Robust and Explainable Identification of Logical Fallacies in Natural Language Arguments

Zhivar Sourati[a,b,*], Vishnu Priya Prasanna Venkatesh[a,b], Darshan Deshpande[a,b], Himanshu Rawlani[a,b], Filip Ilievski[a,b,*], Hông-Ân Sandlin[c] and Alain Mermoud[c]

[a] *Information Sciences Institute, University of Southern California, Marina del Rey, CA, USA*
[b] *Department of Computer Science, University of Southern California, Los Angeles, CA, USA*
[c] *Cyber-Defence Campus, armasuisse Science and Technology, Switzerland*

ARTICLE INFO

Keywords:
logical fallacy
explainability
case-based reasoning
knowledge injection
data augmentation
robustness

ABSTRACT

The spread of misinformation, propaganda, and flawed argumentation has been amplified in the Internet era. Given the volume of data and the subtlety of identifying violations of argumentation norms, supporting information analytics tasks, like content moderation, with trustworthy methods that can identify logical fallacies is essential. In this paper, we formalize prior theoretical work on logical fallacies into a comprehensive three-stage evaluation framework of detection, coarse-grained, and fine-grained classification. We adapt existing evaluation datasets for each stage of the evaluation. We employ three families of robust and explainable methods based on prototype reasoning, instance-based reasoning, and knowledge injection. The methods combine language models with background knowledge and explainable mechanisms. Moreover, we address data sparsity with strategies for data augmentation and curriculum learning. Our three-stage framework natively consolidates prior datasets and methods from existing tasks, like propaganda detection, serving as an overarching evaluation testbed. We extensively evaluate these methods on our datasets, focusing on their robustness and explainability. Our results provide insight into the strengths and weaknesses of the methods on different components and fallacy classes, indicating that fallacy identification is a challenging task that may require specialized forms of reasoning to capture various classes. We share our open-source code and data on GitHub to support further work on logical fallacy identification.

1. Introduction

The purpose of constructing an argument is to prove conclusions that are in some way unknown or doubtful or that have been challenged and called into question [8]. A *logical fallacy* is a logical mistake in the reasoning used to transition from one proposition to the next, which results in a faulty argument [3]. Logical fallacies form a broad category of violations of argumentation norms, including structure, consistency, clarity, order, relevance, and completeness. Detecting whether an argument is fallacious and the corresponding actual violation, is in practice a subtle task. Detecting one or more fallacies in an argument, however, does not prove its conclusion to be false - they merely detect a flaw in the reasoning that attempted to prove that the conclusion is true.

Logical fallacies have been of interest to social science since the early days of mathematics and philosophy [4]. More recently, the societal relevance of logical fallacies has been greatly amplified due to the wide adoption of the World Wide Web, which enabled a free exchange of large amounts of information, including an easy spread of misinformation [121, 125, 2] and propaganda [29, 10, 49]. Misinformation and propaganda are thorny issues for social media platforms on the Web and have been increasingly addressed through the growing teams of moderators [41, 86], and are under the scrutiny of different organizations and governmental bodies, such as the UN [64]. Similarly, the EU plans to ratify addressing misinformation as part of the Digital Services Act [27], as the spread of harmful and incorrect arguments can sway the population and lead to political shifts and civil unrests [67].

Considering the *subtlety* and the *volume* of fallacious arguments, manually checking each by a human has become impossible. Moreover, the very *subjective* nature of the tasks tends to open room for disagreement on the classification when multiple annotators or moderators are involved. This motivates the need for automated methods that can quickly process an argument, understand its intent, and detect possible flaws in the reasoning. The algorithms need to be *robust*, i.e., work well for an argument in an open domain, and *explainable*, i.e., provide an explicit trace of their reasoning for human collaborators like social media moderators. Prior work on taxonomizing logical fallacies [26, 4, 8] and the initial efforts to develop logical fallacy benchmarks [62] has set the ground for comprehensive and trustworthy logical fallacy methods. However, as these works have been attempted in isolation, comprehensive methods and tasks are lacking.

Building a comprehensive evaluation setup and methods for logical fallacy identification has several key challenges. First, while prior work has provided a list of taxonomies for organizing logical fallacies, it is unclear how they can be organized and aligned with existing benchmarks. Second,

[*] Corresponding author

souratih@isi.edu (Z. Sourati); vprasann@isi.edu (V.P.P. Venkatesh); darshang@isi.edu (D. Deshpande); hrawlani@isi.edu (H. Rawlani); ilievski@isi.edu (F. Ilievski); hongan.sandlin@ar.admin.ch (H. Sandlin); alain.mermoud@ar.admin.ch (A. Mermoud)
ORCID(s):

logical fallacies require an abstraction from syntax to high-level semantics revolving around structure and soft logic. This makes pure language model-based methods insufficient for fully solving the task. Third, arguments rely heavily on background factual and commonsense knowledge. A robust and explainable method needs mechanisms to make implicit (assumed) knowledge in fallacies explicit. Fourth, given the large set of fallacies and the relatively small amount of annotated examples for supervised learning, data sparsity is a serious issue. To build robust and explainable methods, it is essential to devise scalable mechanisms that can combat data sparsity.

In this paper, we consider the research question: *How can we build methods for robust and explainable identification of logical fallacies in natural language arguments?* We consolidate prior work on taxonomizing logical fallacies into a three-stage framework of logical fallacy identification tasks, ranging from deciding whether there is a logical fallacy in an argument (logical fallacy detection), performing classification in high-level classes (coarse-grained classification), and finally performing classification into a wider range of specific classes (fine-grained classification). To deal with the need for abstraction and to fill knowledge gaps, we experiment with three families of methods: prototype-based reasoning, instance-based reasoning, and knowledge injection. We combat data sparsity through suitable methods for data augmentation and curriculum learning.

The contributions of this paper are as follows:

1. We design a three-stage framework of logical fallacy identification tasks, inspired by fallacy classification theories. We map and enhance existing datasets into this pipeline to provide a well-motivated and representative evaluation set.

2. Our framework includes a wide range of methods with a focus on robustness and explainability: prototype-based reasoning, instance-based reasoning, and knowledge injection. We complement these methods with strategies for distant learning from more data based on data augmentation and curriculum learning.

3. We conduct an extensive evaluation of these methods on our datasets, focusing on their robustness and explainability. Our results provide insight into the strengths and weaknesses of the methods on different components and fallacy classes, indicating that fallacy identification is a challenging task that may require specialized forms of reasoning to capture various classes.

The rest of this paper is structured as follows. A comprehensive study of different classification schemas on logical fallacies, together with our three-stage framework, is presented in Section 2. Prior work that detects logical fallacies or uses related methods to ours is reviewed in Section 3. We describe the adopted methods in Section 4 and the experimental setup in Section 5. Our results accompanied by the extra ablation studies are presented in Section 6. We discuss our findings and conclude the paper in Sections 7 and 8.

We make all our code and data available on GitHub at https://github.com/usc-isi-i2/logical-fallacy-identification.

2. Organizing Logical Fallacies

There are two broad categories of fallacies: *formal*, involving the error in the logical structure of the argument, and *informal*, mostly concerned with the content of the argument or the latent error in their expression of logic [45]. In this study, we focus on the latter. Within informal fallacies, various definitions and categorizations of logical fallacies have been proposed since antique Greek philosophers such as Aristotle [4]. Aristotle's Sophistical Refutations [4] and John Locke's An Essay Concerning Human Understanding [73] can be considered as the cornerstones of works on logical fallacies, followed by notable contributions by others, especially Copi [26], Barker [8], and Watts [122]. We elaborate on each of the aforementioned philosophical theories in Section 2.1. Then, in Section 2.2, we devise our logical fallacy framework that is rooted in these philosophical theories, and it formalizes them into three stages: fallacy detection, coarse-grained classification, and fine-grained classification. We describe the coarse- and the fine-grained classes that constitute our taxonomy of logical fallacies.

2.1. Existing Theories of Categorization for Logical Fallacies

Aristotle [4] distinguishes several kinds of deductions (syllogisms) in [4]. Broadly, he groups the fallacies into the ones dependent on language (*In Dictione*) and the ones not dependent on language (*Extra Dictionem*). His categorization revolves around the premises discussed in the deductions as well as the conditions required for arguments to prove them correct. According to Aristotle, an argument satisfies three conditions, and "is based on certain statements made in such a way as necessarily to cause the assertion of things other than those statements and as a result of those statements." Thus an argument may fail to be a syllogism in three different ways: (1) the premises may fail to necessitate the conclusion, (2) the conclusion may be the same as one of the premises, and (3) the conclusion may not be caused by (or grounded in) the premises. Aristotle's fallacies are primarily fallacious deductions that appear to be correct on the surface. There are six classes of fallacies dependent on language: *Equivocation*, *Amphiboly*, *Combination of Words*, *Division of Words*, *Accent*, and *Form of Expression*. Additionally, there are seven kinds of logical fallacies (*sophistical refutation* in Aristotle's words) that can occur in the category of fallacies not dependent on language: *Accident*, *Secundum Quid*, *Consequent*, *Non-Cause*, *Begging the Question*, *Ignoratio Elenchi* and *Many Questions*. In summary, Aristotle classifies fallacies into thirteen classes.

Barker [8] classifies logical fallacies based on the validity of the assumptions made when transitioning from premises to conclusions, as well as the validity of the premise and the conclusion themselves. Barker defines validity as follows. First, a valid argument would comprise premises that are all true. Second, it would not need the conclusions to satisfy their validity. And finally, its conclusions can be directly derived from the premises. This view is closely similar to Aristotle's, as well as the requirements that [88] have analogously proposed. Neglect of the third requirement gives rise to the fallacies of *Non Sequitur* that are fallacies that have an insufficient link between premises and conclusions. Neglect of the second requirement gives rise to fallacies of *Petitio Principii* in which "the premises are related to the conclusion in such an intimate way that the speaker and his hearers could not have less reason to doubt the premises than they have to doubt the conclusion". Neglect of the first requirement gives rise to the remaining category of fallacies in which premises are present that are not necessarily true all at once, even if the link between premises and conclusions is as rigorous as can be. In summary, identifying fallacious arguments would boil down to analyzing the validity and soundness of the claims as well as the sufficiency and necessity of the premise of arguments to satisfy the needs of the conclusion to be true. There are three levels of classification proposed in [8] that, on the finest level, would sum up to twenty classes of fallacies, although his categorization allows for more as well and he does not argue for a bounded definition or particular number.

Locke [73] can be credited with the contribution of *Ad-Arguments*, which are arguments "that men, in their reasoning with others, do ordinarily make use of to prevail on their assent; or at least so to awe them as to silence their opposition." Locke discusses three kinds of such arguments: *Ad Verecundiam*, *Ad Ignorantiam*, and *Ad Hominem*. According to him, these are not fallacies, but have been developed beyond his conception and have been named as such [48]. *Ad Verecundiam*, or *Appeal to Authority* is a fallacy when it is either on the ground that authorities (experts) are fallible or for the reason that appealing to authority is an abandonment of an individual's epistemic responsibility [53]. *Ad Ignorantiam*, or *Appeal to Ignorance*, happens when one demands "the adversary to admit what they allege as a proof, or to assign a better." In other words, the *Ad Ignorantiam* fallacy happens when the argument claims a proposition to be true because there is no evidence against it. According to Locke, *Ad Hominem* was a way "to press a man with consequences drawn from his own principles or concessions." That is, to argue that an opponent's view is inconsistent, logically or pragmatically, with other things he has said or to which he is committed to [53].

Copi [26] defines fallacies as "a form of argument that seems to be correct but which proves, upon examination, not to be so." Copi discusses both deductive invalidities and inductive weaknesses as sufficient reasons for arguments to be fallacious. From the eighteen *informal fallacies* he categorizes, eleven are borrowed from [4] and the other seven can be traced back to [73]. He breaks down fallacies into *formal fallacies* and *informal fallacies*. With his definition over *formal fallacies* pertaining to the deductive fallacies, he classifies *Affirming the Consequent*, *Denying the Antecedent*, *The Fallacy of Four Terms*, *Undistributed Middle*, and *Illicit Major* as *formal fallacies*. Focusing on the *informal fallacies*, Copi defines two broad categories as *Fallacies of Relevance* and *Fallacies of Ambiguity*. *Fallacies of Relevance* include *Accident*, *Converse Accident*, *False Cause*, *Petitio Principii*, *Complex Question*, *Ignoratio Elenchi*, *Ad Baculum*, *Ad Hominem Abusive*, *Ad Hominem Circumstantial*, *Ad Ignorantiam*, *Ad Misericordiam*, *Ad Populum*, and *Ad Verecundiam*, while *Fallacies of Ambiguity* include *Equivocation*, *Amphiboly*, *Accent*, *Composition* and *Division*.

We conclude that the described categorizations [4, 26, 8, 73] mostly agree on the definition of fallacious arguments as well as the broad categorizations of fallacies. The main difference lies in the fine-grained categorizations: Aristotle [4] discusses the thirteen ways arguments can be fallacious, while Copi [26] proposes eighteen different fallacy groups. Barker [8] categorizes fallacies into twenty classes although he does not delineate the exact categorization or the number of classes, and all presumably borrow *Ad Fallacies* from Locke [73]. These discrepancies require computational approaches for logical fallacy identification to choose between the proposed theories. For our experimental work, we adopt the broad categorization of [26], and the fine-grained classification by [54] and [62]. We describe our categorization further in Section 2.2.

2.2. Logical Fallacy Framework

We design a three-stage framework (Figure 1) as an overarching testbed for prior research on logical fallacies. The first stage of the *logical fallacy detection* aims to identify whether a logical statement contains a logical fallacy or not. The detection is formalized as a binary classification task to identify the arguments that are logically fallacious in any sense. If a fallacy has been detected, the goal of the second stage is to categorize the fallacy into one of a few broad classes (e.g., *Fallacy of Relevance*). In the third stage, the aim is to further classify a fallacy into a fine-grained class (e.g., *Ad Populum*).

Following [26], we consider the following four coarse-grained classes: *Fallacy of Relevance*, *Fallacy of Defective Induction*, *Fallacy of Presumption*, and *Fallacy of Ambiguity*. Figure 1 shows the sub-categorizations we make from these coarse-grained classes to fine-grained classes described in [62]. To perform the mapping, we use the definitions of fine- and coarse-grained classes given in [26]. We next describe our fallacies in detail.

Fallacy of Relevance occurs for arguments with premises that are logically irrelevant to the conclusion. *Fallacy of Relevance* subsumes the fine-grained classes *Ad Hominem, Ad Populum, Appeal to Emotion, Fallacy of Extension, Intentional Fallacy*. All of these fallacy classes present different means for using peripheral premises as support for claims. *Ad Hominem* contains sentences where an attack over the subject acts as a premise for the claim made in those sentences, while *Appeal to Emotion* involves manipulating the recipient's emotions to prove a claim. *Ad Populum* involves

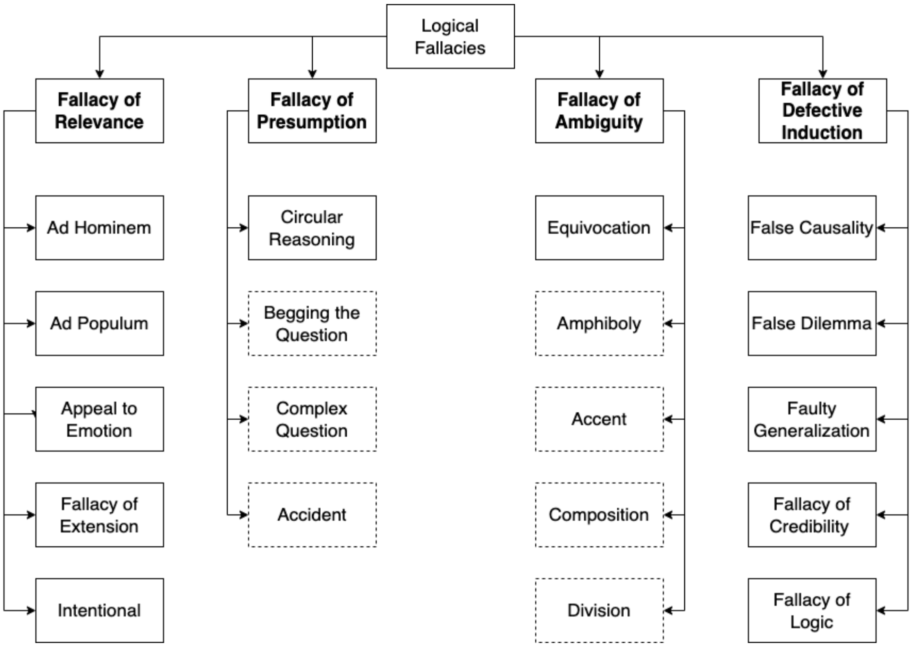


Figure 1: Three-stage taxonomy of logical fallacy identification. Coarse-grained classes are shown in boldface, while regular font is used to show fine-grained classes. We use solid and dotted boundaries to distinguish between fine-grained classes that we include and exclude in our experimental study, respectively.

Table 1
Examples for fallacious arguments belonging to different coarse-grained and fine-grained classes covered in our work.

Coarse-Grained Class	Fine-Grained Class	Example
Fallacy of Relevance	*Ad Hominem*	Boris is not qualified to make suggestions about our penal system. As an ex-convict, he would always take the criminals' side.
	Ad Populum	Aliens must exist because most people believe in them.
	Appeal to Emotion	Luke didn't want to eat his vegetables, but his father told him to think about the poor, starving children in a third world country who don't have anything to eat.
	Fallacy of Extension	If you don't drive a car, you hate the Earth.
	Fallacy of Relevance	I know you want to imprison me for having murdered my parents, but judge, have mercy on me, I'm an orphan!
	Intentional	A woman decides to visit a certain doctor after only asking advice on the best doctors from ONE friend.
Fallacy of Defective Induction	*False Causality*	The temperature has dropped this morning, and I also have a headache. The cold weather must be causing my headache.
	False Dilemma	Subscribe to our streaming services, or get stuck with cable!
	Faulty Generalization	My friend said her Math class was hard, and the one I'm in is hard, too. All Math classes must be hard!
	Fallacy of Credibility	My uncle is a mechanic and he says you shouldn't spank children. He says it's ineffective.
	Fallacy of Logic	Employees are like nails. Just as nails must be hit in the head in order to make them work, so must employees.
Fallacy of Presumption	*Circular Reasoning*	Quinoa is a delicious, plant-based source of protein because it tastes so darn good.
Fallacy of Ambiguity	*Equivocation*	The officer told me to freeze but it was too hot out to be freezing, so I was justified in running away.

affirming claims based on popular belief, and *Fallacy of Extension* uses exaggeration for affirming claims based on the corresponding sentences. *Intentional Fallacy* is directed towards using subconscious choices to incorrectly support an argument.

Within the broad class of ***Fallacy of Defective Induction***, the premises seemingly provide ground for the conclusion but upon analysis prove to be insufficient and weak for supporting the claim made. *Fallacy of Defective Induction* is specified via five fine-grained categories, namely *False Causality, False Dilemma, Faulty Generalization, Fallacy of Logic, and Fallacy of Credibility.* Arguments that jump to a conclusion without implying a causal relationship between the premise and the claim fall under *False Causality*. If the specific causal relationship between the premise and the claim is generalized to a wider category of subjects, the argument is categorized as *Faulty Generalization*. Arguments that cast doubt regarding the credibility of the subject making a claim constitute for *Fallacy of Credibility*. When an argument presents a premise that erroneously limits the options available, it constitutes a *False Dilemma*. When the logical construct of the argument is inaccurate and misleading, it constitutes a *Fallacy of Logic*.

Fallacy of Presumption takes place when the inference to the conclusion depends mistakenly on unwarranted assumptions. *Fallacy of Presumption* includes the following fine-grained classes. *Circular Reasoning* occurs for arguments that come back to the beginning without proving themselves. Other classes that fall within *Fallacy of Presumption* are: *Begging the Question*, where the conclusion is treated like an assumption from the premise of the statement; *Complex Question*, where the argument is framed as a loaded question that intends to prove another latent unproved assumption; and *Accident*, where generalization is applied to specific cases that are out of scope.

Fallacy of Ambiguity occurs when words or phrases are used in an equivocal way, thus causing ambiguity in the logic that connects the premise and the conclusion. The fallacy class *Equivocation* is a *Fallacy of Ambiguity* due to the presence of phrases in arguments that are used interchangeably in different parts of the sentence, leading to ambiguity in logic. Other classes in *Fallacy of Ambiguity* include *Amphiboly, Accent, Composition, and Division*. In the case of *Amphiboly*, the usage of words that could be used interchangeably leads to a false interpretation in the grammatical construction of the sentences. *Accent* fallacy is one, where a specific phrase or word carries a different contextual meaning in the premise and the conclusion. Mistaken inferences about parts of a whole argument for drawing inferences about attributes for that argument constitute the *Composition* fallacy. *Division* fallacy is the reverse of the *Composition* fallacy, where mistaken inferences about the whole argument are used for drawing inferences about attributes of parts of it.

We provide examples for each of the fine-grained and coarse-grained classes in Table 1. A simplifying assumption we make in this work is that each fallacious argument belongs to exactly one broad class and exactly one fine-grained class. Prior work [29, 62] has shown that this assumption does not always hold, for example, *"Drivers in Richmond are terrible. Why does everyone in a big city drive like that?"* as cited in [62], is an example that belongs to *Ad Hominem* but does have flavors of *Faulty Generalization* as well. This gives room for arguments to be categorized into different fallacy classes simultaneously. Our simplifying assumption restricts our classification task to a multi-class task rather than a multi-label task.

3. Related Work

In this section, we review prior computational work on logical fallacy detection and the related task of propaganda detection. We also review related work that leverages the methods of case-based reasoning, knowledge injection, and curriculum learning.

Logical Fallacy. Prior computational work formalizes arguments containing logical fallacies to make them suitable for ingestion by rule-based systems and theoretical frameworks. Gibson et al. [46] formalize and identifies *formal logical fallacies* using Argument Markup Language (AML) and discusses the theoretical questions that arise in the study of fallacy. Yaskorska et al. [126] adopt a structure-aware approach to identify, include, and eliminate *formal fallacies* in natural dialogues. Nakpih and Santini [88] present a model that discovers *non sequitur fallacies* in legal argumentation using Prolog language and check the validity, soundness, sufficiency, and necessity of argumentation using logical rules. These works mostly focus on *formal fallacies*, which are defined in terms of their structure. In our work, we focus on *informal fallacies*, whose detection and classification rely on linguistic and world knowledge.

One of the few studies done on *informal fallacies* [62] proposes the task of logical fallacy detection, where arguments are classified into thirteen fine-grained fallacies. This work evaluates the effect of using large pretrained language models on two datasets, called LOGIC and LOGIC Climate. Apart from using large pretrained language models, Jin et al. [62] try to abstract away from the surface of the arguments by exploiting coreference resolution and entity linking, in order to identify logical fallacies that are structurally fallacious in their arguments. Similarly, Goffredo et al. [47] alongside presenting an annotated dataset of 31 political debates from the U.S. Presidential Campaigns, use transformer-based language models and process four parts of arguments, i.e., the dialogue context, argument components (premise and claim), fallacious argument snippet, argument relation (attack or support) separately, classify them, and train all the models jointly. They show that detecting argument components, relations, and context (see also [107]) in debates is a necessary step to improve the model's performance. The main difference between [47] and our study is the fact that we do not need and use any context to classify logical fallacies. Furthermore, our framework does not assume any specific structure for text, and hence can be more generalizable. In our work, we reuse the dataset from

[62], and also extend its evaluation framework by: (1) introducing a binary detection and coarse classification stage, (2) employing methods with robust properties to satisfy the needs of classification of logical fallacies that go beyond language understanding brought by vanilla language models, (3) adapting our methods with native explainability, and (4) carrying out a more extensive set of experiments and analyses.

Propaganda Detection. Recent research has developed benchmarks and techniques for propaganda detection in natural language documents. A significant portion of these works focuses on extracting better features as well as novel methods that would help the model boost its performance [91, 50, 94, 65, 117]. There has also been a surge focusing on the interpretability of models in propaganda detection [128, 127, 39]. Dimitrov et al. [34] show that propaganda techniques function as shortcuts in the argumentation process that connect to the emotions of the audience and often include logical fallacies. In [51], logical fallacies are called "hallmarks of propagandist messaging", which implies that logical fallacies can be seen as components within the broader task of propaganda detection. However, as pointed out by Jin et al. [62], the two tasks overlap but are distinct, since propaganda detection focuses on arguments that aim to influence people's opinions often using misinformation as a tool [74, 61], while logical fallacy detection aims to understand gaps in argumentation. There is also a practical difference between the formalization of these two tasks, as propaganda detection data has typically focused on longer input documents, while logical fallacy datasets have generally relied on focused and isolated text inputs. In our study, we utilize the overlap between some of the propaganda techniques and fallacy classes, by augmenting the training data for logical fallacy classification with a dataset gathered explicitly around propaganda detection [81].

Case-Based Reasoning. The case-based reasoning framework has been used to learn from past experiences explicitly in medical applications [92, 93] and mechanical engineering [7, 96]. One of the most important aspects of case-based reasoning is its inherent interpretability. Walia et al. [118] use case-based reasoning as an interpretation model for Word Sense Disambiguation, while Brüninghaus and Ashley [17] apply case-based reasoning to predict the outcome of legal cases. Ford et al. [40], Ge et al. [43], Han et al. [52] advocate for the increase in comprehension of the black-box models and their explainability as well as transparency using example-based explanations by the end-users. Similar to our work, Spensberger et al. [114] explore the effect of case-based reasoning on the student social workers and their fallacy recognition abilities and find that those who have access to worked examples perform better during the experiment. In this paper, we adopt two complementary case-based reasoning methods. First, we adopt the instance-based reasoning method proposed by [112] that enriches the inputs with similar cases and with different case enrichments (e.g., based on counterarguments), and evaluates the impact of different modeling decisions and case representations on the model performance. We apply this method to our three-stage evaluation framework and perform further ablation studies to understand its performance in relation to modeling decisions and against other systems. Second, we include a prototype-based reasoning method, that maps novel examples to prototypical ones to classify logical fallacies. With both of these methods, we use case-based reasoning both as a means to enhance the performance of our model and simultaneously as a proxy to explain the behavior of the model classifying logical fallacies.

Knowledge Injection. The challenge of generalizability and transferability for logical fallacy classifiers has been discussed in [62], by testing the model on a dataset containing unseen domain-specific subjects. This motivates the need for the injection of background knowledge. Injection of background, especially commonsense knowledge in language models has been proposed within tasks of multiple-choice question answering. Combining neural language models with commonsense knowledge graphs (KGs) like ConceptNet [113] or ATOMIC [108] can be done by lexicalizing knowledge into task-targetted evidence paths and combining them with the task input [76, 84]. The idea in K-BERT [71] is similar - here a multi-head attention layer is used to combine evidence from background knowledge and the input task. Other forms of knowledge injection have been popular as well, such as using graph and relation networks [70, 130], or introducing the entire KG at training time regardless of the task at hand [95, 77]. Notably, prior work has shown that the impact of the injected knowledge strongly depends on the overlap between the knowledge in these graphs and the downstream question answering task [77, 59]. Due to the nature of logical fallacies, they can cover daily-life matters and events spreading throughout social media, and this calls for domain-specific knowledge for comprehension of certain logical fallacies. However, to our knowledge, exploiting external knowledge has not yet been fully explored in logical fallacy detection. Trying to fill in the gap and utilize commonsense knowledge in the detection of logical fallacies, we use [71] to incorporate knowledge from arbitrary knowledge bases and benefit from potential enhancements.

Curriculum Learning. Curriculum learning has been proposed in [12] and [38] from the computer science and psychology perspectives respectively. The key idea of curriculum learning is that starting from simple examples and learning from examples in an organized and meaningful way can contribute positively to the learning process. Using pure language model-based methods does not suffice for a reliable classification of logical fallacies [62], due to known issues of robustness and induction capabilities of vanilla language models on unseen data [119, 78]. This motivates us to leverage continual curriculum learning to attempt to improve the convergence and robustness capabilities of models, an idea that has not yet been explored in logical fallacies. The application of curriculum learning to logical fallacies in our work is facilitated by the availability of datasets at different granularity levels.

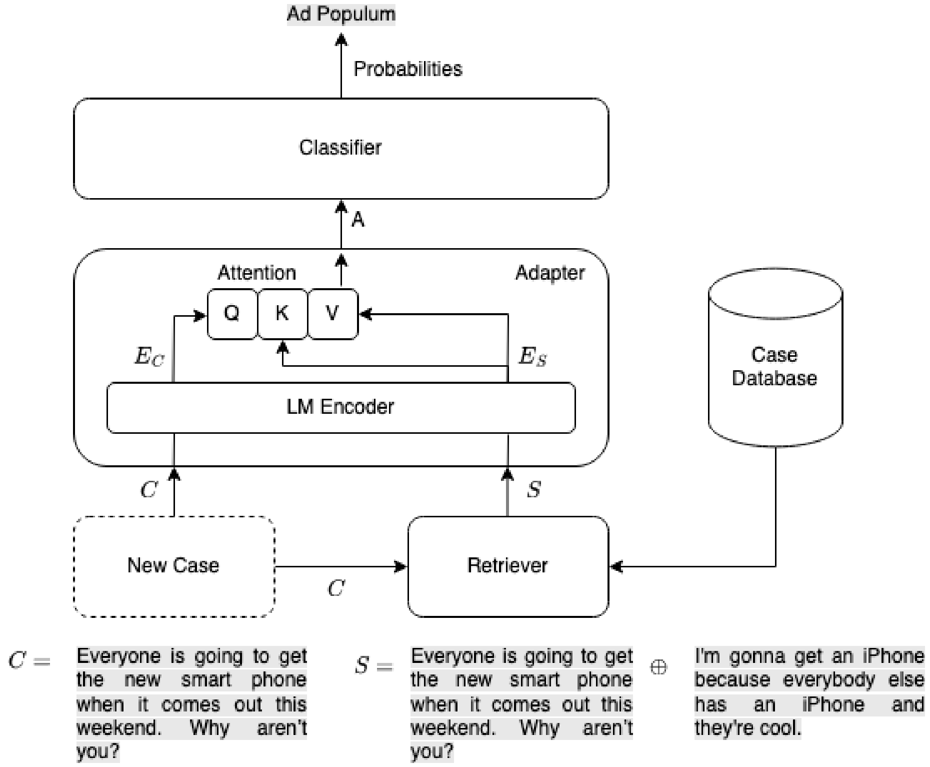

Ad Populum
Probabilities
Classifier
A
Attention
Q
K
V
Adapter
E_C
E_S
LM Encoder
Case Database
C
S
New Case
Retriever
C
$C =$ Everyone is going to get the new smart phone when it comes out this weekend. Why aren't you?
$S =$ Everyone is going to get the new smart phone when it comes out this weekend. Why aren't you?
$\oplus$
I'm gonna get an iPhone because everybody else has an iPhone and they're cool.

Figure 2: Three stages of the IBR pipeline. Using the new Case C, retriever finds k similar examples $\{S_1, S_2, ..., S_k\}$, and creates $S = C \oplus < SEP > S_1 \oplus S_2 \oplus ... \oplus S_k$. The adapter encodes these two inputs and tries to adapt S based on the new case C. Finally, the classifier uses the rectified information from the adapter to classify the new case by outputting the probabilities corresponding to belonging to each class of fallacies (in the example shown above, $k = 1$).

4. Method

Due to the difficulty, as well as the contention over the categorization and classification of logical fallacies [54], we use methods that humans usually adopt when faced with problems that require complex reasoning. According to [97, 14, 105], people use similar or prototypical examples of a situation or problem to solve or approach a new one. The alluded similarity can be in the various levels, namely, coarse-grained features such as the whole argument or statements, but also in the more fine-grained features and in terms of the extra knowledge one might have about concepts or entities discussed in the sentences as discussed by [5]. Having in mind the simplicity as well as explainability of using similar examples or experiences to reason about and solve new problems or situations, we adapt methods for Instance-based Reasoning, Prototype Learning, and Knowledge Injection (§4.1). Another approach that humans follow for learning how to solve problems is starting from easy or simpler tasks and gradually shifting to harder ones to learn [38], which has been shown to work even better than other learning strategies by Chen and S. Savage [20]. This has been shown to be the case for neural networks as well [36], not as a barrier, but as a way of training more robust models referred to as Curriculum learning (§4.2). Finally, we devise data augmentation strategies to address data sparsity and improve the stability of our models [129] (§4.3).

4.1. Explainable Reasoning Methods

4.1.1. Instance-Based Reasoning

Instance-based reasoning (IBR) [30] is the process of solving new problems based on the solutions of similar past problems [80]. IBR is reported to resemble the way humans think and approach new problems to save time and effort instead of starting from scratch [97]. IBR is a formalization of the general idea of Case-based reasoning (CBR) [80]. Within CBR, rather than comparing new problem instances with instances seen before like in IBR, we use past similar problems and experiences and attempt to perform explicit generalization or induction.[1]

IBR starts with a set of cases or training examples; it forms generalizations of these examples, albeit implicit ones, by identifying commonalities between a retrieved case and the target case, and tries to approach the new case using known solutions to past cases. Our IBR formulation (Figure 2) follows the three-stage pipeline proposed by [112] consisting of: (1) *Retriever* - given a target problem, retrieve similar cases with known solutions from memory, (2)

[1] We cover another variant of CBR, prototype theory, in §4.1.2.

Adapter - adapt the retrieved similar cases to help the decision on the new case, and (3) *Classifier* - classify the new case based on the adapted exemplars. The last step in this pipeline corresponds to two steps in the formulation by [1]: classify the new case based on the previous examples, and retain the new problem alongside its adapted solution and resulting experience in memory for later use in a more explicit way. We next describe the design of the retriever, the adapter, and the classifier.

Retriever is responsible for finding similar cases S_i to the new case C from a database and passing them to the adapter together with the new case ($S = C \oplus < SEP > S_1 \oplus S_2 \oplus ... \oplus S_k$ extracting k similar examples). The retriever uses language model encoders to get the feature vectors for each new case as well as all the previous cases in the retriever database and uses these features to compute their cosine similarity.[2] The retriever obtains the k most similar examples from the database, which are then passed on to the adapter module.

We experiment with SimCSE [42], a Transformer-based retriever that is optimized for capturing overall sentence similarity using a contrastive loss function. We also include sentence encoders that are reportedly able to manipulate a wide range of concepts, by using Sentence-BERT [100] based on MiniLM [120]. We also include Transformer models that have been trained to distinguish emotional expressions, since it has been shown that emotions can be used to manipulate masses [13] and they are intuitively important to detect certain logical fallacies, such as *Appeal to Emotion*. To capture the usage of empathetic and emotional terminology, we use a RoBERTa model [72] fine-tuned on the WASSA 2022 Shared Task dataset [9].

Adapter transforms the retrieved cases $\{S_1, S_2, ..., S_k\}$ together with the new case C denoted as S as well as the new case C, and prioritizes earlier cases that are most helpful. The adapter consists of two parts: an encoder and an attention mechanism. As an encoder, we use a language model that takes as input C and S and produces a set of raw hidden states E_C and E_S respectively without a head layer on top.

The attention mechanism selects the most important information to be considered from similar cases. Based on the second step of the pipeline by [1], after the similar cases are retrieved, some of these similar cases should be manipulated or adapted to help the classifier at the end of the pipeline, since not all similar cases will be equally helpful for the model. We formalize this step with an attention mechanism on top of the encoded cases (E_S and E_C) to filter the retrieved cases or shift the attention to where it helps the model best to reason about new cases. More concretely, we use a Multi-headed attention component [116] that fetches the *new case* embedding E_C as the query and the combined embeddings E_S as both keys and values. We include both the new case as well as similar cases in S to avoid losing information from the new case. The output of this component, i.e., the attention output A has the same shape as E_C and E_S and is fed to the last step of IBR, i.e., the classifier.

Classifier layer at the end of the pipeline is applied on top of the adapter output A to predict the labels. As a classifier, we use a two-layer perceptron with a $gelu$ [56] activation function. Given a number of classes C, we compute C logits and their corresponding probabilities of belonging to each class c. We use cross-entropy loss as our learning objective.

Overall, the IBR method is similar to a language model with a classification head on top with an important distinction. By using a retriever and finding similar examples to the new case and integrating these new examples in the classification process, we benefit in two ways: (1) we use similar examples of an argument to help the model classify the argument more accurately, and simultaneously, (2) enhance the explainability of the model, showing the end-users similar examples of an argument to lift end-users' understanding of the capabilities and acquired knowledge of the model [101].

4.1.2. Prototype-Based Reasoning

Prototype theory [105] is a theory of categorization in psychology and cognitive linguistics, in which there is a graded degree of belonging to a conceptual category, and some members are more central than others. In prototype theory, any given concept in any given language has a real-world example that best represents this concept, i.e., its *prototype*. Like IBR, prototype-based reasoning (PBR) is also an instance of case-based reasoning, and there has been some controversy about the superiority of one over the other. There are both claims about the superiority of prototypical examples over normal examples [63], as well as their counterparts [82] who state that a context theory of classification, which derives concepts purely from exemplars works better than a class of theories that included prototype theory (§6.4).

We build on the deep learning adaptation of the prototype theory by the Prototex [31] method. The architecture of Prototex is shown in Figure 3. Prototex is based on the Prototype Classification Network proposed in [69]. The Prototex architecture contains an encoder f and a special prototype layer p, where each unit of that layer stores a weight vector that resembles a prototypical example. The prototype layer includes both positive and negative prototypes, aiming to help the models distinguish between the presence and absence of features that support any given class. The input x is first encoded into a latent representation that is shared between the input data and the prototype layer p. This representation is used to calculate the euclidean distance with the prototype layer p, resulting in a distance vector d. We mask the distance vector with a distance mask layer m. The role of the distance mask m is to make the model only optimize the proximity of input examples of a particular class to a fixed set of prototypes. In other words, the distance mask directs the prototypes to represent prototypical examples of a particular class instead of a mixture of arbitrary classes. The

[2] We also experiment with encoding the input examples as either AMR graphs [6], using explanation graphs [106], or their combination, however, we do not pursue this direction further due to poor performance and explainability.

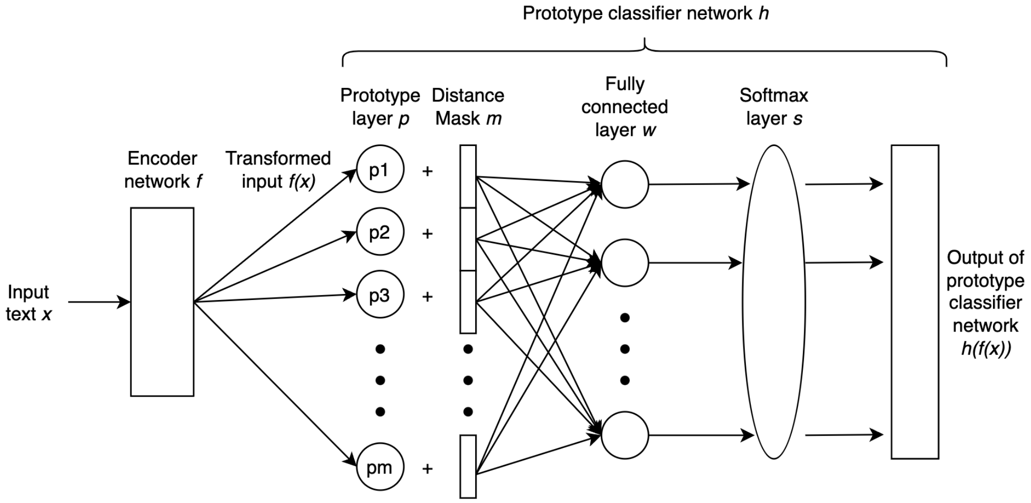


Figure 3: Our PBR method, using an adaptation of the Prototex architecture.

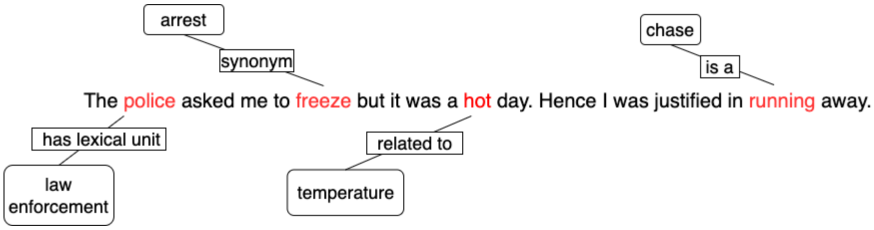

arrest
synonym
chase
is a
The police asked me to freeze but it was a hot day. Hence I was justified in running away.
has lexical unit
law enforcement
related to
temperature

Figure 4: Example Sentence Tree Construction in K-BERT.

masked distance vectors between input examples and prototypes are further fed to a fully connected layer *w* followed by a softmax layer *s* to classify a particular data point. To have interpretable prototype vectors, the model is optimized with auxiliary loss terms that bring the embeddings of the training examples closer to the prototypes and also the embeddings of the prototypes closer to the input examples.

The Prototex method was originally designed for binary classification between propagandistic and non-propagandistic sentences. We modify the Prototex architecture to support a multi-class classification setup. Moreover, the original architecture uses a sequence-to-sequence model, BART [68]. For a fair comparison to our other methods and inspired by the best results on logical fallacy reported in [62], we replace the BART encoder model in Prototex with a self-supervised language model, Electra [23]. We do not use the decoder network and instead focus on the learned prototypes and their explanations.

4.1.3. Knowledge Injection

Many fallacy classes rely on the ambiguous structure of the logical construct in sentences to introduce flaws in arguments. Let us consider the example sentence *The police asked me to freeze, but it was a hot day. So I was justified in running away*, which belongs to the fallacy class *Equivocation* (Figure 4). Here, the word *freeze* is used in two contexts, one for where the *police asked to freeze* and another, where the antonym of *freeze*, i.e, *hot* is used in the sentence. Such sentences, with latent fallacies, illustrate the need for models to have access to commonsense knowledge.

We propose a knowledge injection (KI) formulation, where background commonsense knowledge is combined with the original input for the language model. We adopt a popular method for injecting background knowledge in language models, called K-BERT [71]. K-BERT introduces knowledge injection to a BERT [33] model by querying a structured knowledge base. This knowledge base consists of a set of triples of the form (*subject, predicate, object*). In the first layer, i.e., the knowledge layer, triples from the knowledge base are connected along with the tokens of the sentences, forming a sentence tree, as illustrated in Figure 4. The embedding layer of K-BERT flattens out the sentence tree by retaining the structural information in the form of a visible matrix. As stated in [71], a crucial goal of K-BERT is to prevent false semantic changes to the original sentence due to the addition of sentence trees from the knowledge base. K-BERT functions similarly to BERT [33] but uses a masked self-attention mechanism. The masked self-attention mechanism takes the visible matrix calculated by the seeing layer and ensures that the knowledge branches are not isolated from the tokens they are associated with and do not change the context of the general sentence that they are connected

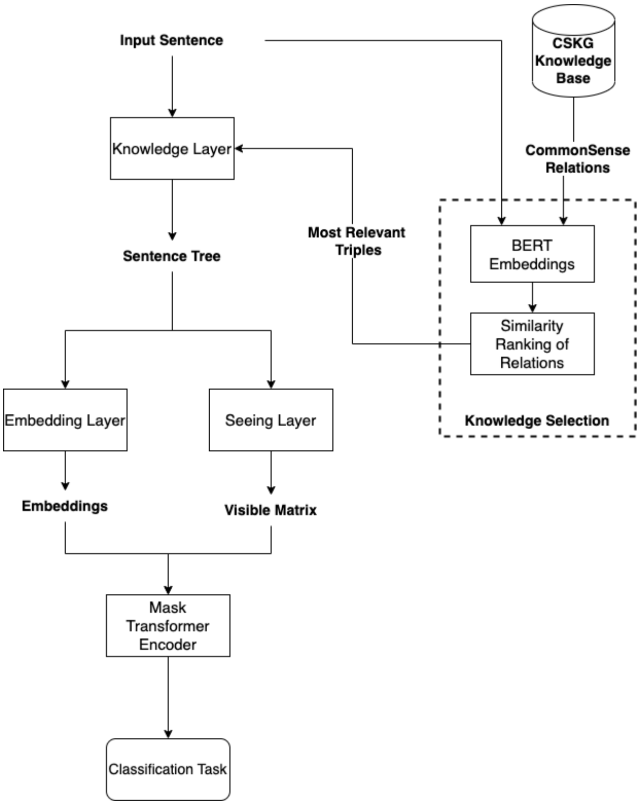


Figure 5: Our Knowledge Injection architecture, which is an adaptation of the K-BERT method.

to. The classification task in K-BERT uses the Masked Language Modeling objective.

Our KI adaptation of the K-BERT method focuses on the input of the knowledge layer, as shown in Figure 5. We adapt K-BERT to leverage knowledge from the Commonsense Knowledge Graph (CSKG) [60], which consolidates commonly used public commonsense sources like ConceptNet [113], ATOMIC [108], and WordNet [83]. The information in CSKG is structured as (*subject, relation, object*) triples. To link to these triples, we extract all non-stopword tokens from the sentences as individual words and we match them with triples in CSKG where the words act as subjects.

Since CSKG contains multiple relations associated with the same subject, a key question is how to prioritize or select relations (triples) that are most relevant and informative for the input sentence. Following [77], we only use the 14 highly semantic relations in CSKG, namely *'Causes', 'UsedFor', 'CapableOf', 'CausesDesire', 'IsA', 'SymbolOf', 'MadeOf', 'LocatedNear', 'Desires', 'AtLocation', 'HasProperty', 'PartOf', 'HasFirstSubevent', 'HasLastSubevent'*. Furthermore, we add a Similarity Ranking component, which ranks the retrieved triples according to their relevance to the original sentence. To do so, we estimate the contextual similarity of the triple to the original sentence by using the cosine similarity of their BERT [116] embeddings as a proxy. The cosine similarity is directly used to order the triples in order of priority. The triples with the highest similarity are injected into the original sentence, thus enriching it with commonsense knowledge. In our experiments with KI, we investigate the impact of the width and depth of the knowledge retrieval procedure. For this purpose, we test different *branching factors* (*b*), representing the maximum number of obtained relations per subject, and different *numbers of hops*, representing the path length between the subject and the subsequent relations discovered iteratively based on the entities of the previously discovered relation.

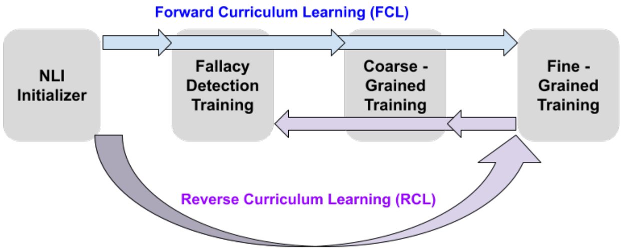

Forward Curriculum Learning (FCL)
NLI Initializer
Fallacy Detection Training
Coarse - Grained Training
Fine - Grained Training
Reverse Curriculum Learning (RCL)

Figure 6: Three-stage curriculum pipelines for Forward Curriculum Learning and Reverse Curriculum Learning.

Returning to our example in Figure 4, we see that the knowledge derived from CSKG helps in providing the context for the word *freeze*, as a *synonym* for arrest. Similarly, background knowledge tells us that the word *hot* is *related to temperature*. On the surface, the words *freeze* and *hot* seem to be used in the same context, but the background information from the knowledge base helps in indicating that they are based on two completely different contexts. The background knowledge for *police* also bolsters that the usage of the word *freeze* was intended for an arrest. This additional knowledge helps in identifying the ambiguous usage of words and connects the terms based on making implicit knowledge explicit. As the additional context (arrest, law enforcement) is not directly connected, we rely on the ability of the BERT LM to estimate the contextual similarity between these terms. Thus, the combination of CSKG and LMs would lead to the classification of the logical fallacy in the sentence as one of *equivocation*.

4.2. Curriculum Learning with Language Models

Curriculum learning (CL) [12] is a strategy that exploits the varying complexity across ordered tasks in a pipeline to increase performance. CL uses previously learned concepts in the task pipeline and applies this information to more complex tasks in the latter half of the pipeline. We follow prior work [103] to formulate two variants of CL (Figure 6). Our Forward Curriculum Learning (FCL) strategy exposes the model to increasingly demanding tasks, similar to how humans learn concepts. We also experiment with the inverse strategy of Reverse Curriculum Learning (RCL), which starts with a difficult task and gradually adapts the model for increasingly easy tasks.

Forward Curriculum Learning (FCL) For FCL, we primarily experiment with continuous training of Transformer language model variants. We try to induce fallacy knowledge in a discrete, three-stage curriculum pipeline, going

Table 2
Augmentation examples.

Original Sentence	**Augmentation Method**	**Augmented Sentence**
Even without watching the movie, I just know that it would not be as good as the book.	WordNet	*Yet* without watching the *picture show*, I just *make love* that it would not be as good as the book.
	Word2Vec	Even without watching the *moive*, I just know that it *could* not be as good *regarded* the book.
	RoBERTa	Even without *viewing* the movie, *you* just knew that it would not be as good as the book.
	Backtranslation (DE-EN)	Even without *seeing* the *film*, *all I* know *is that* it *wouldn't* be as good as the book.
The news is fake because so much of the news is fake.	WordNet	The news *be* fake because so much of the *word* is fake.
	Word2Vec	The news *becomes* fake *anyway* so much of the news is *bogus*.
	RoBERTa	The *data* is fake because so much about the *information* is fake.
	Backtranslation (DE-EN)	The *messages are* fake because so *many messages are* fake.

from the simplest (binary fallacy detection) to the most complex (fine-grained classification) tasks. Through the binary classification stage, we aim to introduce the structural and topical knowledge required to identify fallacies in arguments. The model uses this information in the subsequent (coarse-grained) stage to learn about the broad categories of fallacies. These learned coarse representations are then transferred to and trained further on the fine-grained fallacy classification objective.

Reverse Curriculum Learning (RCL) Rohde and Plaut [103] discovered that learning from simple to complex examples is sometimes not as effective as learning complex patterns directly first. Although they revised their claims in a subsequent paper [104], we explore the capabilities of the models trained with a reverse curriculum, i.e., moving inversely from complex to simple examples, which allows us to compare the different curriculum learning strategies for the task of logical fallacy identification. For RCL, we first train on the fine-grained classes and use these weights for the coarse-grained classification task. We ultimately test their applicability on the binary fallacy detection task.

4.3. Data Augmentation

Besides curriculum learning, we experiment with using data augmentation for addressing data sparsity. We devise two data augmentation strategies: modifying the original task data and adapting related benchmarks.

Augmentation by Modifying the Original Task Data. We apply commonly used text augmentation techniques for improving the performance and enhancing contextual understanding for logical fallacy detection and classification. We begin with a basic WordNet [83] similarity-based augmentation. This involves using the synsets to substitute the words in the input with words that have the closest meaning according to the synset. Second, we evaluate word embedding substitution methods based on Word2Vec and transformer embeddings. These substitutions involve finding word vectors that are closest to the input word vector in the embedding space and replacing them. Lastly, we experiment with a more recent technique of back-translation, popularized by [90] and originally proposed by [35]. This involves translating the input sentence into a language that is syntactically and morphologically dissimilar and subsequently reverse-translating this translation back to the original language. To select languages, we follow the insights from prior work [90, 35, 110]. As the parental tree for a language must be analyzed, languages that have fewer cognates are preferred as they enhance variety. Additionally, the use of two translation models trained on different datasets has been found to usually work better and provide more diversity to the output sentence. The most popular choices for back-translation model pairs are German ↔ English, Turkish ↔ English, and French ↔ English.

Table 2 shows representative examples of the obtained augmentations for two input sentences. We observed that the WordNet and Word2Vec techniques introduced excessive noise in our trials, which ended up deteriorating the performance of our models. For the back-translation, we experiment with German ↔ English translation models for the augmentation because of the syntactical dissimilarity between the two languages. Although the back-translation method was able to broaden the variety of the sentence structure, it occasionally led to the rephrasing of the actual fallacious components of the sentences. Therefore, while we believe that back-translation and transformer-based substitution together would work best with improved translation models, in this work, we focus on augmentation with RoBERTa embedding-based synonym substitution (RESS).

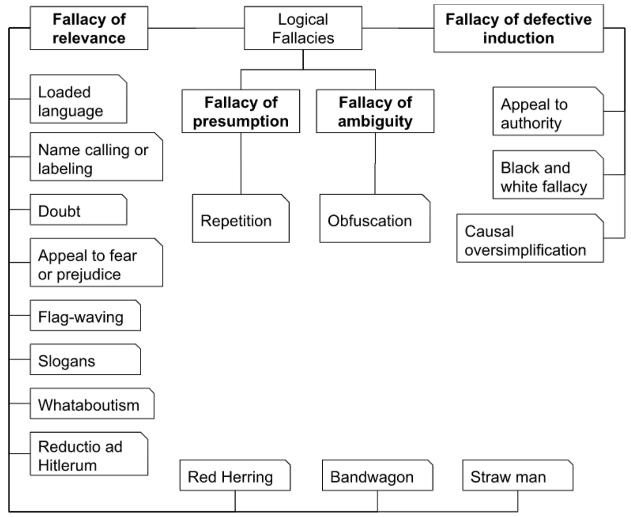


Figure 7: Three-stage taxonomy of propaganda detection.

Augmentation by Adapting Related Benchmarks. We investigate the possibility of augmenting the training data with human-curated datasets created for the related task of propaganda detection. As discussed in §3, this task applies various logical fallacy techniques including *Ad Hominem*, *Red Herring*, *Appeal to Emotion*, and *Irrelevant Authority*. We adopt the *Propaganda Techniques Corpus (PTC)* [29], which includes techniques that can be found in journalistic articles and can be judged intrinsically, without the need to retrieve supporting information from external resources. The taxonomy of PTC is illustrated in Figure 7.

The PTC dataset consists of news articles, where each sentence can have between zero, one or more fallacy annotations. As such, we adapt the PTC dataset for augmentation as follows. If a sentence contains more than one propaganda technique, then that sentence is duplicated with all its respective labels. We also combine one previous sentence, as a context, with the original labeled sentence only if the previous sentence does not belong to another fallacy class. As some of the fine-grained classes of PTC differ from those of our logical fallacy framework, we use PTC for augmentation after mapping its 18 classes to coarse-grained classes. To do so, we map the fine-grained classes in the PTC dataset to their closest fine-grained class correspondents in the logical fallacy dataset using the class definitions and descriptions. We then simply apply the broad class mapping created for the logical fallacy dataset and map the PTC fine-grained classes to the logical fallacy coarse classes. As the goal of this merging is to use the PTC coarse-grained classes for augmentation, we only leverage the training set of PTC and discard its development and test sets. Since the imbalance of the dataset worsens after merging, we use the RESS-based augmentation to augment the three under-represented classes in the merged training setup to a minimum of $n = 2000$ samples. We cap the augmentation to this amount so as to avoid repetitions and noise in the augmented dataset, which become dominant in the case of augmenting until the number of samples in the largest class ($n \approx 4,000$). We refer the reader to Table 3 for augmentation statistics.

Table 3
Training data augmentation statistics for PTC.

Fallacy Class	Pre-augmentation	Post-augmentation
Relevance	3950	3950
Defective induction	1040	2000
Presumption	536	2000
Ambiguity	42	2000

5. Experimental Setup

5.1. Evaluation

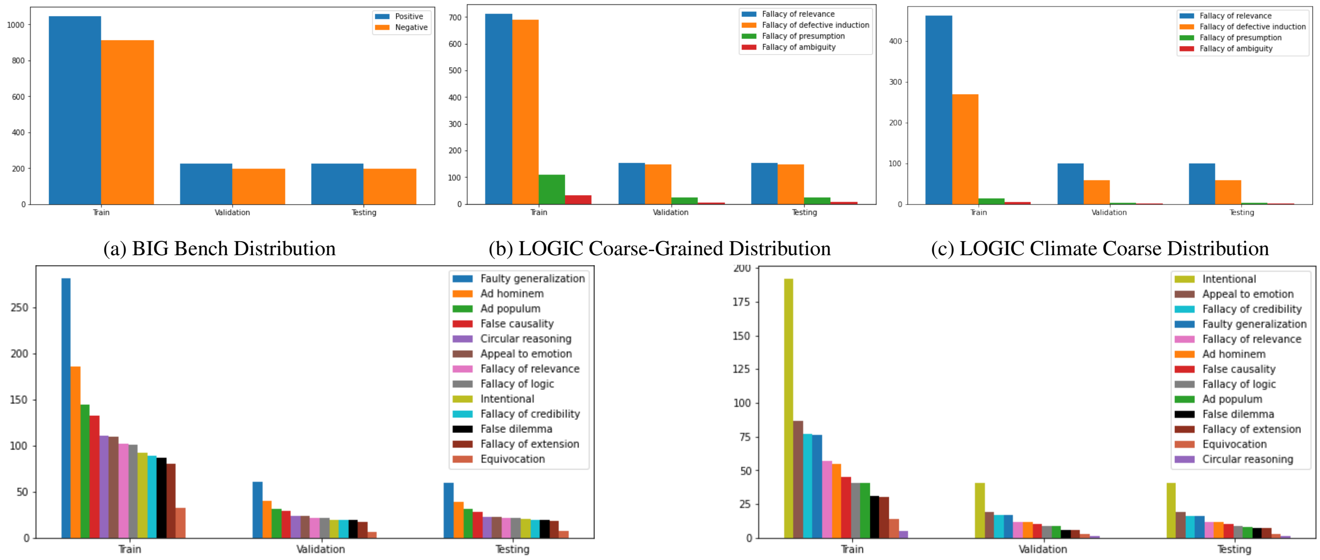
Binary Logical Fallacy Detection. BIG Bench [44] is a benchmarking dataset that is used for probing the representations of large language models to check their biases on various sub-tasks. BIG Bench includes two tasks for probing fallacies: binary logical fallacy detection and the formal fallacy syllogism negation. We use the binary fallacy detection dataset for evaluating whether the methods can distinguish between normal and fallacious arguments. We do not use the formal fallacy syllogism negation dataset since its format and purpose involve the deduction of the validity of sentences on the basis of the two provided premises, which is not directly related to the objective of this paper.

We split the BIG Bench logical fallacy dataset into training, validation, and testing sets, for which the distributions are shown in Figure 8a. The dataset is balanced and contributes 2,800 samples across all three splits.

Fine-Grained Classification. For the fine-grained classification evaluation, we use the LOGIC and LOGIC Climate datasets introduced in [62]. There are thirteen classes within the LOGIC and LOGIC Climate fallacy datasets as described in Table 1. The LOGIC dataset contains everyday fallacious arguments belonging to various topics. We use the cleaned and revised version of this dataset.[3] The LOGIC Climate dataset consists of climate change news articles and fallacious arguments detected in them. We use LOGIC Climate as an evaluation-only dataset. As observed in Figures 8d and 8e, the distributions between the two datasets are different, with *Intentional* being the largest class in the Climate dataset, whereas it is one of the under-represented classes in the LOGIC fine-grained dataset. The LOGIC Climate dataset is included to test the ability of our models to learn these under-represented classes as well as the transferability of the model's knowledge to unseen topics.

Coarse-Grained Classification. We evaluate the coarse-grained classification based on data inferred from the LOGIC and LOGIC Climate datasets. The coarse-grained datasets are curated by mapping fine-grained classes from these two datasets to the coarse-grained categories following Figure 1. In the mapping process, fine-grained classes with $k \leq 20$ samples were removed from their corresponding coarse class

[3] https://github.com/tmakesense/logical-fallacy/tree/main/dataset-fixed



(a) BIG Bench Distribution

(b) LOGIC Coarse-Grained Distribution

(c) LOGIC Climate Coarse Distribution

(d) LOGIC Fine-Grained Distribution

(e) LOGIC Climate Fine-Grained Distribution

Figure 8: Dataset Distributions.

if this coarse class was not under-represented. For LOGIC, we left out the fine-grained classes *Fallacy of Relevance*, *Fallacy of Logic* and *Intentional*, as their mapping to coarse-grained classes was mostly ambiguous for the data examples. This resulted in a four-way coarse classification task for LOGIC and LOGIC Climate into: *Fallacy of Relevance, Fallacy of Defective Induction, Fallacy of Ambiguity, and Fallacy of Presumption.*

The coarse version of the LOGIC dataset shows a clear imbalance. A visual representation of the distribution is shown in Figure 8b. To ensure that the testing and validation splits are representative of this distribution, we sample all our splits using stratification. The splits for LOGIC Climate [62] are created in a similar manner. Their distribution is shown in Figure 8c.

Evaluation Regime and Metrics. We test the models on the BIG Bench and LOGIC datasets by fine-tuning and curriculum learning. We apply the models trained on the LOGIC dataset for fine- and coarse-grained classification in a zero-shot fashion to the corresponding LOGIC Climate data. We report the average model performance over three runs. We use weighted precision, recall, F1-score, and accuracy to characterize the performance of different models. Weighted measures are used to assess the per-class scores more accurately for the available unbalanced testing sets.

5.2. Implementation Details

Baselines. We experiment with six NLI/MNLI base version models: BERT [85], DeBERTa [98], DistilBERT [22], Electra [55] and RoBERTa [99]. We utilize NLI models because we find that they perform better on the tasks of logical fallacy identification. This can be expected given that they are trained on a larger variety of data than MLM or similar models. NLI models have also been shown to have a better grasp of concepts than their MLM counterparts and to produce embeddings with better semantic representations [25]. To contextualize the results, we also evaluate two simple baselines: a random baseline and a baseline that picks classes based on the relative frequency of classes in the training set.

Instance-Based Reasoning. For all the experiments, we use a sweep over the hyperparameters such as weight decay (L2 regularization), learning rate, and feed-forward network dropout rate. Since we use a threshold to filter the fetched similar examples from the retriever, based on cosine similarity, we use a sweep over the used threshold as well. We then use the best combination on the development set and report the average performance on three runs using the best hyperparameters. We use the NLI-initialized Electra-base LM as the underlying encoder for generating input sentence embeddings and train our models for ten epochs in each experiment. As we observe that the 0.5 similarity filter for fetched similar cases from the retrievers yields the best results, we apply a similarity filter on top of the retrievers discarding any fetched case whose cosine similarity to the new case is below 0.5. We use multi-head attention with eight heads. The number of cases (k) used in our experiments ranges from 1 to 10. We do not experiment with more cases due to the stable trend seen when increasing the number of cases.

Prototype-Based Reasoning. We experiment with a different number of positive and negative prototypes and find that 49 positive prototypes and 1 negative prototype works best for the fine-grained classification task. We keep the same number of prototypes for the binary, coarse-, and fine-grained classification tasks. To train the negative prototype, we also include a “None” class, supported by the examples from the negative class in the binary classifica-

Table 4
The corresponding runtime (in seconds) for the experiments done with each method and model per epoch, for the best models on the binary fallacy detection task.

Method	Model	Binary	Coarse	Fine
NLI	Electra	66.0	89.0	108.5
IBR	Electra	103.7	147.6	191.2
PBR	Electra	15.5	38.2	114.8
KI	K-BERT	96.4	107.3	123.9

tion task. We use the NLI-initialized Electra-base as the underlying encoder for generating input sentence embeddings and report the best metrics averaged over three runs. We monitor the validation loss to choose the best model and use early stopping ($patience = 10$) to prevent overfitting. We also compute class weights to handle any imbalance in the training dataset.

Knowledge Injection. For the experiments with K-BERT, we perform a grid search and report the results for the best-performing set of parameters. We use grid search to find the optimal parameters: a learning rate of 2×10^{-5} with a dropout of 0.5. We use the BERT-base model by injecting knowledge from CSKG and fine-tune the KI model over different datasets for five epochs.

Curriculum Learning. For a fair comparison of the curriculum learning pipeline against the baseline model, we report scores on the default hyperparameters of the fine-tuned model, though we expect an overall increase for all metrics of at least 2-3% when these models are tuned. We train for 5, 8, and 10 epochs for each tuning stage respectively in the curriculum learning pipeline to avoid loss of knowledge across multiple fine-tuning stages. We fix the batch size to 32, the learning rate to 5×10^{-5}, and we use the cosine learning rate scheduler while keeping the remaining hyperparameters for our experiments unchanged.

Data Augmentation. We conduct experiments with different augmentation techniques for word-based and sentence-based augmentation using NLPAug [75]. We experiment with a range of augmentation probabilities and the number of suitable substitutions for RESS, discovering the best results with 5 substitutions, while over 10 substitutions leads to a decrease in performance. Similarly, we obtain the best results with the augmentation threshold set between $80-90\%$, and a maximum of three replacements per argument.

6. Results

We run all our experiments on a cluster of A100-PCIE-40GB GPUs. The runtime of our experiments depends on the family of methods used, the dataset, and the size of the model being fine-tuned. We report runtimes for our best models on the binary fallacy detection task as well as coarse- and fine-grained classification task in Table 4. The recorded times show that the runtime of the models is mostly within the same order of magnitude of tens of seconds for binary, around a hundred seconds for coarse-, and between one and two hundred seconds for fine-grained classification. The PBR model is exceptionally efficient to train - its runtime is lower or comparable to the baseline NLI model. IBR takes the longest to run, taking one order of magnitude longer than PBR for binary classification and twice as long for fine-grained classification. As the encoding stage that is part of the retriever in the IBR framework is executed as a preprocessing step and is presented as a look-up table in the training stage, the time that is needed to encode all training examples with an encoder is excluded in this table.

Table 5
Main results for the best models for each method family on binary logical fallacy detection on the BIG Bench dataset.

		BIG Bench (Binary)			
Method	**Model**	Acc	P	R	F1
Random	/	0.499	0.508	0.499	0.499
Frequency	/	0.501	0.501	0.501	0.501
NLI	Electra	0.995	0.995	0.995	0.995
NLI FCL	Electra	0.995	0.995	0.995	0.995
IBR	Electra	**0.997**	**0.997**	**0.997**	**0.997**
PBR	Electra	0.984	0.984	0.984	0.984
KI	BERT	0.776	0.779	0.775	0.777

6.1. Overview of the Results

Tables 5, 6, and 7 show the obtained results for each method: NLI baseline, NLI with FCL, IBR, PBR, and KI on the tasks of logical fallacy detection, coarse-grained classification, and fine-grained classification. Here, we present the best result per method, indicating the corresponding model, and dive into each method in the subsequent sections. All presented results use augmentation data based on modifying the original task data (RESS).

We observe that all methods besides KI can solve the logical fallacy detection task with a nearly perfect F1-score ($98.4\%-99.7\%$), with the IBR method using an NLI-Electra language model reaching the best performance (cf. Table 5). The results on the coarse- and fine-grained tasks show more intriguing patterns. IBR again obtains the best performance on the in-domain task (LOGIC dataset) achieving 82.7% and 62.7% F1-scores on the coarse-grained and fine-grained datasets, respectively (cf. Tables 6,7). However, the trends are more mixed when generalizing to the out-of-domain task of LOGIC Climate. The transfer learning F1-score of IBR (46.6%) falls behind the PBR model (57.3%) on the coarse-grained classification of the LOGIC Climate data (cf. Table 6), while the performance of the NLI method with curriculum learning performs on par with IBR ($\sim 24\%$) for the LOGIC Climate fine-grained task outperforming the other models (cf. Table 7). Among the different language models, most of our methods achieve the best results when using Electra with NLI initialization.

All in all, we observe that CBR models (IBR and PBR) perform better than baseline, curriculum learning, and KI, while offering inherent explainability. We observe a signif-

Table 6
Main results for the best models for each method family on the coarse-grained classification.

		LOGIC				LOGIC Climate			
Type	**Model**	Acc	P	R	F1	Acc	P	R	F1
Random	/	0.249	0.413	0.249	0.298	0.249	0.508	0.249	0.323
Frequency	/	0.415	0.413	0.415	0.413	0.446	0.508	0.446	0.468
NLI	Electra	0.767	0.765	0.767	0.764 ±0.01	0.509	**0.602**	0.509	0.498 ±0.01
NLI FCL	DeBERTa	0.758	0.748	0.758	0.751 ±0.02	0.491	0.552	0.491	0.490 ±0.02
IBR	Electra	**0.829**	**0.827**	**0.829**	**0.827** ±0.01	0.459	0.585	0.459	0.466 ±0.01
PBR	Electra	0.708	0.711	0.708	0.695 ±0.03	**0.578**	0.570	**0.578**	**0.573** ±0.03
KI	BERT	0.787	0.781	0.782	0.781 ±0.03	0.385	0.589	0.385	0.415 ±0.01

Table 7
Main results for the best models for each method family on the fine-grained classification.

		LOGIC				LOGIC Climate			
Type	**Model**	Acc	P	R	F1	Acc	P	R	F1
Random	/	0.076	0.094	0.076	0.079	0.077	0.124	0.077	0.085
Frequency	/	0.094	0.094	0.094	0.093	0.079	0.120	0.079	0.080
NLI	Electra	0.602	0.614	0.602	0.599 ±0.02	0.229	0.276	0.229	0.217 ±0.01
NLI FCL	Electra	0.613	0.624	0.613	0.610 ±0.04	0.236	0.304	0.236	0.243 ±0.02
IBR	Electra	**0.631**	**0.638**	**0.631**	**0.627** ±0.01	**0.254**	0.281	**0.254**	**0.245** ±0.01
PBR	Electra	0.574	0.600	0.574	0.574 ±0.01	0.199	**0.330**	0.199	0.166 ±0.01
KI	BERT	0.488	0.478	0.488	0.482 ±0.03	0.106	0.092	0.106	0.090 ±0.02

icant gap between the performance of all the models on the in-domain dataset (LOGIC) and the out-of-domain dataset (Climate LOGIC), particularly in the fine-grained dataset, which indicates the complexity of knowledge transfer in logical fallacies from topic to topic. Zooming in on the performance of the CBR models on the out-of-domain setting, prototypical examples seem to be more helpful for approaching coarse-grained classes, while simply focusing on the semantic similarity of previous cases to approach new ones is performing better for fine-grained logical fallacies.

These results provide insights into the overall trends between the method families, however, many questions remain open. We next investigate the following questions. *Does augmentation help? (§6.2) Does curriculum learning have a consistent impact across models? (§6.3) Does commonsense knowledge and reasoning by cases have a robust and notable effect on the model performance? (§6.4) Do instances, prototypes, and commonsense knowledge provide intuitive explanatory mechanisms? (§6.5) Which classes are helped by our methods, and which remain difficult to address? (§6.6)*

6.2. Effect of Augmentation

As the LOGIC dataset is highly imbalanced, we hypothesize that data augmentation will help to address this gap, ultimately bringing better performance on this dataset. The challenge with standard augmentation techniques is that logically fallacious statements have a certain structure and arrangement, which we wish to retain even after applying the augmentation technique. We experiment with modifying the original dataset using our RESS method and including data from the neighboring propaganda dataset, PTC.

The obtained results for our models using Forward Curriculum Learning are shown in Table 8. We observe that augmentation is overall helpful on the fine-grained task and harmful on the coarse-grained task. Within the fine-grained task, the RESS augmentation always outperforms the baseline which confirms our expectation that data sparsity is an important challenge and it can be addressed through RoBERTa-based synonym substitution. The PTC augmentation is partially beneficial for some models, owing to the overlap between the propaganda and the logical fallacy data. However, the effect of augmentation with PTC is dominantly negative, signaling that despite the overlap, this dataset is prohibitively different from the logical fallacy data. On the coarse-grained data, we see that augmentation has a negative impact on four out of five models even for the RESS augmentation method.

We investigate this further by monitoring the augmentation impact per class. Comparing the performance of the models between pre-augmented data and post-augmented data in the coarse-grained dataset, models trained on the post-augmented data perform slightly better (up to 11%) on the *Ambiguity* class that is initially under-represented. However, the effects are adversary for the three other classes that initially have much more data points. We attribute this observation to the trade-off between enriching the data and disturbance in the natural distribution that the initial dataset

Table 8
Data augmentation results on the LOGIC dataset: no data augmentation, augmentation with RESS, and augmentation with PTC. All the models in the table are trained using the Forward Curriculum Learning framework – FCL.

		Coarse-grained				**Fine-grained**			
Model	**Augmentation**	Acc	P	R	F1	Acc	P	R	F1
BERT	-	**0.747**	**0.737**	**0.747**	**0.739** ±0.01	0.549	0.571	0.549	0.552 ±0.01
	RESS	0.727	0.717	0.727	0.721 ±0.03	**0.586**	**0.613**	**0.586**	**0.584** ±0.02
	PTC	0.696	0.651	0.696	0.667 ±0.01	0.567	0.590	0.567	0.570 ±0.02
DeBERTa	-	**0.765**	**0.778**	**0.765**	**0.766** ±0.03	0.564	0.627	0.564	0.576 ±0.02
	RESS	0.758	0.748	0.758	0.751 ±0.02	**0.604**	**0.632**	**0.604**	**0.608** ±0.01
	PTC	0.710	0.675	0.710	0.683 ±0.01	0.537	0.590	0.537	0.547 ±0.04
DistilBERT	-	0.711	0.698	0.711	0.704 ±0.01	0.507	0.529	0.507	0.509 ±0.01
	RESS	**0.713**	**0.703**	**0.713**	**0.706** ±0.02	**0.520**	**0.550**	**0.520**	**0.525** ±0.03
	PTC	0.704	0.652	0.704	0.664 ±0.02	0.492	0.534	0.492	0.495 ±0.04
RoBERTa	-	**0.752**	**0.746**	**0.752**	**0.742** ±0.01	0.504	0.538	0.504	0.510 ±0.01
	RESS	0.713	0.710	0.713	0.706 ±0.02	0.569	0.578	0.569	0.565 ±0.02
	PTC	0.699	0.647	0.699	0.666 ±0.01	**0.603**	**0.620**	**0.603**	**0.595** ±0.01
Electra	-	**0.758**	**0.745**	**0.758**	**0.749** ±0.02	0.602	0.621	0.602	0.608 ±0.02
	RESS	0.722	0.711	0.722	0.716 ±0.03	**0.613**	**0.624**	**0.613**	**0.610** ±0.04
	PTC	0.725	0.689	0.725	0.690 ±0.01	0.578	0.596	0.578	0.581 ±0.02

Table 9
Curriculum learning results with different NLI and PBR models on Big Bench and the LOGIC coarse- and fine-grained datasets. All models use RESS augmentation.

		Binary (BIG Bench)			**Coarse-grained**			**Fine-grained**		
Model	**CL Type**	P	R	F1	P	R	F1	P	R	F1
BERT	-	**0.848**	**0.845**	**0.845** ±0.01	0.714	0.718	0.717 ±0.04	0.583	0.583	0.583 ±0.01
	FCL	-	-	-	0.717	0.727	0.721 ±0.03	**0.613**	**0.586**	**0.584** ±0.02
	RCL	0.826	0.827	0.826 ±0.00	**0.783**	**0.779**	**0.778** ±0.02	-	-	-
DeBERTa	-	**0.988**	**0.988**	**0.988** ±0.00	0.746	0.740	0.741 ±0.03	0.607	0.593	0.592 ±0.02
	FCL	-	-	-	0.748	0.758	0.751 ±0.02	**0.632**	**0.604**	**0.608** ±0.01
	RCL	0.908	0.892	0.889 ±0.05	**0.779**	**0.785**	**0.780** ±0.02	-	-	-
DistilBERT	-	**0.848**	**0.847**	**0.847** ±0.01	0.684	0.695	0.683 ±0.02	0.508	0.513	0.505 ±0.02
	FCL	-	-	-	0.703	0.713	0.706 ±0.02	**0.550**	**0.520**	**0.525** ±0.03
	RCL	0.844	0.842	0.841 ±0.01	**0.704**	**0.719**	**0.711** ±0.03	-	-	-
RoBERTa	-	**0.983**	**0.983**	**0.983** ±0.01	0.719	0.714	0.716 ±0.01	0.560	0.545	0.545 ±0.02
	FCL	-	-	-	0.710	0.713	0.706 ±0.02	**0.578**	**0.569**	**0.565** ±0.02
	RCL	0.900	0.899	0.899 ±0.01	**0.736**	**0.741**	**0.732** ±0.01	-	-	-
Electra	-	**0.995**	**0.995**	**0.995** ±0.00	0.765	0.767	0.764 ±0.01	0.614	0.602	0.599 ±0.02
	FCL	-	-	-	0.711	0.722	0.716 ±0.03	**0.624**	**0.613**	**0.610** ±0.04
	RCL	0.957	0.957	0.957 ±0.01	**0.779**	**0.782**	**0.775** ±0.03	-	-	-

possesses. Although by augmenting the dataset we achieve higher performance on the sparse class, the augmentation has a negative effect on the other classes. This also explains the success of data augmentation on the fine-grained classes, which mostly have a low number of training examples. In summary, while augmentation does not increase performance on the coarse-grained task variant, its success on the fine-grained task and on sparsely represented classes motivates the need for further analysis and development of data augmentation methods.

6.3. Effect of Curriculum Learning

The effect of curriculum learning for models trained on RESS-augmented data can be seen in Table 9. We see clear trends for all three tasks that are consistent across the NLI models.[4] We observe that curriculum learning is beneficial for the coarse-grained and the fine-grained tasks, whereas it is detrimental for the binary detection task.

Among the two CL variants tested on the coarse-grained task, we see that RCL performs better than FCL. With the

[4] We observe identical trends when using CL together with the PBR-based Electra model.

reverse curriculum, we notice that using the fine-grained weights for coarse-grained classification improves scores considerably for all models, with DeBERTa performing the best with a 0.78 weighted F1 score. This means that all models learn more about the coarse-grained task from the fine-grained task compared to learning from the binary fallacy detection task (i.e., when we use the BIG Bench initialization weights instead of NLI). Three out of five models still improve their performance in the FCL setup. However, Electra and RoBERTa decrease their performance and increase their variance between runs in this setup, which can be attributed to their sensitivity to hyperparameter values.

On the other two tasks, we only compare a single CL variant to the baseline models. We do not test FCL on the binary task, as there is no task that is easier than the binary detection in our pipeline to initialize the weights from. Analogously, we do not test RCL on the fine-grained task, because our pipeline has no task that is more complex than the fine-grained classification to initialize the model weights from. For the fine-grained evaluation, we see that the coarse-grained initialization performs better than the original NLI initialization. We note that using a forward curriculum leads to an increase in at least 1% F1-scores throughout, with Electra performing the best in this category with 0.61 weighted F1. As described before, we expected the benefit of FCL on the fine-grained task, as the forward curriculum allows the model to learn in stages of increasing difficulty, which enhances model performance at each granularity. We also observe that increasing the number of epochs at each level of the pipeline helps to reduce the forgetting of knowledge during downstream, fine-grained tasks. However, we observe a negative impact when using RCL for binary fallacy detection, which indicates that this task does not benefit from the initialization of models on the fallacy classification tasks.

Overall, our results reliably show that the curriculum learning pipeline is capable of improving performance for the logical reasoning task of fallacy detection and the coarse representations are effective in the final stage of tuning even though they do not always outperform the other initializers in the coarse stage.

6.4. Analysis of Method Sensitivity and Ablations

Next, we perform ablations of the components of our methods and investigate key parameter settings.

6.4.1. Instance-Based Reasoning

We observe that the IBR method performs the best among the methods across all datasets (cf. Table 5, 6, and 7). This indicates that the idea of using similar instances to solve a new problem is effective at various levels of granularity. Although considering the common belief about the trade-off between predictive ability and interpretability ([66], [18], [21]), IBR models could have not behaved as well as other methods discussed, inline with [102] and [57], we observe that IBR models offer good accuracy, as well as potentials for explainability [101]. We investigate the effect of the optimal number of similar cases, and of the designs of the retriever and the adapter on the performance of the method.

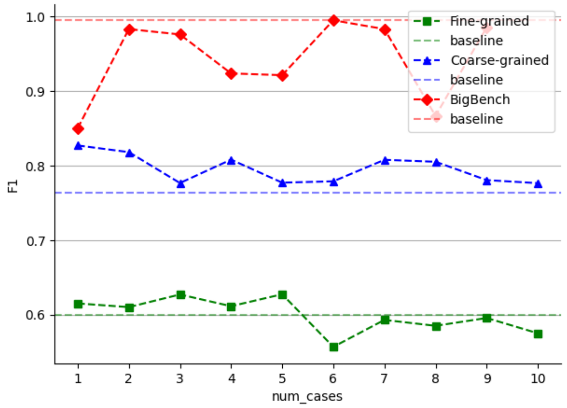


Figure 9: Comparing the performance of the model being exposed to different numbers of similar cases in the IBR framework.

We do not investigate different design choices for the Classifier, which is currently a feed-forward neural network, and as such, a trivial step in the framework.

Optimal number of similar cases. Considering the complexity of sentences containing a logical fallacy, as well as the wide range of subjects they cover and revolve around, it is most likely that for some sentences, there would be more than one already-seen sentence that would be useful or essential for the model's reasoning. It is worth mentioning that although similar cases can potentially help the model classify certain sentences better, due to the fact that retrievers are imperfect and also language models can only capture the surface meaning of the sentences (form in the language) and not necessarily understand the meaning [11], adding more similar cases to the model can be considered noise and not useful. On this ground, we check the effect of the different number of cases shown to the model and assess their impact on the model's performance in Figure 9. As can be observed, for the coarse-grained and fine-grained datasets, there is a soft downward transition between using fewer examples and more examples that shows using more similar cases does not help the model as much as it hinders the process. This pattern differs further between coarse-grained classes and fine-grained classes. In the coarse-grained classification, regardless of the number of cases, the performance of the IBR model is always superior to the baseline, while in the fine-grained classification, having more than five similar examples would hurt the performance and cause a drop even below the baseline. Considering the fact that fewer similar cases means less noise and more similar cases means better coverage in terms of the potential aid from similar cases, we conclude that higher coverage cannot compensate for the excess noise added to the model.

Design of Retriever. For sentences that contain logical fallacies, nuances in meaning are vital to distinguish the actual relevant similar sentences from the ones that are only revolving around the same subject. Building upon this idea,

Table 10
Comparing the performance of the model using different retrievers to fetch similar cases.

Dataset	Retriever	P	R	F1
BIG Bench	empathy	0.969	0.969	0.969
	all-MiniLM-L6-v2	0.983	0.983	0.983
	paraphrase-MiniLM-L6-v2	0.861	0.823	0.822
	SimCSE	**0.997**	**0.997**	**0.997**
Coarse-grained	empathy	0.815	0.813	0.808
	all-MiniLM-L6-v2	0.807	0.801	0.796
	paraphrase-MiniLM-L6-v2	0.788	0.785	0.786
	SimCSE	**0.827**	**0.829**	**0.827**
Fine-grained	empathy	0.622	0.607	0.609
	all-MiniLM-L6-v2	0.616	0.616	0.611
	paraphrase-MiniLM-L6-v2	0.588	0.567	0.567
	SimCSE	**0.638**	**0.631**	**0.627**

Table 11
Comparing the performance of the IBR model with and without using the attention mechanism.

Dataset	Attn	Acc	P	R	F1
BIG Bench	w	**0.997**	**0.997**	**0.997**	**0.997**
	w/o	0.826	0.829	0.826	0.824
Coarse-grained	w	**0.829**	**0.827**	**0.829**	**0.827**
	w/o	0.768	0.762	0.768	0.764
Fine-grained	w	**0.631**	**0.638**	**0.631**	**0.627**
	w/o	0.620	0.631	0.620	0.619

we investigate different pre-trained language models as the retriever's encoder (§4.1.1). The comparison between these encoders is illustrated in Table 10. We observe the superior performance of SimCSE on all the datasets with different granularity levels. We attribute this to the contrastive learning objective used in SimCSE. MiniLM with six layers, an all-round model tuned for many use cases, comes in the second rank. Both SimCSE, as well as the all-MiniLM model trained on NLI, show the relevance and effectiveness of NLI for logical fallacy prediction. However, the paraphrase models, though trained on similar tasks such as AllNLI (concatenation of SNLI [15] and MultiNLI [124]) and sentence compression, come in the last rank.

Design of Adapter. We compare our results on three datasets with and without using the attention mechanism in the third stage (adaptation). The results of this ablation study are presented in Table 11. Confirming our hypothesis, we note better performance in the presence of an attention mechanism to adjust the weights on similar cases when reasoning about the new case C. This observation is consistent across all datasets, which means that attention is a robust adaptation mechanism that helps the model to attend to relevant cases regardless of the granularity of the task.

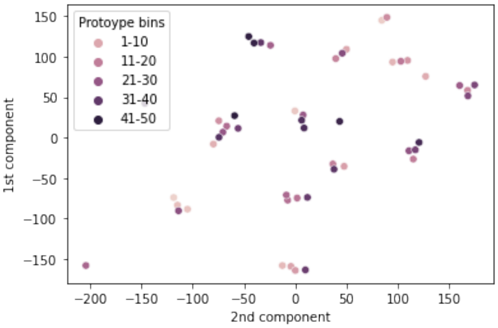


Figure 10: T-SNE clustering (perplexity=2) of the 50 prototype tensors used for fine-grained classification. We flatten the prototypes thereby reducing the 98,304-dimensional data to just 2 dimensions.

6.4.2. *Prototype Learning*

We dive deeper into the connection between prototypes and classes, and the sensitivity of our PBR model on the number of prototypes.

Prototypes Characterizing Classes. We find the prototypes responsible for the classification of each training example and assign them to the respective labels. We observe that the masking mechanism, which we introduce to the PBR method, helps to associate certain prototypes to particular classes. While we expect to see a distinct set of prototypes representative of each class, we observe a mix of distinct and common prototypes representing a particular label. For example, for the class *Fallacy of Logic*, we get prototypes 6, 13, 38, and 7 as the strongest representatives. However, we observe prototype 38 to be a strong representative for five other class labels as well. We believe this is because of the nature of the overlap of fallacy classes, e.g., a fallacious sentence might have flavors of both *Appeal to Emotion* and *Ad Populum*, even if only one of them is annotated as the correct class. Further, we cluster the 50 prototype tensors used for the benchmarking of the fine-grained classification task and color code the prototypes based on their indices, as shown in Figure 10. Here, prototypes 1-10 have a light color and as we go towards prototypes 40-50, the shades get darker. We observe a certain grouping of prototype tensors, which may indicate unique features captured by the prototypes per class.

Prototypes Characterizing Classes. Figure 11 shows the trend of F1-score on the fine-grained classification task for a different number of prototypes. We assign 10% of the prototypes to the negative class for this specific benchmarking. We observe a high sensitivity of the Prototex model to the number of prototypes, where having a too low or too high number of prototypes yields suboptimal results. We find that having a total of 50 (5 negatives) or 100 (10 negatives) prototypes yields the best performance. The PBR method is highly sensitive to the number of prototypes, and,

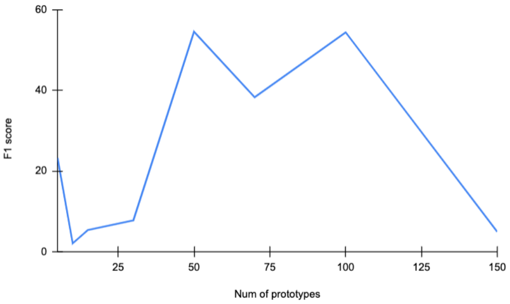


Figure 11: Comparison of the performance of the PBR model for different numbers of prototypes on the LOGIC fine-grained classification task.

Table 12
Comparing the performance of the KI method with and without using similarity ranking of relations.

Dataset	Similarity ranking	Acc	P	R	F1
BIG Bench	w	**0.776**	**0.779**	**0.775**	**0.777**
	w/o	0.750	0.770	0.740	0.739
Coarse-grained	w	**0.787**	**0.781**	**0.782**	**0.781**
	w/o	0.760	0.706	0.746	0.721
Fine-grained	w	**0.488**	0.478	**0.488**	**0.482**
	w/o	0.468	0.489	0.407	0.419

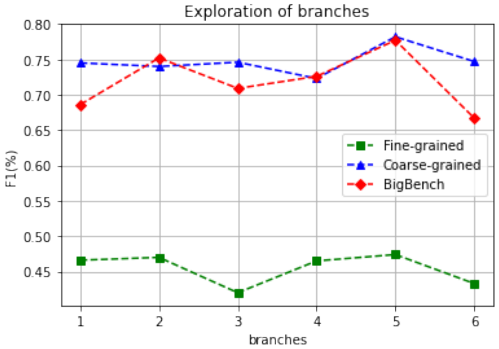


Figure 12: F1-scores for different branching factors for KI.

thus, it is important to tune this hyperparameter for new datasets. Moreover, we investigate whether introducing negative prototypes is beneficial to the PBR model. Similar to [31], we find that including negative prototypes together with a "None" prediction class brings better performance on the logical fallacy coarse- and fine-grained classification tasks, though the performance gain in our case is more limited (2-3% increase in absolute F1-scores).

6.4.3. Knowledge Injection

We assess the performance of K-BERT, [71] on identifying logical fallacies in terms of the decisions made when injecting knowledge from the external KG (namely, CSKG). The information gained from CSKG is used for forming sentence trees that are used as the primary points for knowledge injection. In the process of knowledge injection with CSKG, the tokens of the sentences are broken down and the triples containing the token are appended to the token to form a sentence tree. By default, the KI method creates a sentence tree by using a maximum of two such branches per token. The exploration depth of the relation is limited to a single hop. With regards to picking useful relations, the KI method uses a brute force method for choosing triples for tokens that have multiple relations present within the knowledge base. We investigate the effect of different knowledge selection strategies, numbers of branches, and hops.

Effect of Similarity Ranking of Relations. While information is appended to the sentence tree, we hypothesize that it is more meaningful to have a selection strategy in effect to select relations that add relevant knowledge to the sentence, and this serves as a point for an ablation study. This similarity ranking strategy enhances the performance of the KI method consistently over three different tasks for the different datasets, as observed in Table 12. The performance gain is around six F1-score points for each of the three datasets, confirming our hypothesis that selecting knowledge based on relevance is important, as also shown in [77]. This result also motivates the need for more advanced methods for context-dependent knowledge selection.

Branching Factor Size. In the knowledge layer of K-BERT, the default branching factor is 2. Here, we analyze the performance of the model for different branching factors chosen (with similarity ranking of relations). As observed from Figure 12, a branching factor of 5, gives better performance over the other branching factors. We take this branching factor to represent a sweet spot between providing K-BERT with too little additional knowledge ($b < 5$) and too much additional knowledge ($b > 5$).

Number of hops. The base KI model uses only 1 hop of knowledge. A single hop corresponds to discovering the first relation and entity connected with the token, while by using multiple hops, we discover subsequent depths of relations based on the entities associated with them. Our analysis shows that, in the multi-hop setup, the performance of K-BERT decreases by 3-4%. The drop in performance can be explained by the noise introduced by including multiple hops without careful filtering of the expansion. This finding is consistent with the finding of the best branching factor size that the KI model works better when presented with a smaller set of relevant relations. We look closer at the quality of the retrieved knowledge in the next section.

6.5. Qualitative Analysis

We analyze four cases for which the base model predicts an incorrect class, and our IBR and PBR methods change the prediction to the correct class. The KI method predicts the last two examples correctly as well.

Quality of the Retrieved Cases. For these four exemplars, Table 13 shows the retrieved instances by IBR and prototypical examples by PBR. For PBR, we show the two nearest training examples to the nearest prototype for a given input. We note that 6 out of 8 examples for IBR and all 8 examples for PBR come from the same class, which indicates

Table 13
Input arguments with their fetched similar cases. We mark the exemplars from the same class as the input in bold.

Class	Input Sentence	Similar Cases (IBR)	Prototypical Cases (PBR)
Ad Populum	Everyone is going to get the new smart phone when it comes out this weekend. Why aren't you?	(1) **I'm gonna get an iPhone because everybody else has an iPhone and they're cool.** (2) **Everyone wants the iPhone 11 because it's the best phone on the market!**	(1) **Everyone seems to support the changes in the vacation policy, and if everyone likes them, they must be good.** (2) **Everyone is buying the new iPhone that's coming out this weekend. You have to buy it too.**
Fallacy of Logic	surgeons have X-rays to guide them during an operation, lawyers have briefs to guide them during a trial, carpenters have blueprints to guide them when they are building a house. Why, then, shouldn't students be allowed to look at their textbooks during an examination?	(1) **Doctors refer to medical books all the time when they are treating patients. In the same way, I should be allowed to use a textbook in my medical exam.** (2) **If I say that a surgeon should be allowed to use a guidebook to carry out surgery like a student can use open notes on a test, I have made a ...**	(1) **All Paul Newman movies are great. All great movies are Oscar winners. Therefore, all Oscar winners are Paul Newman movies.** (2) **The lady in the pink dress is Julia Roberts. The reporter thinks Julia Roberts drives a Prius. Therefore, the reporter thinks the lady in the pink dress drives a Prius.**
Faulty Generalization	Everyone knows that teenagers are lazy	(1) **If we let teenagers wear whatever they want to school, they will no longer respect the rules and academic performance will decline.** (2) If we don't teach teens to work harder, the human race is doomed	(1) **If we allow a housing development to be built on Sunny Lake, a resort will come next, and soon we won't have any wilderness left!** (2) **Michael is part of the Jackson Five. Without Tito and company, he will never make it.**
Faulty Generalization	If you forget to floss, you will get cavities, and if you get cavities, you will lose all your teeth by the time you're 30	(1) **If you don't eat breakfast, you'll slouch in your desk. If you slouch in your desk, you'll hurt your back. If you hurt your back, you'll never become President.** (2) four out of five dentists agree that brushing your teeth makes your life meaningful	(1) **If we allow gay people to get married, then the next thing you know people will be wanting to marry their pets!** (2) **You smoke pot? If you keep doing that, you'll be a heroin addict within two years.**

that the modified decision in these cases correlates with obtaining helpful (or even representative) examples from the same class. We note, however, this is not always the case - the retrieved examples for IBR and PBR can also be from different classes. We observe that the corrected prediction of IBR and PBR is based on two scenarios. The first situation, shown with the first three examples for IBR in Table 13, is when the retrieved examples reflect surface similarity, which curiously still helps the model to change its decision. The second situation, observed for the last example of IBR and most PBR examples, is when the model captures the structural similarity and more abstract semantics. As we hypothesize that informal fallacies require a mixture of both aspects, observing that IBR and PBR capture them to different extents is encouraging for future work. At the same time, we also observe cases where the model correctly changes its prediction even though some of the retrieved cases belong to different classes and it is not clear how they help the model prediction. This shows the impact of the other components of our methods (the Adapter and Classifier components for IBR, or the rest of the neural architecture in PBR), but also motivates the need for future work on better models for retrieving semantically and pragmatically similar cases.

Quality of the Retrieved Knowledge. Table 14 shows

Table 14
Examples of extracted triples with our KI method. We use an asterisk '*' to indicate the examples that have been classified correctly by KI.

class	Input Sentence	Sample Triples
Ad Populum	Everyone is going to get the new smart phone when it comes out this weekend. Why aren't you?	(phone, *able to*, communicate), (phone, *intent to*, give or get information), (weekend, *related to*, relax)
Fallacy of Logic	surgeons have X-rays to guide them during an operation, lawyers have briefs to guide them during a trial, carpenters have blueprints to guide them when they are building a house. Why, then, shouldn't students be allowed to look at their textbooks during an examination?	(surgeons, *related to*, operation), (operations, *related to*, surgery), (student, *related to*, education), (lawyers, *related to*, law), (students, *able to*, give exams)
Faulty Generalization	*Everyone knows that teenagers are lazy	(teenager, *capable of*, looking), (teenager, *capable of*, performing), (teenager, *is a*, juvenile person), (teenager, *located near*, street)
Faulty Generalization	*If you forget to floss, you will get cavities, and if you get cavities, you will lose all your teeth by the time you're 30	(floss, *used for*, good oral hygiene), (floss, *related to*, teeth), (floss, *related to*, dental floss), (floss, *related to*, mouth)

the commonsense triples retrieved by KI as background knowledge. As mentioned above, the KI method predicts the first two examples incorrectly, and the last two examples correctly. In example 1, we see that while the retrieved triples focus on the word *phone*, it is the word *everyone* in the sentence that is the main clue to the fact that the sentence belongs to the class *Ad Populum*. The second example shows a case where the background knowledge misleads the model about the subject of the sentence, thus hindering it to perform a correct classification. In the third and fourth examples, the model is able to correctly classify the example as *Faulty Generalization*, and we believe that this correlates with the quality of the retrieved knowledge. For instance, in the last example, BERT receives relevant knowledge such as flossing being used for good oral hygiene and floss related to teeth, which may have helped the model to overturn the wrong prediction into a correct one.

6.6. Per-Class Analysis

Table 15 shows the per-class performance of our models on the fine-grained LOGIC task. Across the different classes, IBR performs best for eight out of thirteen classes and CL comes second, which is consistent with the overall results (cf. Table 7). While we do not observe a clear pattern in terms of the superiority of methods in terms of coarse-grained classes, we do observe that the classes with more data points (top rows in Table 15) are handled better by the CL model, showing that the CL model is able to reach its best performance when more data is available. This is somewhat counterintuitive, as we expect that CL can help the classes with more sparse data. However, we do observe that qualitatively CL has the best performance on the *Ad* classes: *False Causality*, *Ad Populum*, and *Ad Hominem*, indicating that the CL models are able to benefit from transferring knowledge within the same class from the coarse- to the fine-grained task. The fact that the two least populated classes are handled best by the methods KI and PBR indicates a potential for data-efficient reasoning with these methods.

Curiously, we do not see a significant improvement using any of the models on the *Equivocation* class. We attribute the consistent poor performance in this class to two important factors: (1) lack of training data: although we perform augmentation, this augmentation only modifies the original data slightly and does not add substantial variety to help our models understand this class better. (2) as *Equivocation* is the only fine-grained class that belongs to the broader class of *Ambiguity*, our models do not have enough data points to distinguish ambiguous arguments from arguments belonging to the other classes.

7. Discussion

Our evaluation shows that the methods perform relatively well across tasks and even on out-of-domain arguments, while further analysis shows that curriculum learning and data augmentation are promising components of a robust methodology for identifying logical fallacies in natural language (§6.3 & §6.2). While our methods rely heavily on language models, the additional components such as retrieval, attention-based mechanisms, and prototype networks, provide a consistent advantage of the models over

Table 15
F1 Scores per class for LOGIC test dataset using the models trained on the augmented train split with each class having 281 data points (The number of the data points shown for the training split in the table is before augmentation).

		F1-Scores						
fine-grained class	**coarse-grained class**	Baseline	NLI CL	IBR	ProtoTex	KI	# test	# train
Faulty Generalization	Defective Induction	0.656	0.614	**0.660**	0.612	0.549	60	281
Ad Hominem	Relevance	0.596	**0.633**	0.627	0.624	0.607	39	185
Ad Populum	Relevance	0.812	**0.844**	0.814	0.751	0.656	31	144
False Causality	Defective Induction	0.596	**0.727**	0.708	0.698	0.526	28	132
Circular Reasoning	Presumption	0.524	0.708	**0.719**	0.686	0.450	23	110
Appeal to Emotion	Relevance	0.426	0.473	**0.624**	0.445	0.300	23	109
Fallacy of Relevance	Relevance	0.512	0.436	**0.526**	0.374	0.286	22	102
Fallacy of Logic	Defective Induction	0.322	0.619	**0.622**	0.453	0.138	22	101
Intentional	Relevance	0.482	0.356	**0.500**	0.419	0.345	20	92
Fallacy of Credibility	Defective Induction	0.400	0.390	**0.486**	0.473	0.231	19	89
False Dilemma	Defective Induction	0.800	0.765	**0.824**	0.791	0.636	19	87
Fallacy of Extension	Relevance	0.482	0.629	0.541	0.598	**0.649**	18	80
Equivocation	Ambiguity	0.000	0.000	0.000	**0.065**	0.000	7	32

their corresponding baselines (§6.4). Looking closer at the retrieved exemplars in the IBR and PBR methods, we observe that they are often from the same class even when they are not syntactically similar to the input case (§6.5), which contributes to both the accuracy and the explainability of our models. Commonsense knowledge is also useful in particular cases, and potentially misleading in others, signifying the need for better grounding and path retrieval or generation. Looking at the performance per fallacy class, we observe that curriculum learning is able to benefit from knowledge transfer between *Ad* classes, while KI and PBR perform best in the most sparse classes.

This paper pursues robust and explainable methods for reasoning about fallacies in arguments, a task that is not only understudied but also vital to support critical thinking in an educational setting [109, 37]. Our study points to research paths that should be addressed in future work.

Further Innovation on Robust and Explainable Methods. We observe that our models are often unable to perform abstraction and comprehend the classes in a more general sense. This has been apparent from the mixed prototype of PBR (see §6.4.2), the mixed relevance of the examples of IBR (see table 13), and the occasionally confusing triples retrieved by KI (see table 14). We note, however, that detecting and classifying logical fallacies is a challenging task both for modern-day AI as well as for humans, as it requires a complex (and possibly ambiguous) combination of a wide range of knowledge, including an understanding of rhetorical structures and inclusion of background knowledge about affordances and symbolism of concepts [58]. We see two parallel streams of AI methods that should be explored in depth for logical fallacies. On the one hand, a promising new stream relies on neural language models through methods like chain-of-thought reasoning [123], self-rationalization [89], and prompt decomposition [28], coupled with large language models like GPT-3 [16] and Codex [19]. On the other hand, neuro-symbolic methods that, e.g., pose reasoning as a soft logic problem [24] may provide an alternative approach to generalizable reasoning. We invite future work to explore these directions, as well as their intersection, for the challenge of logical fallacy identification.

Focused Evaluation in Realistic and Open-Ended Settings. The task of logical fallacy identification, and even its related task of propaganda detection, has been introduced relatively recently in the field of AI. As such, not only the methods but also the evaluation settings for these tasks are limited at present. In this study, we take a broad perspective, starting from theories of logical fallacies from social science disciplines, and we provide a unified framework that can support a more comprehensive evaluation of fallacies. We plan to extend the evaluation datasets in this paper by further annotation of data for the remaining categories like *Begging the Question* and *Amphiboly* in Figure 1. Moreover, beyond identifying fallacies in the context of propaganda and misinformation, we also propose that logical fallacy identification should be considered in a broader set of use cases, such as forecasting [87], where detecting wrong or misleading arguments may be central to the judgment of the trustworthiness of predictions. It is also important to consider the relation of (formal) logical fallacies to boolean satisfiability (SAT) problems [79], which have been proven to be NP-complete.

Application and Misuse of This Work. Logical fallacies hold the promise to prevent the spread of propaganda, misinformation, and wrong argumentation among the very expansive content circulating daily on social media platforms. This could benefit both industry and governments, and ultimately ordinary social media users. However, strong logical fallacy identification models may also be misused to increase or enhance the diffusion of manipulative discourse [32, 32, 115]. We believe that analogously to the idea that encryption algorithms can be made robust if published and tested by the community [111], our social media systems and communication channels will become more resilient with the progress in developing methods and evaluation tasks for logical fallacy identification.

8. Conclusions

This paper presented an effort to consolidate social science work on logical fallacy organization into a formal framework that can be used to develop and evaluate AI methods. The framework consisted of three stages: fallacy detection, coarse-grained classification, and fine-grained classification. We designed a framework with three methods with native explainability and robustness: instance-based reasoning, prototype learning, and commonsense knowledge injection. To deal with the inherent data sparsity, we paired our methods with approaches for data augmentation and curriculum learning. Extensive experiments on in- and out-of-domain data showed that our methods have the ability to perform robustly across tasks, and retain much of their accuracy on out-of-domain evaluation. Curriculum learning was most helpful for coarse- and fine-grained evaluation, whereas data augmentation brought clear benefits for the most difficult task of fine-grained classification. We found that the explanation by the models in terms of known training instances or structured knowledge is easy to interpret, however, we noticed that the models still largely rely on surface form patterns and similarity in their reasoning. Guided by these insights, we proposed that future research should focus on further innovation in building robust and explainable methods, extending the evaluation to more realistic and open-ended settings, and facilitating open-source applications for social good while minimizing the possibility for misuse of the developed solutions.

Acknowledgements

The first five authors have been supported by armasuisse Science and Technology, Switzerland under contract No. 8003532866. Zhivar Sourati has been also partially supported by NSF under Contract No. IIS-2153546, while Filip Ilievski is sponsored in part by the DARPA MCS program under Contract No. N660011924033 with the US Office Of Naval Research.

References

[1] Aamodt, A., Plaza, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7, 39–59. URL: https://doi.org/10.3233/AIC-1994-7104, doi:10.3233/AIC-1994-7104. 1.

[2] Allcott, H., Gentzkow, M., Yu, C., 2019. Trends in the diffusion of misinformation on social media. Research & Politics 6, 2053168019848554.

[3] Almossawi, A., 2014. An illustrated book of bad arguments. The Experiment.

[4] Aristotle, 1989. On sophistical refutations: On Comin to be passing away - on the cosmos v. 3. Loeb Classical Library, LOEB, London, England.

[5] Arora, S., Wu, S., Liu, E., Re, C., 2022. Metadata shaping: A simple approach for knowledge-enhanced language models, in: Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland. pp. 1733–1745. URL: https://aclanthology.org/2022.findings-acl.137, doi:10.18653/v1/2022.findings-acl.137.

[6] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N., 2012. Abstract meaning representation (amr) 1.0 specification, in: Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL, pp. 1533–1544.

[7] Bardasz, T., Zeid, I., 1993. Dejavu: Case-based reasoning for mechanical design. AI EDAM 7, 111–124.

[8] Barker, S.F., 1965. The Elements of Logic. New York: Mcgraw-Hill.

[9] Barriere, V., Tafreshi, S., Sedoc, J., Alqahtani, S., 2022. WASSA 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories, in: Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, Association for Computational Linguistics, Dublin, Ireland. pp. 214–227. URL: https://aclanthology.org/2022.wassa-1.20, doi:10.18653/v1/2022.wassa-1.20.

[10] Barrón-Cedeno, A., Jaradat, I., Da San Martino, G., Nakov, P., 2019. Proppy: Organizing the news based on their propagandistic content. Information Processing & Management 56, 1849–1864.

[11] Bender, E.M., Koller, A., 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online. pp. 5185–5198. URL: https://aclanthology.org/2020.acl-main.463, doi:10.18653/v1/2020.acl-main.463.

[12] Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, Association for Computing Machinery, New York, NY, USA. p. 41–48. URL: https://doi.org/10.1145/1553374.1553380, doi:10.1145/1553374.1553380.

[13] Bernays, E., 2004. Propaganda. Ig Publishing, Brooklyn, NY.

[14] Biotechnology, Council, B.S.R., 2009. Past experience is invaluable for complex decision making, brain research shows. URL: https://www.sciencedaily.com/releases/2009/05/090513130930.htm.

[15] Bowman, S.R., Angeli, G., Potts, C., Manning, C.D., 2015. A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326 .

[16] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners. URL: https://arxiv.org/abs/2005.14165, doi:10.48550/ARXIV.2005.14165.

[17] Brüninghaus, S., Ashley, K.D., 2006. Progress in textual case-based reasoning: predicting the outcome of legal cases from text, in: AAAI, pp. 1577–1580.

[18] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., Elhadad, N., 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining .

[19] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.d.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al., 2021. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 .

[20] Chen, V., S. Savage, R., 2014. Evidence for a simplicity principle: teaching common complex grapheme-to-phonemes improves reading and motivation in at-risk readers. Journal of Research in Reading 37, 196–214. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9817.12022, doi:https://doi.org/10.1111/1467-9817.12022.

[21] Choi, E., Bahadori, M.T., Kulas, J.A., Schuetz, A., Stewart, W.F.,

Sun, J., 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. URL: https://arxiv.org/abs/1608.05745, doi:10.48550/ARXIV.1608.05745.

[22] Chu, D., 2021. Typeform/distilbert-base-uncased-mnli · hugging face. URL: https://huggingface.co/typeform/distilbert-base-uncased-mnli.

[23] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D., 2020a. ELECTRA: Pre-training text encoders as discriminators rather than generators, in: ICLR. URL: https://openreview.net/pdf?id=r1xMH1BtvB.

[24] Clark, P., Tafjord, O., Richardson, K., 2020b. Transformers as soft reasoners over language. URL: https://arxiv.org/abs/2002.05867, doi:10.48550/ARXIV.2002.05867.

[25] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., 2017. Supervised learning of universal sentence representations from natural language inference data, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark. pp. 670–680. URL: https://aclanthology.org/D17-1070, doi:10.18653/v1/D17-1070.

[26] Copi, I.M., 1954. Introduction to logic. Philosophy 29, 271–271.

[27] Council of the European Union, 2022. Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and amending Directive 2000/31/EC (Digital Services Act) (Text with EEA relevance). URL: http://data.europa.eu/eli/reg/2022/2065/oj. document 32022R2065. Accessed: 2022-11-30.

[28] Cui, L., Wu, Y., Liu, J., Yang, S., Zhang, Y., 2021. Template-based named entity recognition using BART, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online. pp. 1835–1845. URL: https://aclanthology.org/2021.findings-acl.161, doi:10.18653/v1/2021.findings-acl.161.

[29] Da San Martino, G., Yu, S., Barrón-Cedeno, A., Petrov, R., Nakov, P., 2019. Fine-grained analysis of propaganda in news article, in: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp. 5636–5646.

[30] Daelemans, W., van den Bosch, A., 2005. Memory-Based Language Processing. Studies in Natural Language Processing, Cambridge University Press. doi:10.1017/CBO9780511486579.

[31] Das, A., Gupta, C., Kovatchev, V., Lease, M., Li, J.J., 2022. ProtoTEx: Explaining model decisions with prototype tensors, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland. pp. 2986–2997. URL: https://aclanthology.org/2022.acl-long.213, doi:10.18653/v1/2022.acl-long.213.

[32] De Saussure, L., 2005. Manipulation and cognitive pragmatics. Manipulation and ideologies in the twentieth century , 113–145.

[33] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[34] Dimitrov, D., Bin Ali, B., Shaar, S., Alam, F., Silvestri, F., Firooz, H., Nakov, P., Da San Martino, G., 2021. Detecting propaganda techniques in memes, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online. pp. 6603–6617. URL: https://aclanthology.org/2021.acl-long.516, doi:10.18653/v1/2021.acl-long.516.

[35] Edunov, S., Ott, M., Auli, M., Grangier, D., 2018. Understanding back-translation at scale, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium. pp. 489–500. URL: https://aclanthology.org/D18-1045, doi:10.18653/v1/D18-1045.

[36] Elman, J.L., 1993. Learning and development in neural networks: the importance of starting small. Cognition 48, 71–99. URL: https://www.sciencedirect.com/science/article/pii/0010027793900584, doi:https://doi.org/10.1016/0010-0277(93)90058-4.

[37] Facione, P., 1990. Critical thinking: A statement of expert consensus for purposes of educational assessment and instruction (the delphi report) .

[38] Feldman, J., 2003. The simplicity principle in human concept learning. Current Directions in Psychological Science 12, 227–232. URL: https://doi.org/10.1046/j.0963-7214.2003.01267.x, doi:10.1046/j.0963-7214.2003.01267.x.

[39] Ferreira Cruz, A., Rocha, G., Lopes Cardoso, H., 2019. On sentence representations for propaganda detection: From handcrafted features to word embeddings, in: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Association for Computational Linguistics, Hong Kong, China. pp. 107–112. URL: https://aclanthology.org/D19-5015, doi:10.18653/v1/D19-5015.

[40] Ford, C., Kenny, E.M., Keane, M.T., 2020. Play mnist for me! user studies on the effects of post-hoc, example-based explanations &; error rates on debugging a deep learning, black-box classifier URL: https://arxiv.org/abs/2009.06349, doi:10.48550/ARXIV.2009.06349.

[41] Ganesh, B., Bright, J., 2020. Countering extremists on social media: Challenges for strategic communication and content moderation.

[42] Gao, T., Yao, X., Chen, D., 2021. Simcse: Simple contrastive learning of sentence embeddings. URL: https://arxiv.org/abs/2104.08821, doi:10.48550/ARXIV.2104.08821.

[43] Ge, M., Mao, R., Cambria, E., 2022. Explainable metaphor identification inspired by conceptual metaphor theory. Proceedings of the AAAI Conference on Artificial Intelligence 36, 10681–10689. URL: https://ojs.aaai.org/index.php/AAAI/article/view/21313, doi:10.1609/aaai.v36i10.21313.

[44] Ghazal, A., Rabl, T., Hu, M., Raab, F., Poess, M., Crolotte, A., Jacobsen, H.A., 2013. Bigbench: Towards an industry standard benchmark for big data analytics, in: Proceedings of the 2013 ACM SIGMOD international conference on Management of data, pp. 1197–1208.

[45] Gibbs, N.M., 2010. Formal and informal fallacies in anaesthesia. Anaesth Intensive Care 38, 639–646.

[46] Gibson, A., Rowe, G., Reed, C., 2007. A computational approach to identifying formal fallacy. CMNA VII-Computational Models of Natural Argument .

[47] Goffredo, P., Haddadan, S., Vorakitphan, V., Cabrio, E., Villata, S., 2022. Fallacious argument classification in political debates, in: Thirty-First International Joint Conference on Artificial Intelligence {IJCAI-22}, International Joint Conferences on Artificial Intelligence Organization. pp. 4143–4149.

[48] Goodwin, J., 1998. Forms of authority and the real ad verecundiam. Argumentation 12, 267–280. URL: https://doi.org/10.1023/a:1007756117287, doi:10.1023/a:1007756117287.

[49] Gundapu, S., Mamidi, R., 2022. Detection of propaganda techniques in visuo-lingual metaphor in memes. URL: https://arxiv.org/abs/2205.02937, doi:10.48550/ARXIV.2205.02937.

[50] Gupta, P., Saxena, K., Yaseen, U., Runkler, T., Schütze, H., 2019. Neural architectures for fine-grained propaganda detection in news. URL: https://arxiv.org/abs/1909.06162, doi:10.48550/ARXIV.1909.06162.

[51] Hamilton, K., 2021. Towards an ontology for propaganda detection in news articles, in: European Semantic Web Conference, Springer. pp. 230–241.

[52] Han, S., Mao, R., Cambria, E., 2022. Hierarchical attention network for explainable depression detection on twitter aided by metaphor concept mappings. URL: https://arxiv.org/abs/2209.07494, doi:10.48550/ARXIV.2209.07494.

[53] Hansen, H., 2020a. Fallacies, in: Zalta, E.N. (Ed.), The Stanford Encyclopedia of Philosophy. Summer 2020 ed.. Metaphysics Research Lab, Stanford University.

[54] Hansen, H., 2020b. Fallacies. URL: https://plato.stanford.edu/

entries/fallacies.

[55] He, H., 2021. Howey/electra-base-mnli · hugging face. URL: https://huggingface.co/howey/electra-base-mnli.

[56] Hendrycks, D., Gimpel, K., 2016. Gaussian error linear units (gelus). URL: https://arxiv.org/abs/1606.08415, doi:10.48550/ARXIV.1606.08415.

[57] Heo, J., Lee, H.B., Kim, S., Lee, J., Kim, K.J., Yang, E., Hwang, S.J., 2018. Uncertainty-aware attention for reliable interpretation and prediction, in: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2018/file/285e19f20beded7d215102b49d5c09a0-Paper.pdf.

[58] Hitchcock, D., 2017. Do the fallacies have a place in the teaching of reasoning skills or critical thinking?, in: On reasoning and argument. Springer, pp. 401–408.

[59] Ilievski, F., Oltramari, A., Ma, K., Zhang, B., McGuinness, D.L., Szekely, P., 2021a. Dimensions of commonsense knowledge. Knowledge-Based Systems 229, 107347.

[60] Ilievski, F., Szekely, P., Zhang, B., 2021b. Cskg: The commonsense knowledge graph, in: European Semantic Web Conference, Springer. pp. 680–696.

[61] Jiang, S., Metzger, M., Flanagin, A., Wilson, C., 2020. Modeling and measuring expressed (dis)belief in (mis)information. Proceedings of the International AAAI Conference on Web and Social Media 14, 315–326. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/7302.

[62] Jin, Z., Lalwani, A., Vaidhya, T., Shen, X., Ding, Y., Lyu, Z., Sachan, M., Mihalcea, R., Schölkopf, B., 2022. Logical fallacy detection. URL: https://arxiv.org/abs/2202.13758, doi:10.48550/ARXIV.2202.13758.

[63] Johansen, M.K., Kruschke, J.K., 2005. Category representation for classification and feature inference. J. Exp. Psychol. Learn. Mem. Cogn. 31, 1433–1458.

[64] Khan, I., 2021. Disinformation and freedom of opinion and expression. URL: https://documents-dds-ny.un.org/doc/UNDOC/GEN/G21/085/64/PDF/G2108564.pdf. undocs.org/en/A/HRC/47/25. Accessed: 2022-11-30.

[65] Kiesel, J., Mestre, M., Shukla, R., Vincent, E., Adineh, P., Corney, D., Stein, B., Potthast, M., 2019. SemEval-2019 task 4: Hyperpartisan news detection, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA. pp. 829–839. URL: https://aclanthology.org/S19-2145, doi:10.18653/v1/S19-2145.

[66] Lakkaraju, H., Bach, S.H., Leskovec, J., 2016. Interpretable decision sets: A joint framework for description and prediction, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. p. 1675–1684. URL: https://doi.org/10.1145/2939672.2939874, doi:10.1145/2939672.2939874.

[67] Lazer, D.M., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F., Metzger, M.J., Nyhan, B., Pennycook, G., Rothschild, D., et al., 2018. The science of fake news. Science 359, 1094–1096.

[68] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 .

[69] Li, O., Liu, H., Chen, C., Rudin, C., 2017. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. URL: https://arxiv.org/abs/1710.04806, doi:10.48550/ARXIV.1710.04806.

[70] Lin, B.Y., Chen, X., Chen, J., Ren, X., 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning, in: Proc. of EMNLP-IJCNLP, pp. 2829–2839.

[71] Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P., 2019a. K-bert: Enabling language representation with knowledge graph. URL: https://arxiv.org/abs/1909.07606, doi:10.48550/ARXIV.1909.07606.

[72] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019b. Roberta: A robustly optimized bert pretraining approach. URL: https://arxiv.org/abs/1907.11692, doi:10.48550/ARXIV.1907.11692.

[73] Locke, J., 1997. An Essay Concerning Human Understanding. Penguin classics, Penguin Classics, London, England.

[74] Luceri, L., Giordano, S., Ferrara, E., 2020. Detecting troll behavior via inverse reinforcement learning: A case study of russian trolls in the 2016 us election. Proceedings of the International AAAI Conference on Web and Social Media 14, 417–427. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/7311.

[75] Ma, E., 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

[76] Ma, K., Francis, J., Lu, Q., Nyberg, E., Oltramari, A., 2019. Towards generalizable neuro-symbolic systems for commonsense question answering, in: Proc. of the First Workshop on Commonsense Inference in Natural Language Processing, pp. 22–32.

[77] Ma, K., Ilievski, F., Francis, J., Bisk, Y., Nyberg, E., Oltramari, A., 2021a. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering, in: AAAI.

[78] Ma, K., Ilievski, F., Francis, J., Ozaki, S., Nyberg, E., Oltramari, A., 2021b. Exploring strategies for generalizable commonsense reasoning with pre-trained models. URL: https://arxiv.org/abs/2109.02837, doi:10.48550/ARXIV.2109.02837.

[79] Marques-Silva, J., 2008. Practical applications of boolean satisfiability .

[80] Martin, N.G., 1989. Proceedings of the second international conference on quantitative genetics, edited by b.s. weir, e.j. eisen, m.m. goodman, and g. namkoong, sunderland, ma: Sinauer associates inc., 1988, xii + 724 pages, $60.00 (cloth), $38.50 (paper). Genetic Epidemiology 6, 389–390. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/gepi.1370060208, doi:https://doi.org/10.1002/gepi.1370060208.

[81] Martino, G.D.S., Cresci, S., Barron-Cedeno, A., Yu, S., Di Pietro, R., Nakov, P., 2020. A survey on computational propaganda detection URL: https://arxiv.org/abs/2007.08024, doi:10.48550/ARXIV.2007.08024.

[82] Medin, D.L., Schaffer, M.M., 1978. Context theory of classification learning. Psychol. Rev. 85, 207–238.

[83] Miller, G.A., 1995. Wordnet: a lexical database for english. Communications of the ACM 38, 39–41.

[84] Mitra, A., Banerjee, P., Pal, K.K., Mishra, S., Baral, C., 2019. Exploring ways to incorporate additional knowledge to improve natural language commonsense question answering. arXiv preprint arXiv:1909.08855 .

[85] Morris, J., Lifland, E., Yoo, J.Y., Qi, Y., 2020. Textattack/bert-base-uncased-mnli · hugging face. URL: https://huggingface.co/textattack/bert-base-uncased-MNLI.

[86] Morrow, G., Swire-Thompson, B., Polny, J.M., Kopec, M., Wihbey, J.P., 2022. The emerging science of content labeling: Contextualizing social media content moderation. Journal of the Association for Information Science and Technology 73, 1365–1386.

[87] Morstatter, F., Galstyan, A., Satyukov, G., Benjamin, D., Abeliuk, A., Mirtaheri, M., Hossain, K.T., Szekely, P.A., Ferrara, E., Matsui, A., et al., 2019. Sage: A hybrid geopolitical event forecasting system., in: IJCAI, pp. 6557–6559.

[88] Nakpih, C.I., Santini, S., 2020. Automated discovery of logical fallacies in legal argumentation. International Journal of Artificial Intelligence & Applications 11, 37–48. URL: https://doi.org/10.5121%2Fijaia.2020.11203, doi:10.5121/ijaia.2020.11203.

[89] Narang, S., Raffel, C., Lee, K., Roberts, A., Fiedel, N., Malkan, K., 2020. Wt5?! training text-to-text models to explain their predictions. URL: https://arxiv.org/abs/2004.14546, doi:10.48550/ARXIV.2004.14546.

[90] Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., Edunov, S., 2019. Facebook fair's wmt19 news translation task submission, in: Proc. of WMT.

[91] Oliinyk, V.A., Vysotska, V., Burov, Y., Mykich, K., Fernandes, V.B., 2020. Propaganda detection in text data based on nlp and machine learning., in: MoMLeT+ DS, pp. 132–144.

[92] Oyelade, O.N., Ezugwu, A.E., 2020. A case-based reasoning framework for early detection and diagnosis of novel coronavirus. Informatics in Medicine Unlocked 20, 100395. URL: https://www.sciencedirect.com/science/article/pii/S2352914820303683, doi:https://doi.org/10.1016/j.imu.2020.100395.

[93] Pantazi, S.V., Arocha, J.F., Moehr, J.R., 2004. Case-based medical informatics. BMC Medical Informatics and Decision Making 4, 1–23.

[94] Paraschiv, A., Cercel, D.C., Dascalu, M., 2020. Upb at semeval-2020 task 11: Propaganda detection with domain-specific trained bert. URL: https://arxiv.org/abs/2009.05289, doi:10.48550/ARXIV.2009.05289.

[95] Peters, M.E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., Smith, N.A., 2019. Knowledge enhanced contextual word representations, in: Proc. of EMNLP-IJCNLP, pp. 43–54.

[96] QIN, X., REGLI, W.C., 2003. A study in applying case-based reasoning to engineering design: Mechanical bearing design. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 17, 235–252. doi:10.1017/S0890060403173064.

[97] Raaheim, K., 1965. Problem solving and past experience. Monographs of the Society for Research in Child Development 30, 58–67. URL: http://www.jstor.org/stable/1165776.

[98] Reimers, N., 2021a. Cross-encoder/nli-deberta-base · hugging face. URL: https://huggingface.co/cross-encoder/nli-deberta-base.

[99] Reimers, N., 2021b. Cross-encoder/nli-roberta-base · hugging face. URL: https://huggingface.co/cross-encoder/nli-roberta-base.

[100] Reimers, N., Gurevych, I., 2019. Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics. URL: http://arxiv.org/abs/1908.10084.

[101] Renkl, A., 2014. Toward an instructionally oriented theory of example-based learning. Cognitive Science 38, 1–37. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12086, doi:https://doi.org/10.1111/cogs.12086.

[102] Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "why should i trust you?": Explaining the predictions of any classifier. URL: https://arxiv.org/abs/1602.04938, doi:10.48550/ARXIV.1602.04938.

[103] Rohde, D.L., Plaut, D.C., 1999. Language acquisition in the absence of explicit negative evidence: how important is starting small? Cognition 72, 67–109. URL: https://www.sciencedirect.com/science/article/pii/S0010027799000311, doi:https://doi.org/10.1016/S0010-0277(99)00031-1.

[104] Rohde, D.L., Plaut, D.C., 2004. Less is less in language acquisition, in: Connectionist models of development. Psychology Press, pp. 178–218.

[105] Rosch, E.H., 1973. Natural categories. Cognitive Psychology 4, 328–350. URL: https://www.sciencedirect.com/science/article/pii/0010028573900170, doi:https://doi.org/10.1016/0010-0285(73)90017-0.

[106] Saha, S., Yadav, P., Bauer, L., Bansal, M., 2021. ExplaGraphs: An explanation graph generation task for structured commonsense reasoning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic. pp. 7716–7740. URL: https://aclanthology.org/2021.emnlp-main.609, doi:10.18653/v1/2021.emnlp-main.609.

[107] Sahai, S., Balalau, O., Horincar, R., 2021. Breaking down the invisible wall of informal fallacies in online discussions, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online. pp. 644–657. URL: https://aclanthology.org/2021.acl-long.53, doi:10.18653/v1/2021.acl-long.53.

[108] Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N.A., Choi, Y., 2019. Atomic: An atlas of machine commonsense for if-then reasoning, in: Proceedings of the AAAI conference on artificial intelligence, pp. 3027–3035.

[109] Scheffer, B.K., Rubenfeld, M.G., 2000. A consensus statement on critical thinking in nursing. Journal of Nursing Education 39, 352–359. URL: https://journals.healio.com/doi/abs/10.3928/0148-4834-20001101-06, doi:10.3928/0148-4834-20001101-06.

[110] Sennrich, R., Haddow, B., Birch, A., 2016. Improving neural machine translation models with monolingual data, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany. pp. 86–96. URL: https://aclanthology.org/P16-1009, doi:10.18653/v1/P16-1009.

[111] Shannon, C.E., 1949. Communication theory of secrecy systems. The Bell system technical journal 28, 656–715.

[112] Sourati, Z., Ilievski, F., Sandlin, H.A., Mermoud, A., 2023. Case-based reasoning with language models for classification of logical fallacies.

[113] Speer, R., Chin, J., Havasi, C., 2017. Conceptnet 5.5: An open multilingual graph of general knowledge, in: Thirty-first AAAI conference on artificial intelligence.

[114] Spensberger, F., Kollar, I., Pankofer, S., 2022. Effects of worked examples and external scripts on fallacy recognition skills: a randomized controlled trial. Journal of Social Work Education 58, 622–639.

[115] Tymbay, A.A., 2022. Manipulative use of political headlines in western and russian online sources. Discourse & Communication 16, 346–363. URL: https://doi.org/10.1177/17504813221101824, doi:10.1177/17504813221101824.

[116] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. URL: https://arxiv.org/abs/1706.03762, doi:10.48550/ARXIV.1706.03762.

[117] Vorakitphan, V., Cabrio, E., Villata, S., 2021. "don't discuss": Investigating semantic and argumentative features for supervised propagandist message detection and classification, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), INCOMA Ltd., Held Online. pp. 1498–1507. URL: https://aclanthology.org/2021.ranlp-1.168.

[118] Walia, H., Rana, A., Kansal, V., 2019. Case based interpretation model for word sense disambiguation in gurmukhi, in: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 359–364. doi:10.1109/CONFLUENCE.2019.8776909.

[119] Wang, P., Ilievski, F., Chen, M., Ren, X., 2021. Do language models perform generalizable commonsense inference? URL: https://arxiv.org/abs/2106.11533, doi:10.48550/ARXIV.2106.11533.

[120] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M., 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. arXiv:2002.10957.

[121] Wang, Y., McKee, M., Torbica, A., Stuckler, D., 2019. Systematic literature review on the spread of health-related misinformation on social media. Social Science & Medicine 240, 112552. URL: https://www.sciencedirect.com/science/article/pii/S0277953619305465, doi:https://doi.org/10.1016/j.socscimed.2019.112552.

[122] Watts, I., 2021. Logic. Soli Deo Gloria Publications, Morgan, PA.

[123] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D., 2022. Chain of thought prompting elicits reasoning in large language models. URL: https://arxiv.org/abs/2201.11903, doi:10.48550/ARXIV.2201.11903.

[124] Williams, A., Nangia, N., Bowman, S., 2018. A broad-coverage challenge corpus for sentence understanding through inference, in:

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics. pp. 1112–1122. URL: http://aclweb.org/anthology/N18-1101.

[125] Wu, L., Morstatter, F., Carley, K.M., Liu, H., 2019. Misinformation in social media: Definition, manipulation, and detection. SIGKDD Explor. Newsl. 21, 80–90. URL: https://doi.org/10.1145/3373464.3373475, doi:10.1145/3373464.3373475.

[126] Yaskorska, O., Budzynska, K., Kacprzak, M., 2013. Proving propositional tautologies in a natural dialogue. Fundamenta Informaticae 128, 239–253. URL: https://doi.org/10.3233/FI-2013-944, doi:10.3233/FI-2013-944. 1-2.

[127] Yoosuf, S., Yang, Y., 2019. Fine-grained propaganda detection with fine-tuned BERT, in: Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Association for Computational Linguistics, Hong Kong, China. pp. 87–91. URL: https://aclanthology.org/D19-5011, doi:10.18653/v1/D19-5011.

[128] Yu, S., Martino, G.D.S., Mohtarami, M., Glass, J., Nakov, P., 2021. Interpretable propaganda detection in news articles URL: https://arxiv.org/abs/2108.12802, doi:10.48550/ARXIV.2108.12802.

[129] Zheng, S., Song, Y., Leung, T., Goodfellow, I., 2016. Improving the robustness of deep neural networks via stability training. URL: https://arxiv.org/abs/1604.04326, doi:10.48550/ARXIV.1604.04326.

[130] Zhong, W., Tang, D., Duan, N., Zhou, M., Wang, J., Yin, J., 2019. Improving question answering by commonsense-based pre-training, in: Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I, Springer-Verlag, Berlin, Heidelberg. p. 16–28. URL: https://doi.org/10.1007/978-3-030-32233-5_2, doi:10.1007/978-3-030-32233-5_2.