdiness and out-edge sturdiness. The total sturdiness is the average sturdiness over all vertices. Formally, we can restate this intuition as follows:

$$sturdiness(A, n) = \frac{1}{|V|} \sum_{i=1}^{|V|} \left[\frac{1}{2} \left\{ \left(\frac{1}{|in(v_i)|} \sum_{j=1}^{|in(v_i)|} \frac{types(e_{ji})}{n} \right) + \left(\frac{1}{|out(v_i)|} \sum_{j=1}^{|out(v_i)|} \frac{types(e_{ij})}{n} \right) \right\} \right]$$

where, $A = (V, M)$ is the argument, n is the total number of types an edge can have (e.g., for this paper $n = 3$), $types(e_{pq})$ is a function that returns the number of types of edges that exist between vertices v_p and v_q

2.6.2 SKEPTIC *Rationalization Algorithm*

We now turn to the central operation of the Manager, which implements the rationalization Algorithm 1. We will use our example of asbestos removal, shown in Figure 4, to explain the algorithm. The algorithm takes an unstructured text document D as input and any user-defined parameters (sturdiness thresholds, max number of iterations to try before exiting). As a first step, it resolves coreferences as described earlier and updates the text document D to include the most specific

Algorithm 1: Argument Rationalization (Manager) Pseudocode

```
   Input  : Text document $D$, Parameters $params$
   Output: Argument Multigraph $A$
1  $D \leftarrow$ resolveCorefs($D$)
2  $A \leftarrow$ generateInitArgument($D$)
3  $sturdiness \leftarrow$ computeSturdiness($A$)
4  $counter \leftarrow 0$
5  if $A$ has edges then
6      while $sturdiness < params.threshold$ and $counter < params.maxIter$ do
7          for $(premise, conclusion)$ in $sturdiness.pairs$ do
8              $warrantFound \leftarrow False$
9              $warrantCounter \leftarrow 0$
10             while $\neg warrantFound$ and $warrantCounter < params.warrants$ do
11                 $warrant \leftarrow$ generateWarrant($premise, conclusion$)
12                 $connected \leftarrow$ isConnected($premise + warrant, conclusion$)
13                 $informative \leftarrow$ isInformative($premise, warrant$)
14                 $warrantFound \leftarrow connected \wedge informative$
15                 $warrantCounter += 1$
16             end while
17             if $warrantFound$ then
18                 $premise \leftarrow premise +$ (“Let’s assume that” $+ warrant$)
19                 $A \leftarrow$ updateArgument($A, premise$)
20                 $A \leftarrow$ evalConnectedness($A$)
21                 $A \leftarrow$ evalInformativeness($A$)
22             end if
23             $sturdiness \leftarrow$ computeSturdiness($A$)
24             $counter += 1$
25         end for
26     end while
27 end if
28 return $A$
```

coreferent mention with other mentions in the same cluster (Line 1). For example, in the asbestos example, the *“building”* and *“it”* are coreferent mentions, but *“building”* is more specific, so the mention *“it”* is accompanied by the mention *“building”* in square brackets. This way, the coreference are explicitly surfaced in D. In the next step, the Manager generates a base argument (Line 2) including all associated **supports** relations from the Argument Interpretation component, **extends** relations from the Informativeness Evaluation component and **follows** relations from the Connectedness Evaluation Component. The argument A generated in this step may be missing a number of links. This will be what the rest of the algorithm will work to update and enrich. Next, we compute the sturdiness of the base argument (Line 3) which serves as a starting point for evaluating the argument. The sturdiness computation returns a score and also a list of edges that are not maximally sturdy (contain missing edge types).

The algorithm then proceeds to iterate through each of these non-sturdy pairs and (1) generate warrants (Lines 8-16), (2) update the argument (Lines 18-19), (3) evaluate overall connectedness and informativeness (Lines 20-21) and (4) recompute sturdiness of the updated argument (Line 23). The algorithm repeats this exercise until either a threshold sturdiness or iterations are reached. It is worth highlighting here that once a candidate warrant has been generated (Line 11), we check if the combination of the premise and the generated warrant is entails (follows to) the conclusion (Line 12). The reason for this is that we want to ensure that the warrant does not unintentionally contradict the premise-conclusion pair. Similarly, we want to make sure that the candidate warrant is not just a restatement of the premise but actually informative over the premise (Line 13). A candidate warrant is selected only when both these criteria are satisfied. In a sense, we adopt generate-and-test paradigm with SKEPTIC using GPT-3 as a warrant-generator and other evaluator components as a warrant-tester. If necessary, we repeat this process until a suitable warrant is found or until a pre-specified number of tries have been completed.

3. Discussion

In this paper, we have proposed a system to evaluate arguments, but we have not discussed constructing arguments. The process of evaluating another's argument and constructing one's own argument does share some structural similarities including potentially the fact that we might intuitively evaluate our own arguments. However, we believe that argument construction potentially involves additional cognitive machinery that might at least require a level of social reasoning and Theory of Mind to consider whether the listener will believe the argument. Presumably, when constructing arguments, the premises we select and their arrangement in an argument might be heavily influenced by what we believe the listener believes. This paper, however, does not discuss these additional cognitive capabilities, and instead focuses on the evaluation of arguments.

Even within boundaries of argument evaluation, we are focused on *content* and not as much the *source*. Our proposed approach is one of rationalization and extracting warrants and tying the argument parts together to produce an argument interpretation and evaluation that is stronger (and not weaker) than what was in the original input. This assumes that we trust the source, or at least believe the conclusion. A rationalization to find the necessary support and warrants to strengthen a conclusion (this system) is very different from one in which a critical outlook is taken to depose a conclusion. SKEPTIC could be reconfigured to find those warrants that given the same premises contradict the conclusion. We have not tested this capability and as such not within the scope of this paper. With regard to trust, we do not consider the speaker or our belief in the trustworthiness of the speaker. A more complete account of human argumentation must involve a discussion about how we evaluate trust and how that trust evaluation factors into our process of rationalizing or criticizing a conclusion. There are additional dimensions to making a successful argument including emotional aspects, which although potentially useful, are beyond the scope of this paper.

Sometimes, rationalization involves not only extracting warrants (the bridges connecting premises and conclusions) but also drawing associations before any connections can be made. When we hear a premise or a conclusion, associative memories are triggered to generate relevant associations. There may be valuable connections between the premises and conclusions indirectly via these associations. We expect to be able to use the same underlying architecture to explore such implications. The associative memory retrieval system can be guided with prompts to generate implications (rather than warrants). The completions generated by these operations can be included within the expanded premise node or conclusion nodes. One open question is when should these associations be triggered and how do they interact with other associative tasks like warrant generation? Are associations generated regardless of the sturdiness of arguments? Are they generated before or after warrants are generated? We expect to explore these questions in future work.

Argument rationalization as proposed in this paper begins with the assumption that the evaluator of the argument is interested in exercising some “epistemic vigilance” over the argument (Sperber et al., 2010). As such, the intuitive argumentation demonstrated here serves as a starting point for developing further arguments or initiating a dialectic exchange about it. For example, SKEPTIC can pinpoint exactly where an argument is sturdy and where it might be weaker. If integrated into a more general cognitive architecture, it can request the speaker for elaboration on specifically why they think a premise supports an argument. If SKEPTIC identifies an implied warrant, it can clarify with the speaker whether the identified assumption aligns with what the speaker had in mind. In this way, the argument structures generated by SKEPTIC can be used within a larger cognitive system for conscious reasoning, dialog and interaction.

4. Conclusion

This paper describes a computational framework for intuitive argumentation, called SKEPTIC. Given a body of unstructured text, SKEPTIC can extract rich structured representations (graphs) of the underlying argument. In doing so, it selects argument-relevant spans of text (ADUs), and finds argument-relevant relations (supports, attacks, extends, follows) between pairs of ADUs (premise-conclusion pairs). SKEPTIC also discovers assumptions made in the argument but not present in the original text. The assumptions represent implicit knowledge that enable conclusions to inferentially follow from the premises. SKEPTIC is able to accomplish this with a novel cognitive architecture composed of several components that can perform intuitive processing with the aid of state of the art neural foundation models. SKEPTIC is not an end-to-end neural net trained on a task-specific dataset. Instead, it is a cognitive system with an inspectable meta algorithm that directs the processing of various neural components to generate symbolic and interpretable representations. SKEPTIC can be integrated into more general cognitive architectures and used in connection with applications in which it can assist with our critical thinking capabilities.

References

Bhargava, P., Drozd, A., & Rogers, A. (2021). Generalization in NLI: Ways (not) to go beyond simple heuristics.

Bommasani, R., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brown, T., et al. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* (pp. 1877–1901). Curran Associates, Inc.

Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*, 321–357.

Evans, J. S. B. (2003). In two minds: dual-process accounts of reasoning. *Trends in cognitive sciences*, *7*, 454–459.

Friedman, S., Magnusson, I., Sarathy, V., & Schmer-Galunder, S. (2021). From Unstructured Text to Causal Knowledge Graphs: A Transformer-Based Approach. *Proceedings of the Ninth Annual*

Conference on Advances in Cognitive Systems (p. 18).

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Hudson, R. P. (2022). Coreferee. `https://github.com/explosion/coreferee`.

Ilievski, F., Szekely, P., & Zhang, B. (2021). Cskg: The commonsense knowledge graph. *European Semantic Web Conference* (pp. 680–696). Springer.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. Technical Report arXiv:2107.13586, arXiv. ArXiv:2107.13586 [cs] type: article.

Liu, Y., et al. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mercier, H. (2011). On the Universality of Argumentative Reasoning. *Journal of Cognition and Culture*, *11*, 85–113.

Mercier, H., & Sperber, D. (2011). Why do humans reason? Arguments for an argumentative theory. *Behavioral and Brain Sciences*, *34*, 57–74.

Peldszus, A. (2017). *Automatic recognition of argumentation structure in short monological texts*. Doctoral dissertation, UniversitÃ¤t Potsdam.

Peldszus, A., & Stede, M. (2013). From argument diagrams to argumentation mining in texts: a survey. (p. 31).

Reynolds, L., & McDonell, K. (2021). *Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm*. Technical Report arXiv:2102.07350, arXiv. ArXiv:2102.07350 [cs] type: article.

Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, *33*, 16857–16867.

Sperber, D., Clément, F., Heintz, C., Mascaro, O., Mercier, H., Origgi, G., & Wilson, D. (2010). Epistemic Vigilance. *Mind & Language*, *25*, 359–393.

Stab, C., & Gurevych, I. (2016). *Parsing Argumentation Structures in Persuasive Essays*. Technical Report arXiv:1604.07370, arXiv. ArXiv:1604.07370 [cs] version: 2 type: article.

Toulmin, S. E. (2003). *The uses of argument*. Cambridge university press.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). *Chain of Thought Prompting Elicits Reasoning in Large Language Models*. Technical Report arXiv:2201.11903, arXiv. ArXiv:2201.11903 [cs] type: article.

Williams, A., Nangia, N., & Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 1112–1122). Association for Computational Linguistics.