Improving Trust in Fact-Checking Systems with Synthetic Training Data and Explanations

Jean-Flavien Bussotti Pitollet

Improving Trust in Fact-Checking Systems with Synthetic Training Data and Explanations

JAN. 2022 - DEC. 2024

Jean-Flavien Bussotti Pitollet

EURECOM | Data Science Department

December 5th, 2024

PhD Supervisor:
Paolo Papotti

Jury:
Elena Cabrio
Davide Martinenghi
Pietro Michiardi (Jury president)

Abstract

English

In the era of social networks, the propagation of fake news is increasingly frequent. In the last decade, numerous events were affected by misinformation. Whether we think of the last US election, or the COVID crisis, there is little doubt about the influence of such dishonest communication. Therefore, organisms do their best to counter them. Their efforts aim at reducing fake news include fact-checking claims published on social medias.

Traditional methods rely on humans to manually annotate each claim, for example in a politician's speech. However, this method is not adaptable to the vast amounts of data found on social media, nor to crisis times when there can be sudden increases in social media posts.

Therefore, a preferred solution is computational fact checking. In this domain, the best solutions are Machine Learning based. Usually, given a claim, a model retrieves relevant evidence from a corpus. Then, a predictor model assesses the veracity of the claim based on the evidence. In this thesis, we investigate how to support information mitigators, such as fact-checkers, in their work. We focus on two main aspects that are clear bottlenecks for the adoption of computational solutions in practice. The first aspect is the manual effort to bootstrap and refine the systems. The second aspect is to make systems more interpretable.

Bootstrapping a fact-checking system, with a supervised learning approach, requires a lot of training data. As manual annotation is expensive and slow, it is not adaptable to new domains, neither to crisis situations. In such context, governments or companies cannot afford to wait until a new training set is manually defined, as this may be too long. The solution we propose is to generate such datasets. Synthetic datasets are increasingly popular for model fine-tuning or training. In fact-checking, little research has been done on synthetic datasets construction from tabular evidence. In our work, TENET, we propose to handle this issue by generating claims from structured evidence sources. In UNOWN, we switch to the multi-modal setting to perform claim generation using both the tabular and the textual modalities. We provide generated examples as training data to fact-checking systems. The systems trained with our data showcase a label-prediction accuracy comparable to the same system trained with human-created training examples.

Another matter with computational fact-checking is explainability. Fact-checking systems are explainable when users can understand their decisions thanks to a justification. Given an input claim, most state of the art models predict a label without any justification on how they reached their conclusion. Users, in a misinformation context, can be eager to know why the text they are reading is labelled as possibly misleading or even incorrect. In this scenario, current black-box models fails short. In our work, we leverage two state of the art approaches to interpret decision taken by deep learning models. We show that such techniques allow to identify relevant evidence with high accuracy.

This thesis thoroughly explore Synthetic Dataset generation and Explainability of Fact-checking systems. The solutions we propose ease the building of supervised models in low resource domains and ease the understanding of predictions by the users.

French

À l'ère des réseaux sociaux, la propagation de fausses informations est de plus en plus fréquente. Au cours de la dernière décennie, de nombreux événements ont été affectés par la désinformation. Par conséquent, les organismes font de leur mieux pour les contrer. Leurs efforts visent à réduire la quantité de fausses informations en vérifiant les faits des affirmations publiées sur les réseaux sociaux. Les méthodes traditionnelles reposent sur des humains pour annoter manuellement chaque affirmation. Cependant, cette méthode n'est pas adaptable aux vastes quantités de données présentes sur les réseaux sociaux, ni aux périodes de crise où il peut y avoir une augmentation soudaine des publications. Par conséquent, une solution privilégiée est la vérification automatique de faits. Dans ce domaine, les meilleures solutions sont basées sur l'apprentissage automatique. En général, partant d'une affirmation, un modèle trouve des preuves pertinentes dans un corpus. Ensuite, un modèle prédictif évalue la véracité de l'affirmation sur la base des preuves. Dans cette thèse, nous étudions comment soutenir les fact-checkers dans leur travail. Nous nous concentrons sur deux aspects principaux qui constituent des goulots d'étranglement pour l'adoption des solutions automatisées en pratique. Le premier aspect est l'effort manuel nécessaire pour créer un système. Le deuxième aspect est de rendre les systèmes plus interprétables. Créer un système de vérification des faits, avec une approche d'apprentissage supervisé, nécessite beaucoup de données d'entraînement. Comme l'annotation manuelle est coûteuse et lente, elle n'est pas adaptable aux nouveaux domaines, ni aux situations de crise. Dans un tel contexte, les gouvernements ou les entreprises ne peuvent se permettre d'attendre qu'un nouveau jeu de données d'entraînement soit défini manuellement, car cela pourrait prendre trop de temps. La solution que nous proposons est de générer de tels jeux de données. Les jeux de données synthétiques sont de plus en plus populaires pour le fine-tuning ou l'entraînement des modèles. Dans la vérification des faits, peu de recherches ont été faites sur la construction de jeux de données synthétiques à partir de preuves tabulaires. Avec notre système TENET, nous proposons de traiter ce problème en générant des affirmations à partir de sources de preuves structurées. Avec UNOWN, nous passons à un cadre multi-modal pour générer des affirmations en utilisant à la fois des données structurées et non structurées. Nous entrainons les systèmes de vérification des faits à partir d'exemples générés. Les systèmes entraînés avec nos données démontrent une capacité de prédiction comparable à celle d'un même système entraîné avec des exemples écrits par des humains. Un autre problème de la vérification automatique de faits est l'explicabilité. Les systèmes de vérification des faits sont explicables lorsque les utilisateurs peuvent comprendre leurs décisions grâce à une justification. À partir d'une affirmation, les modèles de l'état de l'art prédisent une étiquette pour un texte, sans aucune justification sur la manière dont ils sont parvenus à leur conclusion. Les utilisateurs, dans un contexte de désinformation, peuvent être désireux de savoir pourquoi le texte qu'ils lisent est étiqueté comme « faux ». Dans ce scénario, les modèles actuels de type « boîte noire » échouent. Dans notre travail, nous exploitons deux approches d'explicabilité (xAI) de pointe pour interpréter les décisions prises par les modèles. Nous montrons que les techniques d'explicabilité permettent d'identifier des preuves pertinentes avec une grande précision. Cette thèse explore en profondeur la génération de jeux de données synthétiques et l'explicabilité des systèmes de vérification de faits. Les systèmes que nous proposons facilitent la construction de modèles supervisés dans des domaines à faibles ressources, et facilitent la compréhension des prédictions par les utilisateurs.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Over the last few decades, social networks have grown from being a small part of our lives to being a major time catcher. Studies found out that people under 24 can spend up to 6 hours a day on social networks [1]. Therefore, these platforms shape, or at least influence, users' opinions on popular topics - especially when claims or theories are repeatedly shared by numerous users. So far, these behaviors are questionable, as they tend to lead users to become more narrow-minded, but they are not alarming. Adding fake news to the equation changes the narrative. Misinformation, and its motivated counterpart disinformation, threatens society. Whether considering the COVID crisis, the US elections, or even the bedbug crisis in France [2], the diffusion and influence of such misinformation have disrupted elections or government policies. Consequently, the World Economic Forum placed misinformation as the top risk for 2024 [3]. It is now more important than ever for social media platforms to mitigate this issue.

On fact-checking websites such as Snopes[1] and PolitiFact[2], journalists perform fact-checking by manually collecting claims and verifying them. Along the decision to classify or not a claim as misinformation, they provide external relevant documents justifying their choice. While this is a technically straightforward solution, this task is time-consuming and hard to scale to a large number of claims.

Events such as the COVID crisis caused an explosion in the number of claims to check [4]. As manual fact-checking is expensive and requires time from human annotator to check each claim, computationally based fact-checking solutions became popular [5]. ClaimBuster [6] is the first tentative to find check-worthy claims and automatically provide labels for them. In their work, they output a label coming from a curated datasets of already fact-checked claims. New methods to fact-check claims without a direct comparison with existing claims appeared driven by the emergence of transformer-based models and their text understanding capabilities [7].

Different approaches have been proposed according to the evolution of the underlying deep learning architectures. A first wave of systems have been built leveraging Pretrained Language Models (PLM) – usually encoder only – with fine-tuning. Fine-tuning of language models [8] consists in adapting a pretrained model to a specific downstream task by further training it on a task-specific dataset. The technique relies on supervised learning; thus the training requires a labeled dataset t. Such dataset is built from a corpus C of documents d. Each example from the dataset is a triple of (i) textual claim c, (ii) k evidence piece such as $e = \{e_j| \quad \forall j \in \{0, k\}\}$, and (iii) a label l. In other words, the example of index i is $x_i = (c, \{e_0, \ldots, e_k\}, l)$. Figure 1.1 presents an annotated example from a fact-checking dataset crafted by humans from Wikipedia pages. In the example, the Claim c 'Rome, located in Italy, is famous for the Coliseum, while Nice is a seaside City in France.' is joint with evidence e, which contain both textual evidence, 'Coliseum is in Rome','Nice is a seaside City', and tabular evidence, i.e., the 'Nice', 'France', 'Rome', 'Italy' cell values. All evidence appear in documents inside the corpus C. Along the claim and evidence, the human annotator provides a label l for the example. This dataset of annotated

[1] https://www.snopes.com
[2] https://www.politifact.com

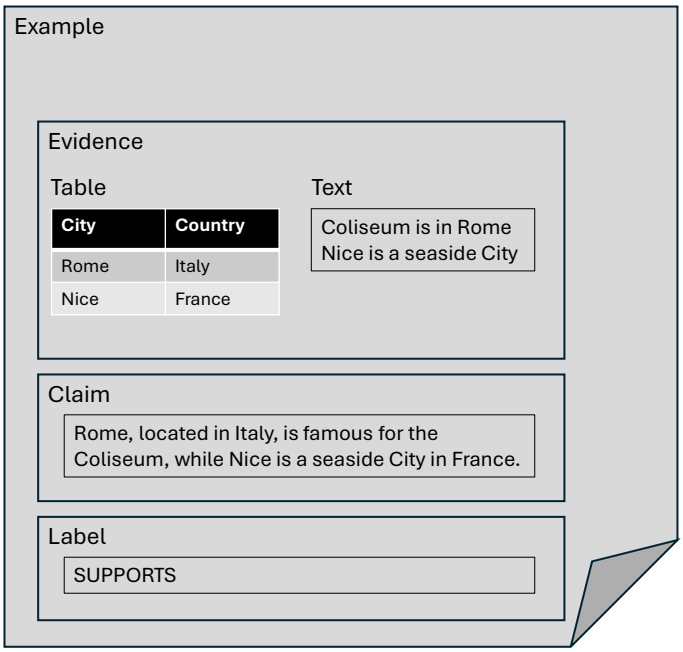

Figure 1.1: Example from a fact-checking dataset

examples (claims, evidence and label) is then used to train a model.

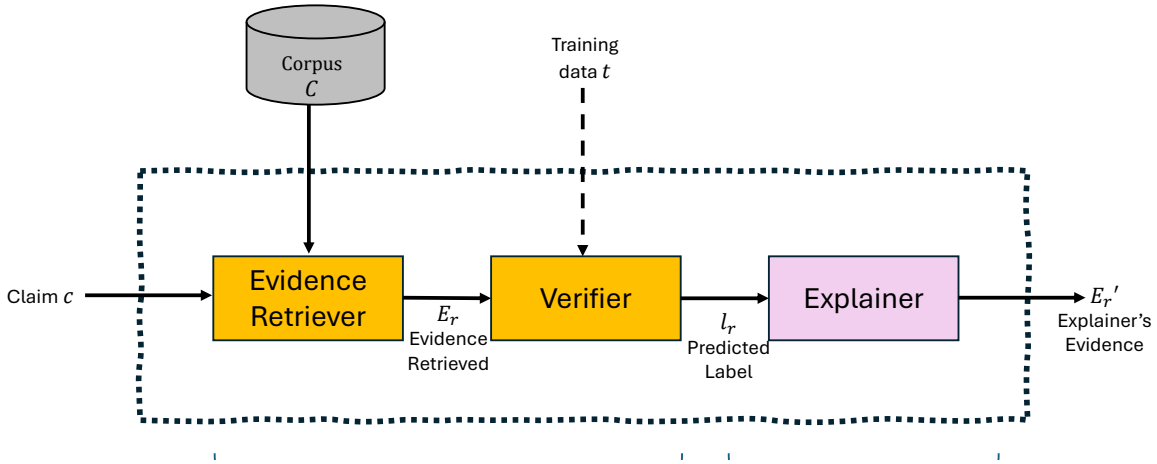

Figure 1.2: Pipeline for fact-checking inference. From the input claim c, the retriever uses C to provide verifier evidence E_r. From this information, the verifier provides a veracity label l_r. The explainer uses outputs from the latter to give a subset of useful evidence E'_r.

Once the models have been trained, they are used to verify unseen claims. Figure 1.2 presents a standard fact-checking pipeline at inference time, where a textual claim and a reference corpus of documents are given as input. The pipeline presents two main models: an Evidence Retriever and a Verifier, both represented with a yellow box in the Figure. The retriever, given the claim c and the corpus C with evidence, outputs the claim-relevant evidence E_r. The verifier uses the retrieved evidence E_r and the claim c to predict a verdict l_r. Originally, the verifier has been fine-tuned using the training data t. Optionally, the last step is an explainer model that outputs a more precise selection of evidence E'_r, restraining E_r only to the evidence that were used by the verifier for taking a decision.

	FEVEROUS			ADS	
Method.	F1 Refutes	F1 Supports		F1 Refutes	F1 Supports
FT-Bert	0.66	0.88		0.47	0.96
LLM-Prompt	0.72	0.86		0.04	0.95

Table 1.1: Performance of fine-tuned (FT) methods versus a Large Language Model (LLM) for fact-checking two datasets. The LLM used is ChatGPT 3.5 Turbo, the FT-Bert is Roberta Large for FEVEROUS and CT-Bert for ADS. FEVEROUS' claims are general knowledge information from Wikipedia, while Ads contain social media's advertisements from Facebook.

The second wave of systems rely on the more recent Large Language Models (LLMs), such as ChatGPT [9] or LLaMA 3 [10]. Initial approaches use the instructed LLM as is, without any further training. This raises the question if an out-of-the-box LLM be an effective fact-checker. Table 1.1 reports results on the well-known fact-checking FEVEROUS dataset, where given the claim and evidence as input, we can compare the predicted label output with the original, gold label. In this setting, where evidence are based on Wikipedia's general knowledge, we see that a not fine-tuned LLM can match the result quality of a fine-tuned model, both for claims labeled as Supported and for those labeled as Refuted. However, things change radically with more domain specific claims. The ADS dataset contains ads published on Facebook, containing claims that cannot be debunked easily with traditional common knowledge [11]. As in this dataset, 88% of the claims have a gold label 'Supports', we report the F1 score of the two labels to take into consideration the skew. The results are lower for both approaches for the 'Refutes' label. However, the non fine-tuned LLM reports very low results in detecting misinformation. Even providing the model with examples in the prompt, a *few shots* technique to counter-balance the lack of fine-tuning [12], does not help improving the prediction quality. These results justify the importance of a fine-tuned model, especially in settings relying on specific domain data. Other studies also report that while LLMs are a good fit for generative tasks such as decomposition of claims, they struggle on predicting a label [13]. Moreover, as LLMs require powerful hardware [14], a fine-tuned smaller model, such as Bert, is preferable as it is cheaper to run and consumes less energy [15].

1.2 Problems

Given an annotated dataset for fine tuning, we can obtain a high-quality fact-checking model for that domain. However, the major drawback is the time needed to build such dataset: writing up one of the most used fact-checking training corpus by a single annotator would take 183 full workdays [7]. As this is expensive and time consuming, money and time can become a bottleneck. Moreover, the dataset would cover only one domain. One may need data to perform fact-checking on claims for a new domain, such as medical or legal [16], but the original model would perform poorly as we saw for the ADS dataset. Most fact-checking corpora contain claims based on Wikipedia facts, but domain shift is an important problem for ML in general [17, 18]. More precisely, a model trained on a certain type of data would struggle to generalize, and would benefit from a training on the target-task data [19]. In other terms, a model trained to label claims similar to "Barack Obama was born in the US" might struggle on the claim "Covid cases went up by 10% since the beginning of 2024 in Germany.". In certain circumstances, especially in crisis times, a new model built using target-task data needs to be ready in short time. Therefore, manual writing of a train set, while feasible, is not desirable. All these reasons motivate the need for an easier and more accessible way to create datasets with training examples.

Apart from the data generation's problem, output explainability is another issue to deal with. Indeed, a fact-checking model outputs a label given a claim. However, most models are built on deep learning architectures, such as transformers. Those architecture are not explainable by nature: we could say that they act as black-boxes [20, 7, 21, 22]. More precisely, it is not feasible to know which evidence was useful to the fact-checking model to take its decision. It is not either possible to know if an evidence

piece pushed the decision toward 'Supports' or 'Refutes', simply by checking the output or the activated neurons [23, 24]. Fact-checking users in general want to know what are the reasons behind a decision, especially for a 'Refutes' label that mark as misinformation some content. A solution could be to provide the retrieved evidence E_r. However, retrievers often struggle to retrieve accurate evidence due to the complexity of indexing vast, diverse data sources [25]. As predictors models are robust to noise [26], increasing the overall amount of retrieved evidence used as input for the predictor is a solution to obtain accurate prediction. As we want to provide the predictor model as many useful evidence as possible to make a decision, E_r can become large (e.g., 20 to 40 sentences or cell values) and therefore too noisy to be useful as explanation source for the user. A system providing a concise and complete justification is thus needed.

In this thesis, we tackle both problems by answering the following research questions.

1 Can we automate the generation of domain-specific training data for a predictor model in a multi-modal setting?

2 Can we explain the output of any black-box fact-checking predictor model? Can we distinguish evidence used by the fact-checking model from unnecessary ones?

1.3 A solution to expensive dataset construction?

Given how expensive are human annotations, we aim to find alternative techniques to build a training dataset. In this section, we discuss research question 1. After presenting the problem, we jump on its challenges and briefly present our solution.

1.3.1 Problem

Creation of examples to fine-tune a model is an expensive task. It is also time-consuming and prone to error. To reduce error numbers, some efforts aim at lowering human error rate in annotation [27]. We explore a different path. To simultaneously mitigate the problems of domain-specificity and cost, a solution is to synthetically create a dataset. This method is inspired by the *data driven AI* principle, where we improve the performance of a solution by focusing on improving the data rather than changing the model [28]. We deal of data generation first restraining to tabular data only, before widening to a multi-modal setting.

Our technique is adaptable to multiple tasks, but we focus on the generation for the Natural Language Inference (NLI) task of fact-checking [29]. The datasets generated by our technique finally serve to train a fact-checking model.

From structured data In this setting, we aim to construct a claim based on tabular evidence as in Figure 1.3. The claim here can be generated and verified by using the four evidence cells provided.

There are several motivations to use tabular data as evidence. Usually, textual data is used to convey information in a narrative form. On the other hand, tabular data provides information in a form more adapted to analytics and operations.

One reason is that most existing annotated datasets use a whole table as evidence for a claim[30, 21]. However, the more advanced datasets report evidence at a cell level [7] and this is the default granularity for effective applications, such as fact-checking, where tables can be very large [31].

Another concern is the use of mathematical expressions, such as max, min, and count, or comparison across values, in real claims. Existing annotated datasets mostly contain claims that can be checked with simple lookup. For instance, in ToTTo [32], a table to text dataset, 80% of claims are relying on text without any mathematical expressions. Existing real problems are beyond what is currently covered in annotated datasets. As we saw that un-adapted training data leads to a model performing poorly during test, we need new datasets providing more complex claims.

Figure 1.3: Example of a claim relying on structured data as evidence

From multi-modal inputs Some scenarios require to fact-check content showing a combination of images and texts [33, 34] as evidence. Others, such as Feverous [7], combine texts and tables, as in Figure 1.4. The claim can only be labelled by using all the modalities: the table provides the city location information, while the text precise the famous spots of each city. Notice that while the evidence is identical to the example in Figure 1.1, the claims are different.

If low-resource domains already lack existing datasets in a unimodal setting, moving to the multi-modal setting only makes it worse. Multi-modality datasets are scarce even for general knowledge domains. New datasets are thus needed to tackle this issue. But those datasets are hard to craft, they bring a variety of challenges that we discuss next.

1.3.2 Challenges

The first challenge is correctness The credibility of a veracity label comes from the reputation of the fact-checking source [35]. Therefore, it is important for a fact-checking tool to be precise in predicting labels, as being approximate can lead to an untrusted method. As a consequence, we need correctly structured examples—in terms of natural language, the evidence used, and the label—to develop effective fact-checking models.

Diversity in the training data is another key challenge. Diversity can be analyzed from multiple angles, we focus on two. The first is the diversity in the different patterns of selected evidence. A naive way to select evidence is to consider only the first sentence in an article, or the last row in a table. Evidence selection across examples in the dataset should cover a large amount of different patterns. The second angle is the diversity in the description pattern in natural language. The claims constructed, even for the same evidence, must be as various as possible - the claims must not just rephrase or summarize the evidence, but perform operations such as count or max on them. This challenge is present both in selecting cells in a uni-modal setting, and in combining sentences with cells in the multi-modal context.

Scalability is a third challenge. A claim generation system should involve as little human intervention as possible. As human cost is one of the main issues we tackle, a synthetic dataset generation system can afford to use some human annotations, but their number should be as limited as possible.

An unavoidable challenge is the generation of 'Refutes' claims. The training of a fact-checking model

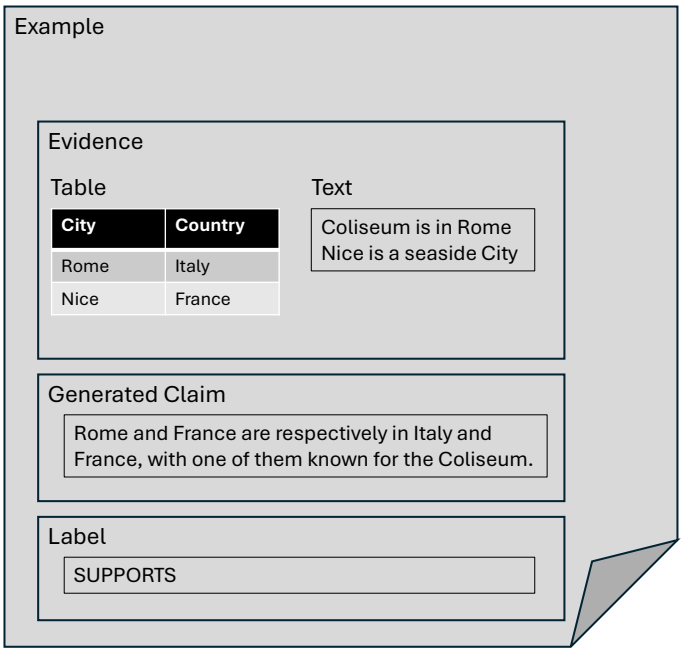

Figure 1.4: A claim relying on both unstructured and structured data

requires such crucial examples in the training set. These examples are crucial for training a robust model, as they help delineate misleading claim from supported ones. Therefore, an example generation system should create both supported and refuted claims. However, crafting high-quality refuting claims is not trivial: these refuted claims must cover the many possible ways to articulate falsehoods. In a tabular setting, one can leverage basic table operations to create claims that are in contradiction with the data evidence. In a unstructured data context, one cannot use the same strategy and needs to find alternative techniques.

Finally, multi-modality is a last challenge. Depending on the task, the system should be able to use as input text and/or other modalities, such as tables. A generation model thus needs to deal with multi-modals inputs. It also needs to be able to merge multiple pieces of evidence, independently of their type, into a single claim.

1.3.3 Solutions

To tackle the cost of building the training data t, we develop example generation methods. We start with a structured evidence setting before moving to the multi-modal setting.

We present the overall generation pipeline for both approaches in Figure 1.5. From the corpus C, the *Example generation* module produces a list of 'Supports' and 'Refutes' claims, each of them relying on different pieces of evidence and pages from the corpus. This list of claims, along the their evidence and the label, will serve as training data t to to the verifier.

From structured data To build a synthetic dataset, a first step is to select the evidence to base the claim upon. Keeping in mind the challenges mentioned above, the quality of evidence selection is crucial and cannot be neglected. Therefore, we select cells mimicking the human patterns found in a *gold* training set, when available. As the evidence is tabular, a natural approach is to use SQL queries. Our solution infers patterns from existing human-crafted example and derives SQL queries from them to get more data evidence. As we need 'Refutes' claims, we perform error injection to create inconsistent tables for such claims. The original, clean table is the evidence in the example, while the erroneous one is used to

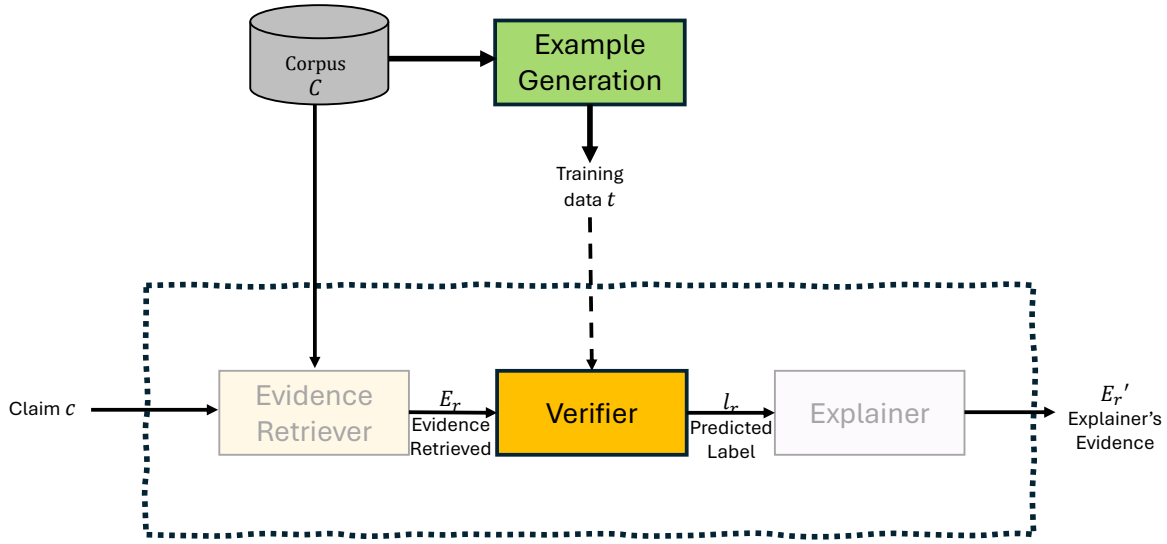

Figure 1.5: Example generation pipeline

generate the claim text in the next step. Error injection in a table is a simple task: shuffling columns and adding or removing tuple permits to obtain a table incoherent with the original one.

In the last step, we produce a textual claim from the evidence. A model fine-tuned on a Table2Text dataset is a natural fit for this task. To be consistent with the challenge of claim variety, we produce different kinds of reasoning claims. For instance, the evidence containing ages of persons is used to produce simple claims 'Mike is 24 and Anne is 54', but also an aggregate one, such as 'The oldest person is 54'. We explore both PLM generation and LLM generation, as the cheapest PLM is suitable for low-resource constraints.

Our results show that automated generation of training data in a tabular setting permits to obtain an effective fact-checking model that performs on par with a model trained with human-written data. This represents a first answer to research question 1.

From multi-modal inputs Switching from uni-modal to multi-modal requires modifications and new techniques to create a dataset. The first step of our pipeline is still to select evidence before generating a claim. To select evidence, we first select cells before adding sentences that are the close to the cells, e.g., in the same page. To tackle the multi-modality challenge, we bring the two modalities into a single one. This is done leveraging table to text models, as in the tabular setting above. Once the tabular content is converted to a textual evidence, we use text summarization techniques to convert multiple sentences into a single claim [36].

Using modality conversion and summarization techniques, we finally build a claim out of the evidence. The results we obtained show that automated generation of training data in a multi-modal setting allows to obtain a fact-checking model as accurate as a model trained with human-written data, answering positively research question 1.

Results overview Results comparing the performance of models trained with generated data instead of human data are shown in Table 1.2. We train fact-checking models with the datasets in the table's first column and report for each dataset the final verification accuracy from both human and generated claims. According to results, the models trained with generated claims are competitive with models trained with human ones, falling only few points below in most case.

Dataset	Type	Work	Accuracy for Human	Accuracy for Generated
TabFact	Table	TENET	68.8	66
Infotabs	Table	TENET	82.5	60
Feverous	Table	TENET	88.6	82.5
	Text	UNOWN	94.5	92.3
	Table+Text	UNOWN	82.1	84.6
MMFC	Table+Text	UNOWN	87.6	76

Table 1.2: Fact-checking accuracy results over four datasets for a verifier trained with human examples (Human) and with examples created by our systems (Generated)

1.4 Explaining a model's decision

Explaining a verifier's decision is an important step in fact-checking. Professional fact-checkers clearly ask for "Models designed in such a way that their outcomes are explainable, unbiased, and more accountable to ethical considerations" [5]. We first describe the problem, then present challenges occurring for explaining systems. Finally, we briefly describe our solutions to tackle the issue.

1.4.1 Problem

A fact-checking verifier takes as input evidence and a claim, and outputs a label of veracity. But what about the reasons behind a label? Models usually work as a black box, predicting an output without any explanation [33]. Therefore, the only information the model returns given the user claim is the label: 'Supports', 'Refutes' or 'Not enough information'. In order to understand the output of the system, users are eager to see the reasons behind the prediction.

Moreover, assessing the reasons behind a decision is an ethical requirement [37]. During training, any model may learn to use information that discriminates groups of people, e.g., on gender [38], when making a decision. Fact-checking is yet another task where such bias may apply. Indeed, a label switch could happen on some claims based on an unrelated facts concerning, for instance, race, in the model inputs. As new datasets appeared to counter-balance minority discrimination in LLMs, explainability remains a key mitigation for this problem [39].

To provide users the evidence behind a model's decision, a simple solution is expose to the users the whole evidence E_r, as given to the predictor by the retriever. However, this is not easy to consume for humans because of the limitations of evidence retrievers. Indeed, when compared to human retrieved evidence, existing techniques provide a low F1 score on retrieved evidence for the same claims, e.g., 0.1 in the most popular corpus [7]. This explains why verifiers are fed with a lot of evidence, up to 40 sentences (or cells) to obtain the best results in claim verification. This implicates that evidence identified by a retriever contain a lot of noise, making it hard for the end user to understand a decision simply by looking at E_r. Moreover, the evidence is ranked on relevance w.r.t. the claim, but such ranking has no information in terms of the role played by the evidence in the model decision. For instance, an evidence piece, such a sentence or a cell value, for a claim can steer the labelling towards 'Refutes', while another piece steers it toward 'Supports' - this information is not available to the retriever module. Finally, providing such set cannot answer the ethical problem mentioned above, as these evidence pieces are not necessarily all used by the predictor model.

We want to be able to explain what are the evidence pieces used to take a decision. Ideally, we aim at providing evidence-based explanations for any black-box fact-checking system. The creation of such method provides a qualitative answer to 2.

1.4.2 Challenges

Explaining fact-checking models is a challenging task. Once we obtain evidence from a retriever, we do not want to provide the user a ranked list of the retrieved evidence, as done by a re-ranker [40]. Such subset is purely based on information-retrieval principles, without any information about the target predictor model. In other words, such ranking cannot capture the behavior of the predictor.

There are few systems that fact-check a claim and simultaneously explain the decision based on the evidence [41, 42, 43]. However, these models fall short when provided with retrieved evidence that is naturally noisy, as opposed to *gold* evidence. Additionally, none of these models can adequately characterize the role played by the evidence. Specifically, they struggle to articulate anything beyond the usefulness of the evidence in the labeling process.

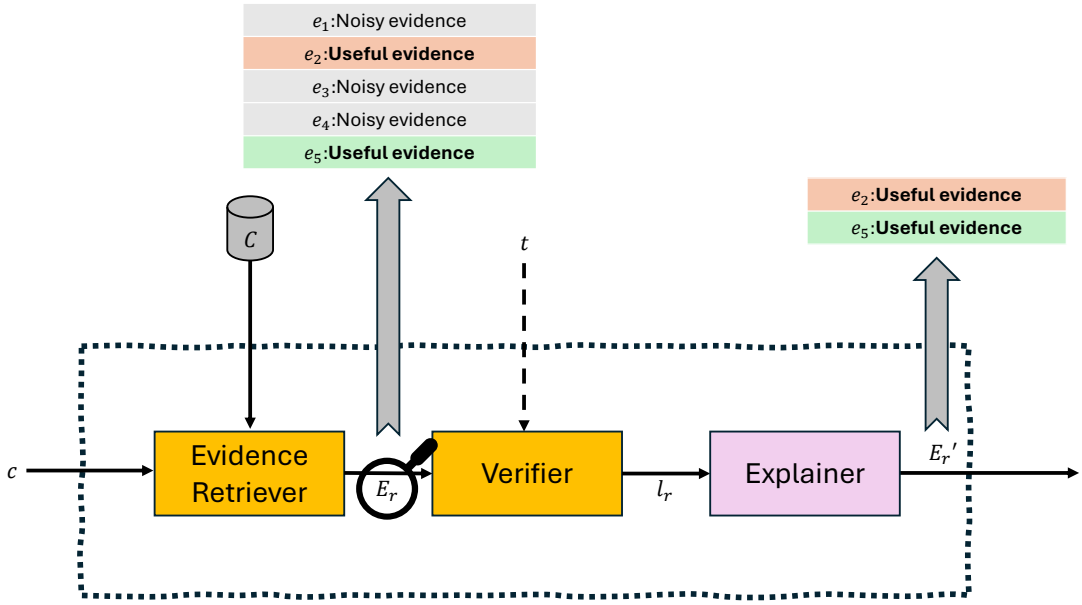

Figure 1.6: The process of filtering retrieved evidence E_r to create a more coherent selection E'_r, reflecting the evidence actually used by the verifier. Each evidence piece push the verifier's classification toward a specific label. Our system detect this behavior, that we highlight in this Figure

Some work explore explanation with an independent module [34, 44]. One of the main challenges in post-hoc explanation systems is dealing with the inherent noise in retrieved evidence. Noise can arise from a variety of sources, including irrelevant or misleading information that complicates the task of discerning relevant evidence from the irrelevant. Consequently, this noise can lead to inaccurate model predictions if not properly filtered.

Additionally, characterizing the role of evidence poses another challenge. Current models typically provide a binary assessment of evidence usefulness without offering insights into how specific pieces of evidence influenced the decision. This lack of granularity limits the explanatory power of the system. A robust post-hoc explanation system should be able to highlight the significance of each piece of evidence, identify which elements supported the final decision, and clarify any counter-evidence considered during the prediction process.

As depicted in Figure 1.6, the integration of the explainer into the fact-checking pipeline must efficiently handle the noisy evidence outputted from the retriever. The verifier processes this noise during its prediction, and then the explainer interacts with the verifier while taking as input both the noisy evidence E_r and the corresponding label l_r. This interaction is crucial to ultimately identify the evidence that is genuinely useful to the verifier, which we designate as the refined set E'_r. Therefore, developing a framework that addresses these challenges is essential for enhancing the interpretability of fact-checking

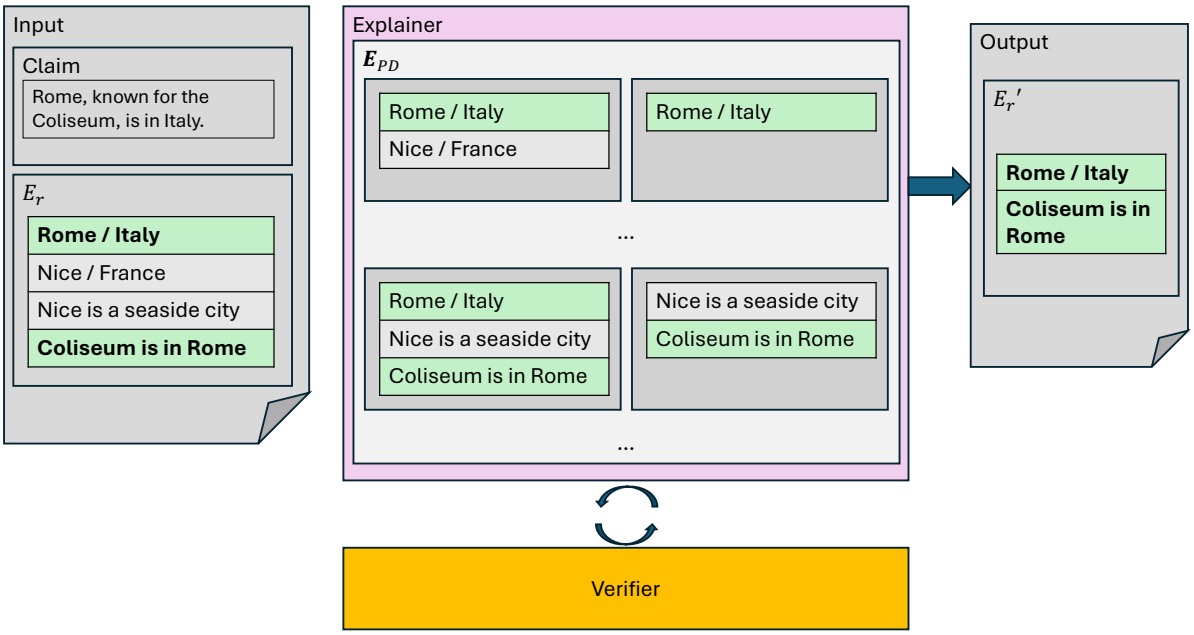

Figure 1.7: Explaining the verifier prediction using xAI techniques. Bold evidence are the useful evidence for taking the decision, and are selected by the explainer as justification.

systems.

1.4.3 Solution

To address research question 2, we adapt and evaluate state-of-the-art post-hoc explainable AI methods [45, 46] to investigate the impact of each piece of evidence on the verdicts produced by existing black-box predictor models. This approach is particularly advantageous as it requires no additional components to be integrated into or interfaced with the model. Instead, the labels for evidence are derived from the model scores associated with each claim label, which allows us to vary the evidence used as input across multiple executions.

We employ two widely used explainable AI (xAI) methods, SHAP [45] and LIME [46], as they are specifically designed to work with any black-box model. By perturbing the inputs of a fact-checking model —specifically, by removing pieces of evidence— we can observe the resulting impact on the predicted label. This output provides insights into the importance of each piece of evidence in contributing to a "Supports" or "Refutes" label. In addition to distinguishing between useful and noisy evidence, these explanatory methods reveal the extent to which a model relies on its underlying knowledge as opposed to the provided evidence when making predictions. By comparing the contributions from evidence and the inherent knowledge of the model, we can assess whether the model predominantly depends on its pre-existing knowledge rather than the evidence supplied. This critical analysis is made possible through our approach.

Figure 1.7 illustrates the architecture employed in this method. Initially, the evidence set E_r is perturbed to generate a collection of modified evidence sets, denoted as $\mathbf{E}_{PD}$. We then run the verifier model on each of these perturbed sets. By analyzing the new scores produced by the model for each perturbation, we estimate the contribution of every individual piece of evidence. Ultimately, this process yields an explanation detailing the impact of each piece of evidence on the final label.

Explanations enable us to address two key questions:

- **Q1**: To what extent does the fact-checking model rely on the evidence and the claim in making its decisions? Does this reliance vary depending on whether a claim is supported, refuted, or

considered to lack sufficient evidence?

- **Q2**: Which pieces of evidence influence the model's predictions? Can the model effectively distinguish between useful evidence and noise, identifying instances of evidence that are unrelated to the claim evaluation?

Our experiments revealed that some fact-checking models heavily depend on the evidence to predict labels, while others assign less significance to it. By employing xAI methods, we can effectively identify which pieces of evidence are useful for the model when evaluating a claim.

1.5 Document Structure

This thesis is organized as follows. First, Chapter 2 provides an overview of existing contributions in Natural Language Processing (NLP) and computational fact-checking. Following this, we present our contributions, beginning with the exploration of synthetic datasets. In Chapter 3, we focus on generating a synthetic dataset based on a unimodal tabular source. Next, in Chapter 4, we expand our approach to include multi-modal sources, incorporating textual evidence alongside the tabular data. In Chapter 5, we shift our focus to explainable AI (xAI) and examine how existing xAI methods can be adapted to enhance fact-checking tasks. Before concluding, we demonstrate a real-world application of our methods in Chapter 6. Finally, Chapter 7 discusses potential future directions for this research.

Chapter 2

Related Work

In the last years, automated fact-checking methods democratized simultaneously to NLP. New approaches in both fields permits to reach unprecedented results, providing solid blocks to construct systems for end users. Before discussing state of the art fact-checking techniques, we first introduce the evolution of Pretrained Language Model (PLMs), as this is the architecture and models used in most of our solution. We then continue our overview by focusing on example generation for AI and specifically on obtaining training data automatically. To conclude, we examine how existing research addresses explainability in the context of fact-checking.

2.1 A Pretrained Language Model Era

2.1.1 Small Language Models

Natural Language Processing (NLP) has evolved significantly over the past few decades, with early methods relying on rule-based systems and statistical approaches using n-gram [47] to model language understanding and generation. Later on, Convolutional Neural Networks have been proposed to improve textual understanding [48]. However, these methods struggled with capturing the complexities of human language. The introduction of neural networks, particularly Recurrent Neural Networks (RNNs) [49], marked a turning point in the field. Among these, Long Short-Term Memory (LSTM) [50] networks introduced significant innovations, addressing the limitations of vanilla RNNs by enabling better handling of long-term dependencies in text sequences. This allowed for preliminary advancements in text understanding and generation. Although there have been improvements with xLSTM models [51], Transformers - with their attention based architecture [52] - revolutionized the field, offering a new level of capabilities and applications. With the profusion of models and frameworks available today, one can easily find a model adapted to a specific task and fine-tune it precisely for such target [53].

Among the vast range of available models, some Small Language Models (SLMs [54]) are particularly noteworthy. Several of them are used across this thesis given their small size that allows fine-tuning and inference even with relatively small GPUs in reasonable time. **T5** [55] (Text-to-Text Transfer Transformer) model is an encoder-decoder model with transformer-based architecture trained on 1T tokens. One of its variant can be fine-tuned to useful to effectively deal with tabular content. More precisely, T5 fine-tuned on the ToTTo corpus [32], which is a dataset containing tables and their textual descriptions. When used to fine-tune T5, it creates a model adept at interpreting table data [56]. Figure 2.1 displays an occurrence of numerical reasoning from ToTTo. Here, the operation is a counting on the years. Thanks to the versality of the dataset in reasoning and desribing, the T5 model fine-tuned on ToTTo is particularly useful to deal with converting structured data to unstructured data, an unavoidable task when building tabular-based claims. **Bart** [57], on the other hand, has been extended into BartNeg, a variant useful for sentence negation with fine-tuning [58]. In our work, we use BartNeg to create example claims with different labels when crafting a training dataset. Finally, **Roberta** [59], a BERT [60] based model, is oftentimes used in a fine-tuned setting for fact-checking applications [7]. This specific fine-tuned model

Table Title: Robert Craig (American football)
Section Title: National Football League statistics
Table Description:None

		RUSHING					RECEIVING				
YEAR	TEAM	ATT	YDS	AVG	LNG	TD	NO.	YDS	AVG	LNG	TD
1983	SF	176	725	4.1	71	8	48	427	8.9	23	4
1984	SF	155	649	4.2	28	4	71	675	9.5	64	3
1985	SF	214	1050	4.9	62	9	92	1016	11	73	6
1986	SF	204	830	4.1	25	7	81	624	7.7	48	0
1987	SF	215	815	3.8	25	3	66	492	7.5	35	1
1988	SF	310	1502	4.8	46	9	76	534	7.0	22	1
1989	SF	271	1054	3.9	27	6	49	473	9.7	44	1
1990	SF	141	439	3.1	26	1	25	201	8.0	31	0
1991	RAI	162	590	3.6	15	1	17	136	8.0	20	0
1992	MIN	105	416	4.0	21	4	22	164	7.5	22	0
1993	MIN	38	119	3.1	11	1	19	169	8.9	31	1
Totals	**-**	**1991**	**8189**	**4.1**	**71**	**56**	**566**	**4911**	**8.7**	**73**	**17**

Target Text: Craig finished his eleven NFL seasons with 8,189 rushing yards and 566 receptions for 4,911 receiving yards.

Figure 2.1: Example from the ToTTo dataset on numerical reasoning

obtains, for any given textual claim, its veracity label with a simple classification task. These SLMs models can work in low-resources setups and are widely adopted in practice.

2.1.2 Large Language Models

Recently, significant strides have been made with more refined architectures, such as Large Language Models (LLMs). Unlike earlier models like T5, which focus on converting text inputs to text outputs specifically, LLMs leverage unprecedented scale and capacity to understand and generate human-like text across a broader spectrum of applications, pushing the boundaries of what these models can achieve. Instruction-based models [61] perform various tasks without additional fine-tuning, simply by providing the right instructions in the prompt. Techniques have appeared to improve the results obtained with this easy-to-apply method. Whether we consider chain-of-thought prompting [62], or expert prompting [63], research has found ways to enhance the capabilities of those models significantly with very low additional user effort. The main drawback of those models is their cost at inference and training. They are very expensive to run, requiring powerful GPUs, with the smallest models operating with parameters going from few billions [64] up to the hundreds of billions [65]. Furthermore, leading LLMs such as **GPT-4** [9] are not open-weighted and require payment through an API. Some open-weights models are worth noting, such as **LLaMa2** [66] and **LLaMa3** from Meta, trained on 2 trillion tokens. A common limitation of those models is that they cannot be easily retrained for specific applications or domain, leading to solutions such as Retrieval-Augmented Generation (RAG), to provides access to information of events subsequent to the model's training [67].

The list of existing LLMs keeps growing daily. Even though we do not exploit them in our work, it is worth mentioning Claude [68], Gemini [69], and Phi-3 [70]. To make the best choice among all those candidates, several studies compare results obtained by these models on different tasks [71, 72].

In a fact-checking context, those versatile models can be used for any step of the pipelines: generation of a synthetic dataset [73], fact-checking, and explainability [44]. However, even if they can be used easily, they are often not competitive with models tailored (with fine tuning) towards a specific task, as in out-of-domain datasets [11].

2.1.3 Fine-tuning

Fine-tuning PLMs is a popular approach to adapt the models to specific tasks or domains, as it allows specializing general-purpose models to effectively handle niche areas. SLMs and LLMs can be fine-tuned, with the latter usually involving quantization techniques to manage their important number of

parameters [74]. Techniques such as Low-Rank Adaptation have also gained traction as they allow more efficient updates to the model without retraining entirely the large network [75]. The fine-tuning process can significantly improve performance especially in domain-specific contexts [76]. This adaptability makes fine-tuning a vital technique for enhancing the utility of PLMs in tasks such as fact-checking.

Specifically for LLMs in their instruct variant, techniques appeared to counter-balance cases where performing an actual fine-tuning is not feasible. The difficulty might arise from the size of the model, or the lack of training examples. Those techniques include few-shot learning [12] and demonstration based prompting [77, 78]. In the former, we provide in the prompt examples of inputs and the desired outputs. For the latter, the examples are more structured with explanations.

2.2 Fact Checking

Fact checking has a long historical precedent, traditionally being the responsibility of editors and human researchers in every newspaper to ensure the accuracy of published content [79]. Here we focus on Computational Fact Checking, a field that has appeared in the last decade to assist human fact-checkers in their tasks. But what is fact-checking?

Fact-checking is the task of verifying the veracity of a given claim. Fact-checking has its own terminology. Misinformation can be defined as false or incomplete information that is generated or spread by a person who believes it to be true [80]. Disinformation, on the other hand, refers to the deliberate dissemination of false information, with the intent to mislead [3, 81].

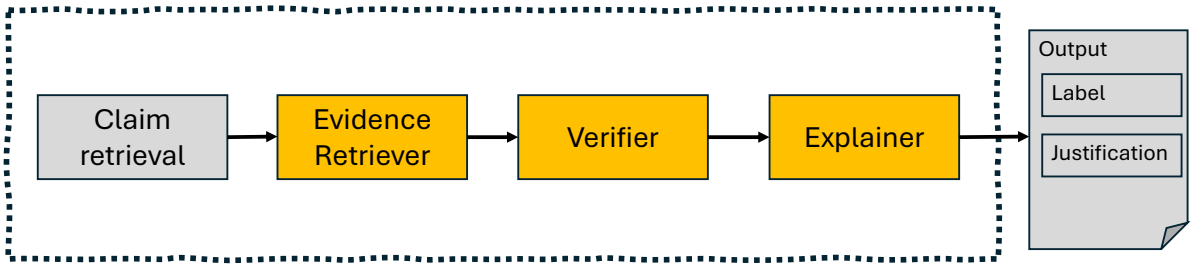

Figure 2.2: Standard fact-checking pipeline

Figure 2.2 presents a traditional fact-checking pipeline, that can apply both for human and computational fact-checking. It starts with retrieving claims that are worth checking, for example from the speech of a politician or from an article online. Finding check worthy claims [82], and finding which claims have already been fact-checked [83], are important tasks as misinformation can spread six times faster than real news [84], requiring quick action. To help deciding which claims are worth checking, one can label the harmfulness level of refuted claims [85]. By interviewing experts, frameworks to categorize misinformation types and rank their threat level have been proposed [81].

Once a claim is noted check-worthy, the next action is to check its veracity. This step requires first to retrieve evidence and then validate the claim against the evidence to assess if the former is accurate [86]. When available, the evidence is complemented with information on the author or the time of publication, as the context brings important information to assess the veracity of a claim and it should not be neglected [81].

Once the checker, human or virtual, is able to give a label to the claim, they produce an explanation. The explanation has the objective of convincing the user to trust the label, basing on the evidence put together [87, 41, 88]. Justification should not be neglected, a number of works [89, 90] emphasize how difficult it is to convince user of a fact-checking results without justification. Failing to provide faithful justification can backfire against the fact-checking output [33]. Close to justification, analysing the motives behind disinformation is an important feature of the pipeline [81]. While it is a common task for fact-checkers, to the best of our knowledge, no fact-checking model is able to help mitigators on it.

Among the main challenges encountered by fact-checkers, the need for more automated procedures is the most important [91, 33]. This is motivated by the amount of content to monitor, which overloads human fact-checkers. We discuss next the main resources that have been proposed to support fact-checkers in their work. While the Claim retrieval step is important, we focus our work in this thesis on three modules in the Figure 2.2 : Evidence retrieval, Verification, and Explanation.

Later in this chapter, we present related methods for synthetic dataset generation, that help building qualitative fact-checking models. We then conclude discussing techniques to explain the output of fact-checking models. All together, these contributions are a step forward helping fact-checkers with the huge amount of claims they have to check.

2.2.1 Automated Fact Checking Overview

Automated Fact Checking (AFC) can be decomposed into several steps, each step having its own particularities [5]. In this part, we report an overview of those steps.

As for manual annotators, an automated fact-checking pipeline starts by finding check worthy claims, before retrieving evidence for those, predicting a label, and finally providing a justification [33]. As justifications are important, ideally, we want AFC systems that are explainable, i.e., they produce a verdict for the given claim and can explain it. However, most existing AFC systems are based on deep learning solutions, such as fine-tuned models obtained with transformers. As those can be considered a "black box" model because of their complex and large architecture, they are not naturally explainable. While some techniques try to explain them by exposing their weights [92], others make the decision-taking understandable by using logical systems [93], as we will see later on in the explainability section. However, when coming to provide the user a convincing justification, for example by exposing the small set of impactful evidence that drove the outcome, those systems fail short.

AFC systems require datasets to be trained or tested on. Some work tried to craft claims from question answering forum [94], as we also do in our creation of an annotated dataset (MMFC). Some datasets are built collecting claims manually from humans [86, 7]. Those datasets serve to train and test fact-checking systems, or even to craft explainability systems. We present other datasets, specific to our tasks, in the next section.

Automated Fact Checking faces many challenges when deployed in practice. Ambiguous claims are an open problem for AFC systems [95, 96]. Claims can be misleading, e.g., "I never lost any chess game" while the speaker never played chess. Disinformation posts often attempt to justify their false claims by selectively quoting or misrepresenting qualitative scientific sources in a dishonest manner [97] – while humans fact-checkers could tackle this issue, automated systems struggle on this task [98]. Automatic models should in general be more accountable to ethical consideration [5] and one way of doing this is to provide clearly the evidence used by a model. Dealing with refuted (likely false) claims is another important challenge. AFC systems particularly struggle on refuted claims compared to human fact-checkers [99]. Another challenge is the ability to go verification of textual claims and multi-modality verification in general [100, 34]. Finally, trustworthiness of evidence sources [101] and bias in the way annotators wrote claim for manually constructed datasets [102] are noteworthy challenges that human fact-checkers need help on.

2.2.2 Datasets

Feverous [7] is a reference dataset in the fact-checking literature and one of the main resources across this thesis. It contains manually curated fact-checking examples, each consisting of a claim, data evidence supporting or refuting the claim, and a label. It consists of 87k claims in total, evidence can be composed of text, tabular data only, or both, making it multi-modal. Figure 2.3 presents an example from this dataset. To the best of our knowledge, it is the only fact-checking dataset relying on both tabular and textual evidence, making it particularly relevant for our work. An important aspect to consider for this type of dataset is the time needed to build it; in that case, a single annotator would need 183 workday to write all claims. Beyond 'supports' and 'refutes', this dataset also contains a third label –'not enough

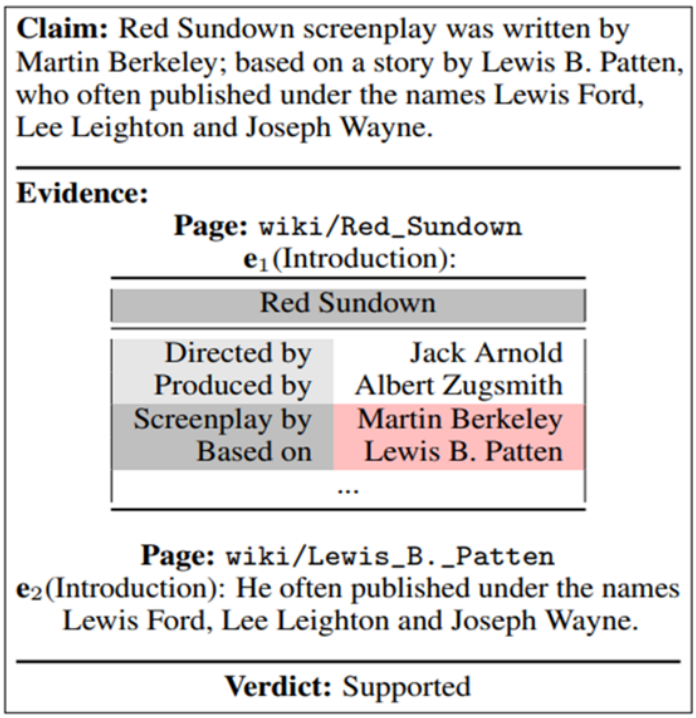

Figure 2.3: Example from a fact-checking dataset

information'– for the cases where the claim cannot be judged given the evidence available in the given corpus. Examples with this labels are a small minority, but we will show that they bring interesting perspectives in an explainability framework.

There are several other datasets worth mentioning for the fact-checking task. First, some datasets that source their evidence on text only. **SciFact** [20] consists of expert-written claims grounded on abstracts of scientific papers as evidence. It contains 1.5k claims with textual evidence and label. **Fever** [86] is the predecessor of Feverous. In this dataset, the 185k human examples are again a collection of claim-evidence pairs grounded on Wikipedia. On the other hand, some datasets contain only tabular evidence. **TabFact** [21] is also based on Wikipedia, it contains 92k claims. Each claim is linked to a whole table that serves as evidence. As the link is at the table level rather than the cell level, as opposed to Feverous, the granularity is coarser. Finally, **Infotabs** [30], also based on Wikipedia, contains 16.5k claims with table evidence, similarly to TabFact.

Claims of those datasets are designed specifically from Wikipedia, therefore, a Fact-checking model trained on them would struggle in out of domain, real conditions. To mitigate this, authors proposed more datasets, such as **FM2** [103] and **AVeriTeC** [104]. The former obtains its claims from an online competition where users build hard claims to deceive other users. The latter involves claims where the evidence can be sourced from the whole internet, and can be retrieved with a web search.

However, existing fact-checking datasets may contain leaked evidence for claims and they may miss sufficient evidence to refute a claim [99]. Therefore, a model trained on such datasets cannot be precisely described as relying on evidence for two reasons. The first is that it might already have seen the evidence in its training, and the second is that it is not feasible to correctly predict a claim as 'refutes' without the corresponding evidence.

2.2.3 Checking Models

In Feverous, along with the dataset, a baseline checking system is provided. The system has the two traditional components, as described earlier for most AFC systems. The first component is a retriever that is fed with a claim and a corpus of documents. To retrieve evidence, the system first selects pages

and sentences using TF-IDF, combined with a PLM to retrieve cell values from tables in the pages. The second component is a predictor, or verifier, that also relies on a PLM oriented toward classification. Given the claim and the retrieved evidence, it provides a verdict. It classifies the claim as either 'supports', 'refutes', or 'not enough information'. This model is built on top of Roberta [59], fine-tuned on different natural language inference datasets [105] before being fine-tuned on the Feverous train dataset.

Several other models have been proposed for fact-checking. Similarly to the baseline system above, more domain-specific model can be built from existing fine-tuned encoders. For example, CT-BERT [22] is a transformer-based model that is pre-trained on a large corpus of COVID-19 related Twitter messages. CT-BERT is specifically designed to be used on COVID-19 content, particularly from social media. In the opposite direction, other models use more complex and expensive agentic approaches on top of LLMs [106]. In OpenTab [107], authors tackle both problems of retrieval and prediction using LLMs combined with SQL. A row selector finds relevant rows for which a LLM generate queries for to answer a question. Once the queries are run, another LLM picks the right answer from the SQL running output. Other models involve both prediction and justification. We will deal of them in the next section.

However, even with better logical and retrieval abilities, fact-checking systems still struggle on complex claims that involve operations such as aggregation for verification. Indeed, existing models cannot perform calculation and computation in a reliable way. In MuMath-Code [108], the authors tackle this issue by using LLMs and training them to generate relevant Python code to solve a target mathematical task. Their work focus on math problem resolution and not on fact-checking. This highlights a promising path on how to tackle this issue.

2.3 Example generation

Synthetic dataset generation has emerged as a pivotal strategy to create training data for a wide array of applications, enhancing model robustness. It is especially effective to counter the expensiveness of human annotation. For example, in the recent NEMOTRON-340B model [65], 98% of the instruct train set has been generated by an 8B model. Synthetic data generation has also proven instrumental in fields such as medical imaging and autonomous driving, where it has significantly reduced the dependency on manually annotated datasets [109, 110].

As the effectiveness of fact-checking models largely hinges on the availability of substantial and diverse training datasets, we discuss synthetic dataset generation for NLP. Since LLMs are getting so powerful, one may think about using them to generate claims directly out-of-the-box. Even though our preliminary experiment has shown their low performance out of the box for fact-checking, we could still use them for text generation. Indeed, 150k training examples for a Chain of Thought task have been created using as a base PDFs of math exercises [108]. Using this synthetic dataset to fine-tune a 7B model, they were able to reach state of the art in AI math Olympiad 2024. Similar exercises have been done for other tasks such as steering LLMs to act as a tutor in assisting students [111]. This show potential for using LLMs to create fact checking examples.

In the context of fact-checking, synthetic data generation has been pivotal in creating large-scale datasets that help improve the performance of fact-checking models, especially in low-resource scenarios [112]. Several works based on SLMs to build training datasets base their generation on existing examples. In **COVID-Fact** [113], the authors create 'refutes' examples from 'supports' human-written claims, releasing a dataset for fine-tuning models. Augmentation is another common method for extending datasets with new examples. Data Augmentation techniques are common data centric approaches, from Computer Vision to NLP [114]. Various techniques exist for NLP augmentation, including adding or removing word and inserting random punctuation [115]. Even simple method deliver interesting results on text classification tasks. There are also approaches that generate synthetic datasets even without existing. Some are based on templates [116, 117]. In CLAIMGEN [118], the authors first generate a question, before generating a claim out of it using a PLM and a Knowledge Graph to build a synthetic fact-checking dataset. Similarly, other works first create questions answers pairs from Wikipedia and then convert them into claims [119].

2.4 Explainability

Most fact-checking models do not provide an explanation along with their decision. However, fact-checkers need to share with readers their interpretation of the evidence [89]. Explainable models are crucial not only for building user trust but also for aiding readers in making informed decisions based on transparent and understandable justifications [120, 121]. There are many possible ways in which a model can be considered explainable, and it is a slippery concept because there is no universally agreed-upon definition. The criteria for explainability often vary based on the context, stakeholders, and application [122]. A model that is simple and easy to understand, such as a decision tree, might still produce results that require domain-specific knowledge to interpret meaningfully. Thus, for different stakeholders, a transparent model may not be equally interpretable across contexts. This variability can lead to conflicting notions of what makes a model interpretable, making it crucial to define those systems with caution.

There are checking systems that provide both a label prediction and a justification, motivated by the need to display the user more than a verdict. In most cases, the explainability module is part of the system and cannot be plugged onto a different, existing system. A first category relies on LLMs, for example to separate a claim in subclaims, predict their labels and justifications [44]. Others use use RAG and in context learning, leveraging LLMs to produce a label prediction for the claim and an evidence selection [67]. However, the explanation provided by the LLM, when asked to justify a veracity label, may not be fully consistent with the actual evidence that led the model to reach the decision. The evidence selected for explanation might differ from the evidence the model truly relied on to make its determination. We must also remember that systems relying on LLMs are currently expensive in terms of cost to execute them.

Other systems rely on SLMs, where, for example, a justification is built at the same time as the fact-checking label is produced [41]. In **CURE** [43], the authors rely on a SLM to infer the usefulness of evidence at the token and sentence level. Therefore, the explanation they provide is a subset of the input evidence. In other systems, attention weights are used to emphasize the most relevant portions of the evidence, where justifications are often represented by scores assigned to each piece of evidence[92, 123, 124]. Another approach that is self explanatory is to use logic-based systems [93, 87], where the explanation can then be directly retrieved from the rules used to obtain the prediction. However, these approaches rely on Knowledge Graph, which limits their applicability in real world setting. Other systems order the relevant subset of evidence in between the retriever and the predictor. For example, re-ranking the evidence of the Feverous retriever model in order to provide the predictor only relevant evidence increase the performance of the pipeline [40]. While this approach is effective to improve the accuracy of prediction, the identified ranking of evidence does not reflect how the predictor uses the evidence, so it cannot be used to explain the output label effectively

Ideally, a justification should explain how the retrieved evidence are used and show the reasoning process taken to reach the verdict [33]. This in order to convey the model's reasoning through a clear and believable explanation. Moreover, transparent and explainable models are essential for identifying and mitigating potential biases that might arise during the fact-checking process, thereby enhancing the fairness of the system [125]. Following this idea, and in contrast with the approaches above, in this thesis we tackle the explanation generation problem using popular post-hoc explanation methods for deep learning systems, such as SHAP [45, 46] and LIME [46], which have the important property of being usable on any system.

In the following chapters, we delve deeper into related works pertinent to the thesis's contributions, providing discussions and contextual analyses that highlight the significance and impact of our research.

Chapter 3

Generation of Training Examples for Tabular Natural Language Inference

Originally published as: Jean-Flavien Bussotti, Enzo Veltri, Donatello Santoro, and Paolo Papotti. *Generation of Training Examples for Tabular Natural Language Inference*. Proc. ACM Manag. Data 1, 4, Article 243 (December 2023), 27 pages. https://doi.org/10.1145/3626730

Diving into the world of synthetic dataset generation, we first start with tabular data as source. In this work, we leverage NLP techniques with database query language, in order to generate high quality claims for fact-checking models.

3.1 Introduction

In Natural Language Processing (NLP), a large class on natural language inference (NLI) problems aim at classifying a given hypothesis, such as a textual statement, as true/false/unknown given some evidence. While this is a well-studied problem for the setting with text as evidence [126], recently it has emerged a new class of applications that focus on inference with structured data as evidence, i.e., *tabular natural language inference* (TNLI). Example applications are table understanding [127, 30] and computational fact checking, where systems label text claims according to input structured data [31, 128, 129][1].

The best solutions for TNLI are supervised. Manually defined datasets for TNLI have been proposed, such as Feverous [7], TabFact [130], and Infotabs [30]. However, these datasets have three main issues. (i) They cover only some generic topics with tables from Wikipedia. For example, if there is a need for fact-checking claims for emerging domains such as Covid-19, a new annotated corpus must be crafted by manually writing examples using the tabular reports published by governments. (ii) They are not comparable in scale and variety to those available for textual NLI [126]. In terms of reasoning requirements, about 80% of the examples in Totto [131] have sentences describing the data with text that does not contain mathematical expressions, such as max, min and count, or comparison across values. (iii) They contain bias and errors that may lead to incorrect learning in the target models [132].

The problem of the lack of labeled examples has been treated in the literature for NLI, but it has not been tackled yet for TNLI. If some examples are given, in a *warm start* setting, existing NLI augmentation methods can be used in the TNLI setting: the text part of the example can be rewritten with augmentation w.r.t. the (fixed) data [133]. While these methods increase the number of examples, they do not generate a new corpus that raises the variety and complexity of the examples w.r.t. the structured data, ultimately with minor impact on the accuracy of the TNLI tasks. Moreover, in a *cold start* setting, where training data is not available, there is no proposal yet on how to create annotated examples for TNLI starting only from the tables.

[1] E.g., https://coronacheck.eurecom.fr

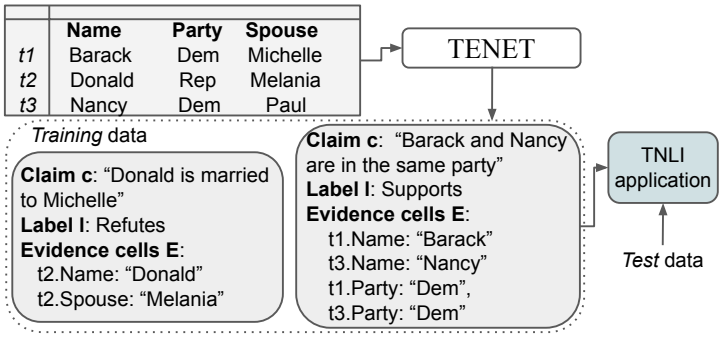

Figure 3.1: Given any table, TENET generates new training examples for a target TNLI application. The first example has a hypothesis that is refuted according to the data evidence.

In this work, we argue that user provided tables can be exploited to generate ad-hoc training data for the application at hand. Our system, TENET[2] (TExtual traiNing Examples from daTa) generates large annotated corpora of training examples that are complex and rich in terms of data patterns, linguistic diversity, and reasoning complexity. Figure 3.1 shows an overview of our proposed method. The system generates training data for the target TNLI application given only a table as input. Once the examples are generated, they are used to train the inference model that is validated on test data.

TENET is built around three modules that cover the three main elements of a complete and annotated TNLI example.

Data Evidence. A key intuition in our approach is that tabular data already contains rich information for new examples. Content changes across datasets, and every relation has its own active domain. Moreover, data relationships across entities and their properties are arranged differently across datasets. To identify *data evidence* to create a variety of examples, we propose alternative approaches to select sets of cells from the given table, including a query generation algorithm for the semi-supervised case. A query returns a set of evidence, such as *Donald* and *Michelle* in the first example in Figure 3.1, each partially describing an example.

Textual Hypothesis. Once the data is identified, we obtain the textual statement (or *hypothesis*) for the annotated example. Given a set of cells, we generate queries that identify such data evidence over the input table. Every query characterizes the data with different conditions (e.g., selections with constants) or constructs (e.g., aggregate). From the query and the evidence, we create a text with a prompting method that exploits the human-like generation abilities of large pre-trained language models (PLMs), such as GPT-3 [134]. Our prompting leads to a variety of hypothesis that are factual, such as *Barack and Nancy are in the same party* in the second example in Figure 3.1, while maximizing the coverage of the provided evidence and minimizing hallucination.

Inference Label. Finally, we need the corresponding *label* for every example. While Supports examples are obtained naturally, as the hypothesis reflects the evidence from the table, for Refutes examples we introduce generic methods built around the idea of injecting errors in the data evidence. Once the data is modified, the process for text generation is applied to the "dirty" data to obtain hypothesis that are refuted w.r.t. the original "clean" data.

Our contributions may be summarized in the following points:

- We introduce an end-to-end system that generates TNLI example from tabular data (Section 3.2). The system is generic, as it does not make assumptions w.r.t. the content of the input tables. The architecture supports both unsupervised generations (cold start) when only tables are provided, and semi-supervised (warm start), where some manually written examples are available. While the former is more general, the latter generates high-quality examples even in settings where the number of tables available for training is limited.

[2]Code and datasets available at https://github.com/dbunibas/tenet

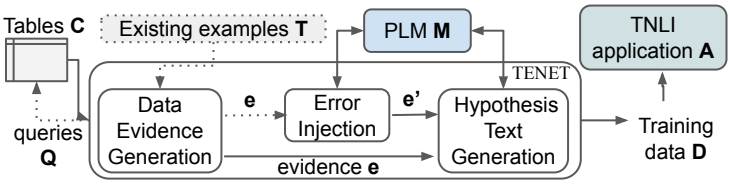

Figure 3.2: TENET overview. Existing examples are optional. Any text-to-text pre-trained language model (PLM) can be used, e.g., ChatGPT. Any target TNLI application can be supported, e.g., tabular fact-checking.

- We introduce algorithms for the generation of the three main components of an annotated example: data evidence (Section 3.3), textual hypothesis (Section 3.4), and Refutes counter-examples (Section 3.5). In every component we enforce variety in terms of data patterns and reasoning challenges.
- We show results for five TNLI test datasets, comparing the results obtained by training with manually written examples vs those obtained with training data generated by TENET (Section 3.6). Training examples generated with TENET lead to reasoning models that outperform the accuracy of the same models trained with data from human annotators in a variety of settings. We also show that TENET's examples can be used in test data for the validation of reasoning models.

We then conclude the paper with a discussion of related work (Section 3.7) and open research directions (Section 3.8).

3.2 Overview of the solution

Problem Formulation. Let r be a tuple in the instance I for a relational schema R and A_i an attribute in R. We refer with *cell value* to the value of tuple r in attribute A_i and with *table* to the instance I for simplicity[3]. A *textual hypothesis* is a sentence in natural language.

A Tabular Natural Language Inference (TNLI) application takes as input a pair (table c; textual hypothesis h) and outputs if h is supported or refuted by c. *Data evidence* is a non empty subset of cell values from c that varies from a small fraction in some settings [7] to the entire relation in others [130][4]. Solutions for the TNLI task rely on supervised models trained with annotated examples - our goal is to reduce the effort in creating such training data.

We consider solving the *example generation problem* for a TNLI application A where we are given the label space L for A, a corpus of tables C, and (optionally) a set of training examples T for A. Every example is composed by a quadruple (h, l, e, c) with textual hypothesis h, label $l \in L$, set of data evidence cells e contained in one relational table c in the corpus C. We assume access to a text-to-text pre-trained language model (PLM) M. We do not assume access to the TNLI application A at hand. In this work we focus on L with *Supports* and *Refutes* labels only, as those are the most popular in TNLI corpora, e.g., 97% of the examples [7].

In the *warm start* version of the problem, training examples for A are available and used by TENET. In the *cold start* version of the problem, we drop the assumption on the availability of the examples T. In this case, we aim at creating new training examples D for A just by using the tables in C.

Process and Challenges. TENET is designed around three main steps, as depicted in Figure 3.2. Given a relation table $c \in C$, it first gathers the evidence (set of cells) e to produce a Supports example. Second, to enable the generation of a Refutes example, it injects errors in table c to create its noisy version and derive data evidence e'. Third, a textual claim (hypothesis) h is generated for every data evidence e.

[3] Some TNLI corpora contain both relational and entity tables, i.e., relational tables transposed with a single row. TENET supports both, but we focus the presentation on relational ones for clarity.

[4] Our proposal is independent from the size of the data evidence and its retrieval.

The quadruple (data evidence e, textual claim h, label Supports/Refutes, table c) is a complete example for training data D for the target TNLI application A. However, the three steps come with their own challenges.

Table 3.1: **People** table. Cells in bold form data evidence e_1.

	Name	Age	City	Team
t_1	**Mike**	**47**	SF	DBMS
t_2	**Anne**	**22**	NY	AI
t_3	John	19	NY	DBMS
t_4	Paul	18	NY	UOL

Data Evidence. Training examples D must capture the variety of relationships in a table, such as those relating cell values in the same tuple or attribute. An hypothesis is defined over a group of cell values, such as the data evidence e_1 highlighted in bold in Table 3.1 for tuples t_1 and t_2, i.e., names of two people with different age values. Hypothesis "Mike is older than Anne" captures the relationship across these four cell values. A data evidence with two cell values, e.g., Name for tuple t_1 and Age from tuple t_2 can lead to an hypothesis, e.g., "There is a person called Mike and a person 22 years old", but such sentence does not capture relationships across tuples nor attributes. In general, for effective training, the data evidence covered by the examples should cover the variety of patterns that can be identified in a relation.

One approach for the data evidence generation is to pick different sets of cell values at random. While this simple approach is effective and enables an unsupervised solution, there are meaningful patterns, such as e_1, that may be covered rarely by accident. One approach to improve this task and obtain meaningful patterns with a smaller number of generated examples is to infer data patterns from human-provided examples T, when those are available. For an example in T, we identify a query q that returns the cell values in its data evidence as one result row. We then execute such query over the relation. The query leads to more sets of cells (one per result row) that enable the generation of examples following the same data pattern, for example involving t_3 and t_4.

Hypothesis. Given a table c and an evidence set $e \in c$, the latter can be described with a textual sentence. However, the *way* a set of cells is converted to a sentence has a huge impact on the variety and the reasoning complexity of the training data. Indeed, given a set of cells from a table, many alternatives exist for describing it in natural language. Consider again data evidence e_1 in the example. The values in bold can be correctly described with "Mike is older than Anne." or "There are two persons with age higher than 19.". The more alternative sentences for a given data evidence are created, the better the training set for the target model. Unfortunately, most efforts for automatic data-to-text are focused on *surface*, or *look-up*, sentences [131], such as "Mike is 47 years old and Anne 22.". While this kind of sentences are fundamental, we aim at maximizing the variety in the training data. For this goal, we generate various queries that return evidence e given c. Such queries represent different ways of semantically describing the data. We then propose prompting methods for PLMs to generate alternative sentences to describe the evidence set according to the semantics of the queries.

Label. By construction, the generated data evidence is coherent with the semantics expressed in the input table. An evidence set leads to an example with a Supports label w.r.t. the data in the table. However, applications also need examples with a Refutes label, i.e., textual claims not supported by the input table. We tackle this problem with an error injection approach, perturbing the input table to break the original relationships across cell values. This new version of the table is then used to identify again an evidence set e', which leads to a textual hypothesis that does not reflect the semantics of the original (clean) table.

3.3 Data evidence generation

We distinguish *cold start*, where training data for the target application are not available, and *warm start*, where examples exist.

Cold start. Given a table c from the corpus C, a method to gather evidence is to *randomly* selects subset of cell values from c. In a simple setting, the number of cells for each data evidence e_i can be picked from a uniform distribution between 1 and m, where m in TNLI datasets is usually 10 or less. This method can be extended by profiling any available training corpus for TNLI and obtain a distribution of the evidence size, or this can be provided by the user. For Table 3.1, a possible data evidence is the set of cells "Mike", "19', "DMBS". Intuitively, with a large number of samples, the random selection eventually models all possible patterns in the tables.

Warm start. If there exist examples T for the application A, we can replace the random selection with a method designed for using T. The intuition is that every example in T has a data evidence e_i that represents a human-defined pattern over the data. Our assumption is that humans express more meaningful patterns that those that we can guess at random. Therefore, being able to capture these patterns enables us to quickly create diverse sets of data evidence.

To identify a pattern, we resort to the task of query synthesis from the cell values in the data evidence. Given an existing example from D, we refer to it as the *seed* s. An example comes with its label l_s, evidence set e_s, textual hypothesis h_s, and the table used to verify it c_s. Given the set of cell values e_s and table c_s as input, we want to identify the query q that outputs such e_s among its results. Executing such query over the original table c_s, we obtain more data evidences $e_1, \ldots, e_n$ that follow the original data pattern in e_s.

Consider again the example in Table 3.1 with cell values in bold in the first two rows (t_1 and t_2) as seed data evidence e_s. Given such input, we want an algorithm producing a query that returns all pairs of distinct names with their different ages, such as

```
q: SELECT c1.Name, c2.Name as Name2, c1.Age, c2.Age as Age2
    FROM people c1, people c2
    WHERE c1.Age > c2.Age AND c1.Name <> c2.Name
```

Table 3.2: Evidence cell values sets identified by querying c_s with q derived from e_s.

tid	Name	Name2	Age	Age2
e_1	Mike	Anne	47	22
e_2	Mike	John	47	19
e_3	Mike	Paul	47	18
e_4	Anne	John	22	19
...	...	...	...	..

Query q executed on the seed table c_s returns the relation in Table 3.2. Each row in the query result is a set of cells for data evidence with the same pattern modeled by the original evidence e_s in the seed. Notice how e_s is also among the results (e_1). Every row in the result of the query can be used to create a new Supports example.

From Examples to Queries. For a given seed example, the textual hypothesis h_s is available. As there are several approaches to infer SQL queries from text (i.e., *text to SQL problem*, or *semantic parsing*), it seems natural to apply one of those to obtain the query above. However, in our approach we derive the query from the data evidence because text to SQL methods are not applicable to our setting. There are three reasons to explain this failure.

First, in a text to SQL task, the input is a NL question and the goal is to obtain the corresponding query. However, our hypothesis are factual expressions. This breaks the assumption in such methods, which are trained on questions such as "What is the region for Kabul?" or "What are the cities in Germany

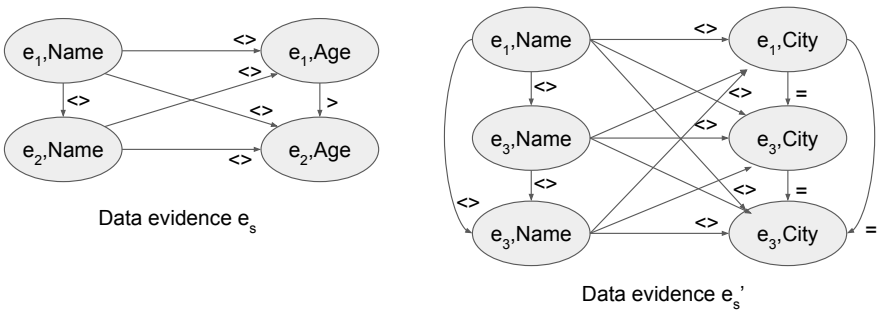

Figure 3.3: Evidence graphs derived from two seed examples.

with more than 10000 residents?". Also, when compared to examples in text to SQL corpora, TNLI examples have longer sentences (average of 25 vs 12 words) and contain a larger number of entities (average of 10.5 vs 4.3 nouns). We tested a system [135] over our hypothesis and, in a manual evaluation of its output for 40 hypothesis, it was able to return partially correct SQL queries only in 20% of the cases.

Second, the table structure in semantic parsing datasets is relational only, while TNLI corpora include entity tables, which have attribute labels in the first attribute and are popular on the Web.

Third, even if existing systems could express a query to identify the data evidence e_s precisely, that would not be useful for our setting. For the goal of generating more examples, we need a query that returns the original data evidence e_s as one of its row results, as in Table 3.2 (tuple e_1), together with more cell sets (tuple $e_2, e_3, \ldots$). The other rows are crucial in our setting as they have the information to produce new examples that follow the same data pattern from the seed. More precisely, given a seed example involving data evidence e_s and table c_s, we are interested in obtaining an *evidence query* (or e-query) q_e that returns the cell values in e_s *in one row of its results* when executed over c_s. This problem is clearly different from general semantic parsing.

While we cannot build on existing solutions, we have the ability to access the data evidence e_s in the seed example, which has precise information about the output of the query. We discuss next how to exploit such seed data evidence to obtain the evidence query.

E-Query Generation. At the core of our solution, we rely on an *evidence graph* to represent relationships among cells in data evidence e_s. Each node corresponds to a cell from e_s and has a label with a pair of values: its attribute label and its row id. The pair acts as identifier for the cell and allows the reconstruction of row and attribute relationships. A (directed) edge across two nodes represent the relationship between their values, e.g., equality, difference, greater than/less than. An example graph derived from data evidence e_s with tuples (Mike, 47), (Anne, 22) is reported in the left hand side of Figure 3.3, while the right hand side reports a graph for data evidence e'_s with tuples (Anne, NY), (John, NY), (Paul, NY).

Once the evidence graph is derived, we construct the query from it by associating every tuple id across the nodes in the graph to a tuple variable in the query, e.g., e_s leads to two variables in the query, while e'_s to three. These variable are used in the FROM clause, e.g., for e_s we get `FROM people c1, people c2`. We then create the SELECT clause going over the union of the nodes and reporting each node with its variable and attribute, e.g., for e_s we get `SELECT c1.Name, c1.Age, ...`. Finally, we add the conditions by navigating the edges according to their direction (equality, greater than, lower than) and corresponding variable and add those to the WHERE clause, e.g., for e_s we get `c1.Age > c2.Age AND c1.Name <> c2.Name AND c1.Name <> c2.Age AND ...`.

The procedure for query generation is detailed in Algorithm 1. Consider the graph derived from data evidence example e_s in Figure 3.3. This graph g, together with the table c_s, is our input for the generation of the e-query q. We initialize the three clauses q_s, q_f, q_w of the query with keywords 'SELECT', 'FROM' and 'WHERE', respectively (lines 1-3 in the algorithm). We start the graph traversal from a node n (line 5), for example node with label e_1.Name. We collect the tuple idx for the node (1 in the example, line

Algorithm 1: Generate Query

Input: table c_s , evidence graph g
Output: query q

```
1  q_s="SELECT"
2  q_f="FROM"
3  q_w="WHERE"
4  visited=[] //Set of visited edges for graph traversal
5  foreach node n ∈ g do
6      tupleIdx = n.getTupleId //Return the tuple idx for the node
7      attName = n.getName //Get the attribute name for the node
8      alias = 'c' + tupleIdx //Get the relation alias for the node
9      if alias ∉ q_f then
10         q_f += c_s + ' AS ' + alias +',' //new relation in From
11     q_s+= alias + '.' + attName + 'AS' + attName + tupleIdx + ',' //new attribute in Select
12     foreach edge e : (n, d) do
13         if e ∉ visited then
14             visited += e //add edge to visited
15             condition = e.getCondition //Get the condition (<, >, =, or <>) for the two nodes
16             q_w += alias + '.' + attName + condition + 'c' + d.getTupleID + '.' + d.getName + ' AND ' //new
                   condition in Where
17 q = q_s + q_f + q_w //union of clauses and removal of pending ',/AND'
18 return q //return query
```

6), the attribute name ('Name', line 7) and the relation alias in the query (c1, line 8). The alias for node e_1.Name is not in q_f, so we add it (lines 9-10). We also add the selection condition 'c1.Name,' - pending commas are removed at the end (line 17). We now process outgoing edges for the node, that become conditions in the Where clause (line 12). If an edge has not been visited, we add the corresponding condition to the Where clause (line 16). For example, for the edge going from Node e_1.Name to Node e_2.Name, we add "c1.Name <> c2.Name AND" - pending AND are removed at the end (line 17). The resulting query is obtained by concatenating the three clauses and returning it (lines 17-18). Once a query is derived, its execution on c gives a result table like the one in Table 3.2.

3.4 Hypothesis generation

One problem in example generation is converting to NL text the data evidence from a table. This generation process is known as the *data-to-text* problem: given the data evidence, i.e. a set of cells, the goal is to create a sentence for the given cell values faithfully. Existing solutions handle this problem with the creation of *surface sentences*, e.g., for evidence (Mike, 47), (Anne, 22) they describe the cells with a sentence like "Mike has 47 years and Anne 22.".

However, TNLI corpora contain sentences that go beyond surface sentences. Our goal is to generate a variety of hypotheses from the data evidence in the way they are described. For the evidence example above, our goal is to generate also sentences such as "Mike is older than Anne." or "Mike is the oldest person, followed by Anne.". These more challenging hypotheses can still be verified with the same evidence, but require more reasoning and are therefore valuable as training examples for the TNLI models.

To tackle this problem, we resort again to the expressive power of SQL. We split the generation process into two steps: i) we compute all the queries over the table such that every query gives as result the data evidence, and ii) for every query, we use a PLM M to generate the desired hypothesis. We discuss these two steps next.

3.4.1 Semantic Queries for Text Variety

Our intuition is that data evidence can be described by the several SQL queries that identify it in the table. These queries are alternative ways to describe the data. By computing the queries, we immediately obtain a *semantic* characterization that can be used to generate hypothesis beyond surface sentences. Given a table c and data evidence e, a *semantic query* (or s-query) over c returns exactly e before the execution of the aggregate functions. Notice that in this case we are not after multiple results, as in the e-query that identifies several examples at the extensional level (over the tuples). The goal is query diversity for the same set of cells; we want variety at the intensional level (over the data description).

Analyzing the examples in popular TNLI corpora [7, 130, 30], we identify two main types of queries.

Table 3.3: Data evidence e_2 (in bold) and e_3 (underlined).

	Name	Age	City	Team	Salary
t_1	Mike	**47**	SF	DBMS	50k
t_2	Anne	**22**	NY	AI	50k
t_3	John	**19**	NY	DBMS	35k
t_4	Paul	**18**	NY	UOL	55k

Local s-query. This type of query leads to hypotheses related only to the values in the evidence. We name them *local*, as they do not involve information outside the cells in the data evidence. Consider evidence e_1 Table 3.1; possible queries for it are:

- Surface (or Lookup) s-query: a query that selects cells only with constant selections; `SELECT c1.Name, c2.Name, c1.Age, c2.Age FROM People c1, People c2 WHERE c1.Name = 'Mike' AND c2.Name = 'Anne' AND c1.Age = 47 AND c2.Age = 22`
- Comparison s-query: a query that compares two or more rows by at least one attribute; `SELECT c1.Name, c2.Name, c1.Age, c2.Age FROM People c1, People c2 WHERE c1.Name = 'Mike' AND c2.Name = 'Anne' AND c1.Age > c2.Age`

Global s-query. This type of query generates hypotheses related to information in the entire table. Here, SQL constructs involve constants and attributes not only in the data evidence. If we also consider the table then more queries can be defined:

- Filter s-query: it selects the cells in the evidence according to conditions. For example, given e_1 and Table 3.1, an s-query that identifies the people with AGE greater than 19; `SELECT c1.Name, c1.Age, FROM People c1 WHERE c1.Age > 19`.
- Aggregate s-query: it selects the cells used in an aggregate operation. If the column is numerical, the aggregate function can be sum, avg, count, max, or min. If the column is categorical, then only count is used. Evidence e_1 cannot be identified with an aggregate s-query. However, if the evidence is the entire Age column, as for e_2 in Table 3.3, an aggregate s-query identifies such values, as we test the exact containment of the cell values involved in the query before the aggregate function. E.g., an aggregate query that returns the highest age in the group is: `SELECT MAX(Age) FROM People`
- FilterAggregate s-query: it selects the result of an aggregate over a group identified by a selection. Evidence e_3 in Table 3.3 contains cells (22, NY, 50k), (19, NY, 35k), (18, NY, 55k) and can be identified by a FilterAggregate s-query stating that there are three people from NY with an average age of 19.7 years and the minimum salary is 35k: `SELECT COUNT(City), AVG(Age), MIN(Salary) FROM People WHERE City='NY'`

Generating s-queries. Given as input the evidence e and the table c, we want to infer every query q_i such that $q_i(c) = e$ before execution of the aggregates.

Unfortunately, the problem of synthesizing even simple queries from a subset of cells has been shown to be not tractable [136, 137]. Our case is even more challenging as we are interested in queries with aggregates and filters. To keep the query generation lightweight, our trade-off is to consider only the subset of 5 possible s-query types presented above, effectively biasing the query generation presented in Algorithm 2.

Algorithm 2: Generate S-Query

Input: set of evidence cells e (i.e., a set of $tid.attr$), table c
Output: sQueries

1 *sQueries* = [] //Output
2 *eAttrs* = { } //Set of attributes used in e
3 *eTids* = { } //Set of tids used in e
4 *eAttrsTids* = { } //Dictionary<attr,tids>: $\forall$ attr. $\rightarrow$ set of tids $\in e$
5 *eAttrsVals* = { } //Dictionary<attr,values>: $\forall$ attr. $\rightarrow$ set of values $\in e$
6 *oAttrsTids* = { } //Dictionary<attr,tids>: $\forall$ attr. $\rightarrow$ set of tids $\notin e$
7 *oAttrsVals* = { } //Dictionary<attr,values>: $\forall$ attr. $\rightarrow$ set of values $\notin e$
8 **foreach** *cell* $v \in d$ **do**
9 *eAttrs* += v.attr; *eTids* += v.tid; *eAttrsTids[v.Attr]* += v.tid
10 *eAttrsVals[v.Attr]* += getCellValue(c, v)
11 **foreach** *cell* $v \in$ *(data* $\setminus e$*)* **do**
12 *oAttrsTids[v.Attr]* += v.tid; *oAttrsVals[v.Attr]* += getCellValue(c, v)
13 n = len(eAttrs) //Number of different attributes in e
14 sQueries += new Surface(project: cells in e)
15 **if** *NOT (sameTIDs(eAttrs, eAttrsTids)* $\wedge$ *len(eAttrsTids[eAttrs[0]]) > 1)* **then**
16 //If for some attribute there are different seleted tuple ids, only a surface query is allowed
17 **return** sQueries
18 **foreach** $attr \in eAttrsVals$ **do**
19 *comps* = findAllowedOperators($attr$, $eAttrsValues$)
20 **foreach** $comp \in comps$ **do**
21 sQueries += new Comparison(project: cells in e, condition: tid in eTids AND generateBooleanComparisons($comp$, $attr$, e)
22 **if** *eAttrsVals[attr]* $\cap$ *oAttrsVals[attr]* $= \emptyset$ **then**
23 sQueries += new Filter(project: eAttrs, condition: attr in eAttrsVals[attr])
24 sQueries += combineAggregateOperators(aggregate: eAttrs, condition: attr $\in$ eAttrsVals[attr])
25 **if** *isNumerical(attr)* $\wedge$ *min(eAttrsVals[attr]) > max(oAttrsVals[attr])* **then**
26 sQueries += new Filter(project: eAttrs, condition: attr > max(oAttrsVals[attr]))
27 sQueries += combineAggregateOperators(aggregate: eAttrs, condition: attr > max(oAttrsVals[attr])
28 **if** *isNumerical(attr)* $\wedge$ *max(eAttrsVals[attr]) < min(oAttrsVals[attr])* **then**
29 sQueries += new Filter(project: eAttrs, condition: attr < min(oAttrsVals[attr]))
30 sQueries += combineAggregateOperators(aggregate: eAttrs, condition: attr < min(oAttrsVals[attr])
31 **if** *len(oAttrsTids[0]) == 0* $\wedge$ *len(oAttrsTids[i]) == 0* $\wedge$ *len(oAttrsTids[n]) == 0* **then**
32 sQueries += combineAggregateOperators(aggregate: eAttrs)
33 **return** sQueries

We first initialize data structures (lines 1-12). *sQueries* contains the s-queries for the given evidence e and table c. For each attribute in e, we keep track of the rows in the evidence (*eAttrsTids*), the cell values of the evidence (*eAttrsVals*), the rows of the data not in the evidence (*oAttrsTids*) and, the cell values in the data not in the evidence (*oAttrsVals*). Considering evidence e_3 and table c in Figure 3.3, *eAttrsTids[Age]* contains t_2, t_3 and t_4, while *oAttrsTids[Age]* has the only row not included in e_3 for attribute Age, namely t_1. Similarly, *eAttrsValues[Age]* contains the three selected values for age, 22, 19 and 19, while *oAttrsValues[Age]* has 47.

The data structures are used to check what s-queries can be generated for the input at hand. Surface

Algorithm 3: combineAggregateOperators

Input: attributes $attrs$, possible empty $condition$, evidence e
Output: sQueries $sQueries$

1 *sQueries* = []; *aggrAttrs* = {}
2 **foreach** $attr \in attrs$ **do**
3 *aggrAttrs*[attr] = *findAllowedAggr*($attr$, e)
4 **foreach** *permutation* p *in permutations(aggrAttrs)* **do**
5 // p contains a list of aggr functions on different attributes *attrs*
6 *sQueries* += new Aggregate(aggregate:p, condition: $condition$)
7 **return** *sQueries*

s-queries can always be generated (line 14), corresponding to returns exactly the cells in the evidence. These queries are flexible and allow us to handle any kind of evidence, while other s-queries require more structured evidence. In particular, to generate comparison, filter, and aggregate queries, all the rows in the evidence should have the exact same attributes selected. In the case when only one row is selected, or some rows have different attributes, the algorithm will stop and only the surface s-query is returned (lines 15-17). To give an example, using evidence with (Mike, 47), (Paul, NY), (DBMS, 35k), the algorithm will generate only the surface s-query.

If the check at line 15 is passed (i.e. we did not interrupt the procedure), we can produce different s-queries. For each attribute in the evidence e, we first check if the attribute enables a comparison among its values in e with the auxiliary function findAllowedOperators (line 19). If some comparison operators are discovered (like $<$, $>$, $=$) then for each comparison (*comp*), we add a new comparison s-query to *sQueries* (line 21). Since in a SQL query in the WHERE clause we can only define pairwise comparisons, we use the utility function *generateBooleanComparisons* to generate all such pairwise comparisons depending on the number of rows in e for the given attribute $attr$. This allows us to generate a WHERE clause that involves t_1.Attr *comp* t_2.Attr ... t_{n-1}.Attr *comp* t_n.Attr.

The next s-queries require a filter over one attribute. Such a query can be generated whether a group of values is selected together for an attribute. More formally, we check if, for an attribute $attr$, none of the values selected in e are present in $d \setminus e$ (lines 22-23). In our example e_3, for attribute $City$, the evidence contains the value NY, and it can be considered as a filter since all rows with the NY value for $City$ are selected. The corresponding WHERE clause is $City$ = "*NY*". Another filter is for a numerical attribute can be triggered when values in the evidence are below/above a constant, e.g., attribute Age in e_3, containing all people younger than 47. This check (lines 25 and 28) verifies that all the values in the evidence are greater (lower) than the values for the attribute outside the evidence ($d \setminus e$).

In addition, once a Filter s-query is generated, we check if an aggregate operation can be used to combine them in a FilterAggregate query (lines 24, 27, 30). An additional function, *combineAggregate-Operators* performs this check.

Notice that *combineAggregateOperators* generates all the permutations for the allowed aggregate operations on given attributes and generates a set of s-queries (Algorithm 3). In particular, for each attribute $attr$, we first compute the allowed aggregate operations (line 3). Given an attribute, *findAllowedAggr* returns count() for categorical attributes and count(), avg() for numerical attributes; if e contains also the min/max value in d then it also returns the min()/max() function. We then generate all the permutations of the attributes and the aggregate operation for each attribute. For example, evidence e_3 in Table 3.3 admits a count(), avg(), and min() for the Age and Salary attributes, while allowing only count() for City. Thus possible permutations generated include [count(Age), count(City), count(Salary)], [avg(Age), count(City), count(Salary)], [min(Age), count(City), avg(Salary)].

Finally, in lines 31-32 of Algorithm 2, we check if an entire column is selected, and for them, we generate multiple Aggregate s-queries with the same approach described above.

We are not claiming that the list of s-query types above is exhaustive and covers all the possible queries that identify the evidence. For example, we are not covering the "Order by" s-query, e.g., the one

for e_1 that identifies the top 2 people with the highest AGE: `SELECT Name, Age FROM People c1 ORDER BY Age DESC LIMIT 2`, leading to the sentence "Mike and Anne are the two oldest persons.", or "Group by" s-query, where one might want to compare aggregate values between two groups to generate a sentence like "People in DBMS team have an average salary higher than people from AI team.". However, our study of the TNLI corpora shows that the five types above cover most of the hypotheses used in practice. In Section 3.6.3, we show how additional s-queries have a small positive impact in the accuracy on the target TNLI application.

Table 3.4: S-Queries generated by TENET.

Name	Example
Surface	**SELECT** a1.Name, a2.Name, a1.Age, a2.Age **FROM** People a1, People a2 **WHERE** a1.tid = t_1 *AND* a2.tid = t_2
Comparison ($<$, $>$, $=$)	**SELECT** a1.Name, a2.Name, a1.Age, a2.Age **FROM** People a1, People a2 **WHERE** a1.tid = t_1 *AND* a2.tid = t_2 *AND* a1.Age $>$ a2.Age
Filter	**SELECT** Name, City **FROM** People **WHERE** City in "NY"
FilterAggregate	**SELECT** max(Age) **FROM** People **WHERE** City in "NY"
Aggregate	**SELECT** count(Name), avg(Age) **FROM** People

Table 3.4 reports the different type of s-queries that might be discovered from Algorithm 2, together with some examples.

3.4.2 Text Generation

Once we know for each evidence the possible s-queries, we generate textual sentences that form the hypothesis for the TNLI example.

We exploit the text generation capabilities of pre-trained large language models (PLMs), such as those in the GPT family [138]. A PLM is trained over huge amounts of textual data, which gives it proficiency in writing, and on source code, which gives it the ability to be instructed with functions.

Table 3.5: Functions used by TENET in ChatGPT prompts.

S-Query	Function	Example
Surface	read(attrList)[*]	Anne is 22 years old and Paul is 18.
Comparison	compare(op, attr)	Anne is older than Paul.
Filter	filter(cond, attr)	Anne, John and Paul are from NY.
FilterAggregate	filter(cond, attr); compute(func, attr)=val	The oldest person from NY is 22 years old.
Aggregate	compute(func, attr)=val	Mike is the oldest person.

For each s-query, we define a task that describes the text generation function that we want to use. Such generation functions are defined by us with the prompts for the PLM. The text generation functions mapped to the relative s-queries are reported in Table 3.5 with examples of the text they generate. The

op parameter is related to operators $=$, $<$ and $>$ for numerical attributes, while $=$ only for categorical attributes. The *cond* parameter is the condition used in the WHERE clause. The *func* parameter refers to an aggregation function among count, avg, sum, min, or max. For example, given the Filter s-query `SELECT Name, City FROM People WHERE City in ("NY")`, we derive the operation `filter(in ("NY"), City)`.

To avoid hallucination of the model in calculating aggregate functions, for `compute` functions we calculate the value val from the evidence and feed it to the PLM; this enables us to avoid such computation with the PLM, as it brings little to this task. For example, given the evidence e_3 in Table 3.3, we explicitly calculate the avg for the attribute Age, and use the calculated value in the operation *compute(avg,Age)=19.66*. This helps the PLM to express a sentence like "The average age is 19.66". In general, our approach based on evidence and s-queries returns factual hypothesis, while a baseline solution based only on prompts for the PLM create examples with hallucination that ultimately lead to worse performance in the target TNLI task. We report on this comparison in Section 3.6.3.

Table: Cars
Model | Year
---
500 v3 | 2012
Clio v6 | 2018

Function:
`compare(>,Year)`
Example:
"The 500 v3 is an older model than the Clio v6"

Table: Table Name
Attr_1 | Attr_2 | ... | Attr_n
---
v_1^1 | v_2^1 | ... | v_n^1
...
v_1^m | v_2^m | ... | v_n^m

Function: *(Desired verbalization expressed by function f)*
Example:
(it returns a textual sentence using v_1^1 ... v_n^m *according to f)*

Figure 3.4: One of the 16 examples for in-context learning (left) and generic serialization of the evidence in the prompt at test time (right).

Since in most cases PLM cannot be fine-tuned, as they are offered with APIs, we opt for using them with a prompt with some examples (few shots, or in-context learning in NLP terminology) [139]. The prompt consists of two parts.

The first part is fixed and contains 16 examples of how to verbalize the data evidence with an operation. For every example, it reports the table name and the text linearization of the schema [29]. Then each row of the evidence is linearized in the same way. If some attribute lack a value in the evidence, we add "null" as the cell value. Finally, we define the expected textual hypothesis. For example, a comparison operation is reported in the left hand side of Figure 3.4 for a data evidence with two tuples and two attributes.

The second part reports unseen data evidence and an operation to steer the model to generate the desired text. The right hand side of Figure 3.4 shows the input before instantiating it with the table and operation at hand.

3.5 Refutes examples generation

The methods above produce Supports examples, i.e., the label states that the data evidence entails the textual hypothesis. This is true by construction, as the hypothesis are derived directly from the evidence. However, TNLI applications have also Refutes examples, where the evidence contradicts the hypothesis. Our approach to the generation of Refutes examples relies on our method for generating the Supports ones. We generate a Refutes example for every Supports one. Given some evidence e from the original input table c, we inject noise in a copy c', so that we derive a new evidence e'. An hypothesis h' is then derived from e'. Hypothesis h' is a Supports sentence for c', with evidence e', but it is also a Refutes

sentence w.r.t. the original (clean) table c and evidence e. The new example is the tuple with label Refutes, c, h' and evidence e.

Table 3.6: A modified version of the "People" table with shuffling of the original "Age" values and one injected tuple.

	Name	Age	City	Team
t'_1	Mike	18	NY	DBMS
t'_2	Anne	19	NY	AI
t'_3	John	22	SF	DBMS
t'_4	Paul	47	NY	UOL
t'_5	Mary	17	NY	SYS

Consider again Table 3.1, denoted as c and the evidence e=(Mike, 47), (Anne, 22). First, we create a copy c' of the table and manipulate it to inject noise. We shuffle in c' the values for 50% of the attributes (we discuss in Section 3.6.3 how we set this threshold) involved in e. The resulting table is reported in tuples t'_1 - t'_4 in Table 3.6, only Age has been shuffled. This step breaks the original relationships across cell values at the tuple level. We then either introduce a new tuple in c', such as t'_5, or remove from c' one tuple at random. This step changes the cardinality of the tuples, which is key for s-queries involving aggregates, and introduces out-of-domain values. The generation of the new values depends on the type. For categorical attributes, we use a PLM, which generates "Mary", "NY" and "SYS" for tuple t'_5. For numerical attributes, we generate lower/higher values than the min/max value for every active domain - these new values break the original min/max/avg property for the updated attribute, e.g., the new min value "17" in t'_5. Finally, we remove from c' any row that appears in c.

Given the new "noisy" table c', we directly apply the generation of Supports claims from Section 3.4. We use evidence e to generate an e-query q over c, then we use q to obtain the new evidence e'. Finally, we generate h' from e' and c'. Hypothesis h' is supported by evidence e' (table c'), but it is refuted by original evidence e (table c). For example, a claim may state "Mike is younger than Anne", which is refuted as hypothesis w.r.t. the data in Table 3.1. As another example, consider the evidence e_2 for the FilterAggregate s-query (Table 3.3), which takes all Age values. In this case, there is no shuffling, but the new evidence e'_2 includes 17. Therefore a Refutes claims for operation Max cannot be generated, as (47) is a valid evidence in e_2, but an hypothesis involving Min can be generated, as (17) is not in e_2.

3.6 Experiments

We organize our evaluation around four main questions. First, does TENET automatically generate training data of quality comparable to those manually created by human annotators? Second, what is the impact of the information stored in the PLMs at the core of most inference models for TNLI? Third, what is the impact of the models and parameters used in TENET? Fourth, what are the costs of TENET, in terms of execution time and budget for external APIs?

Before getting into the discussion of the results, we present datasets, models, and metrics used in the evaluation.

Train Datasets. We use three datasets from the TNLI literature: FEVEROUS [7], TABFACT [130], and INFOTABS [30]. Each dataset comes with one subset (split) of examples for training and one for test. Every annotated example consists of a *table*, a *textual hypothesis*, *data evidence* (subset of the table), and a Supports/Refutes *label*. All examples are manually written by humans. We report dataset statistics in Table 3.7; "Avg # of row/attributes" is per table.

As a baseline, we extend the original training datasets with an augmentation for text [140]. Given an example, we produce seven new versions of it by changing the textual hypothesis using back translation, wordnet, word2vec, synonyms, random word swap, random word deletion, random word insertion (*Aug*).

We also produce training datasets for our techniques. Given a corpus of tables, we always generate

Table 3.7: Statistics for the datasets. All datasets except OUTOFDOMAIN and SWAPPED have train and test splits.

		Source	# of examples	Avg hyp. length	Avg # of row/atts
Train	FEVEROUS	Wiki	10k	122.0	10/3
	TABFACT	Wiki	92k	73.0	14.5/6
	INFOTABS	Wiki	16.5k	55.5	14.5/2
Test	FEVEROUS	Wiki	1k	123.2	10/3
	TABFACT	Wiki	25k	70.8	14.5/6
	INFOTABS	Wiki	7k	57.1	15.5/2
	OUTOFDOMAIN	UCI	0.15k	45.0	16/8
	SWAPPED	¬(Wiki)	1k	105.0	10/4

the *Tenet Cold* (*TenetC*) dataset (Section 3.3). If examples have annotation for data evidence, we can also generate the dataset for *Tenet Warm* (*TenetW*). Hypothesis are created with s-queries (Section 3.4) and negative examples are generated according to Section 3.5. For each given table, we produce three Supports and three Refutes hypothesis, therefore all TENET datasets are balanced in terms of labels.

For every table, TENET creates one example with a surface query (cause those are the most popular kind in the corpora and can always be generated) and two for the two rarest s-queries among the other four types (Comparison, Filter, Aggregate, FilterAggregate). Table 3.4 reports s-queries from the more commonly observed in the corpora to the rarer. If the complex s-queries cannot be generated, the remaining examples are obtained with surface queries.

Test Datasets. The datasets from previous papers (FEVEROUS, TABFACT, and INFOTABS) have their own testing datasets with annotated examples manually written by humans (statistics in Table 3.7). However, as all these models use tables from Wikipedia, we create also a test dataset with eight out-of-Wikipedia (OUTOFDOMAIN) tables selected from different sources. Finally, TENET can go beyond its role in the training step and be used also to generate test datasets, which is useful for evaluation of existing methods. In this spirit, we also generate a test dataset, SWAPPED, as described in Section 3.6.2.

Inference Models for TNLI. In this work, our goal is to show the quality of automatically generated training data. We therefore do not propose new TNLI models and adopt the ones in the original papers. In FEVEROUS, the inference predictor is a RoBERTa (large) encoder fine-tuned for classification on multiple NLI datasets [141]. In TABFACT, the inference predictor is built as a program synthesis problem, modeled as a latent program search followed by a discriminator ranking [142]. In INFOTABS, the inference predictor is also a RoBERTa (large) encoder fine-tuned for classification.

Pre-trained Language Models. For the hypothesis generation (Section 3.4) and the error injection (Section 3.5), we assume that a pre-trained language model (PLM) is available. We tested several PLMs and use ChatGPT as default. We report a comparison of T5, fine-tuned on ToTTo, and ChatGPT in Section 3.6.3.

Metrics. We report accuracy for the TNLI task: how many Supports/Refutes classification decisions are correct over the total number of tests. We also report execution times and cost (for external APIs) in running the models (Section 3.6.4).

3.6.1 Quality of Training Examples

We start by comparing results with training data with examples generated from the same sets of tables. The tables are taken from FEVEROUS, TABFACT, and INFOTABS datasets. As state of the art solutions, we directly use the manually written examples (*Human*), eventually augmenting them (*Human+Aug*). For TENET methods, we take the corresponding tables of the original training data and generate examples with *TenetC* and *TenetW*. For every experiment, we increase the number of input tables, collect or generate

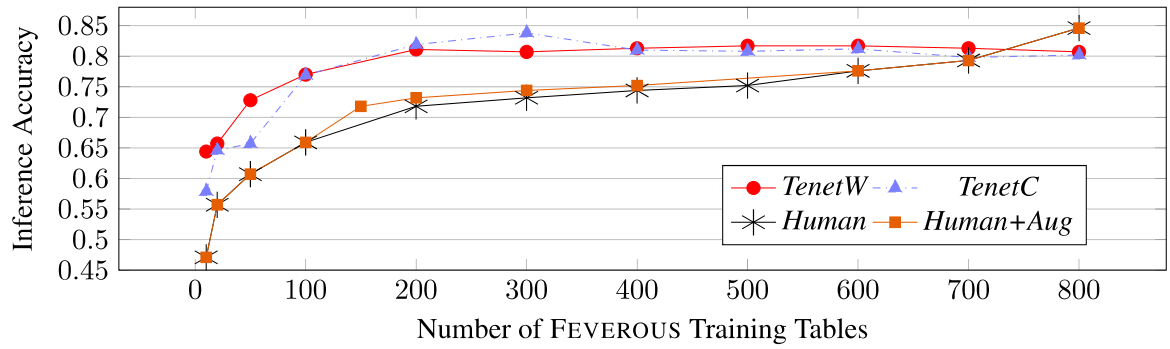

Figure 3.5: Inference accuracy for different training datasets over the FEVEROUS test data. The *x* axis is the number of tables in training set. *Human* is FEVEROUS original training data.

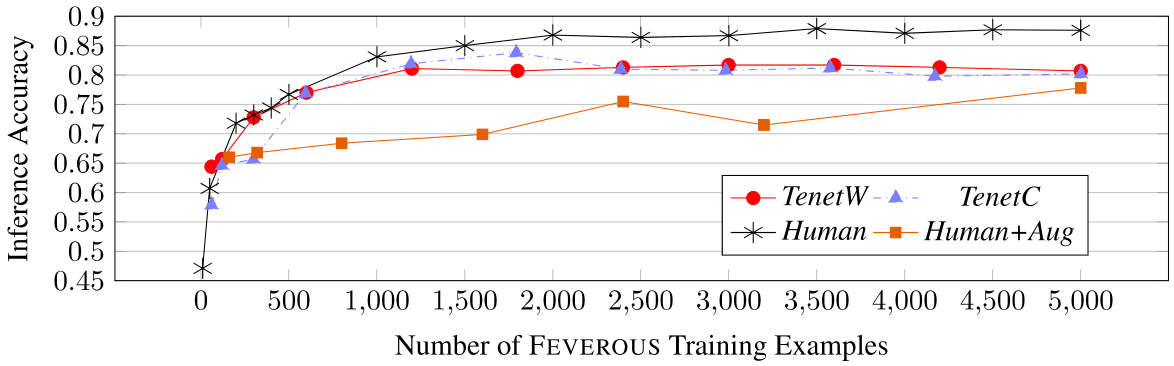

Figure 3.6: Inference accuracy for different training datasets over the FEVEROUS test data. The *x* axis is the number of examples in training set. *Human* is the FEVEROUS training data.

the examples, and run the inference model to compute the accuracy on the test data. For example, given a subset of the original examples in FEVEROUS training corpus, *TenetC* generates evidence and hypothesis using only the table in every example, while we use the original example for *Human*. We finally assess the quality of the examples, both original and generated, on the same test splits.

The TNLI accuracy results in Figure 3.5 for the FEVEROUS test data show the impact of examples, which is a proxy for their quality. Up to 700 input tables, both TENET-generated datasets outperform the examples written by humans, with more than 20 absolute points in cases with less than 150 tables. Even with only 200 tables available for the training step, both TENET example generation methods achieve an accuracy over 0.8 on the (manually crafted) original test data. If we augment the Human examples with those generated by *TenetW*, we observe accuracy at 0.8 even with only 150 tables in the training corpus.

TENET benefits by the fact that for every input table, it extracts one data evidence and generates three Supports and three Refutes examples, while the humans wrote one example per table. To make a comparison over the same number of examples, we report the same experiment, but with results plotted according to the total number of examples, regardless of the number of tables.

Figure 3.6 compares the results obtained with sets of examples of the same size, but from different methods, on the FEVEROUS test data. TENET's examples (both *TenetW* and *TenetC*) lead always to higher accuracy than the original examples with traditional augmentation (*Human+Aug*). Moreover, TENET's examples lead to comparable accuracy w.r.t. the human-written corpus up to around 1.5k examples. After this value, the results are quite stable for our generated datasets, while they slowly increase for those written by humans. This is consistent with Figure 3.5, as *Human* outperforms TENET when using at least 800 tables. Our explanation is that there is a long tail of reasoning cases that are not covered by the five s-queries that we have designed, e.g., FEVEROUS test data has a small fraction of hypothesis involving arithmetic operations. While new s-queries can be added, the plot shows that with only five types we can

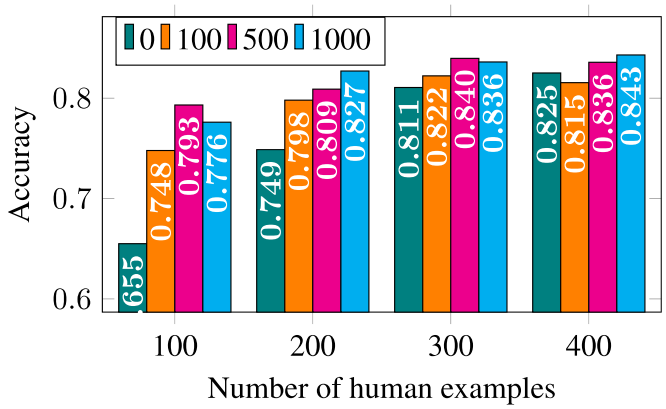

Figure 3.7: Inference accuracy on FEVEROUS when training with the union of human examples (100 to 400) and TENET generated examples (0 to 1000). The first bar is for *Human* examples only, other bars are for *Human+Tenet* examples.

already obtain automatically very good training datasets.

Figure 3.7 reports the results for the training done with a combination of *Human* and *Tenet* examples for FEVEROUS. We report the impact of different numbers of generated examples. Increasing the size of the generated training data increases the accuracy on the test set. The benefit of TENET examples is higher with smaller numbers of human training examples.

Table 3.8: Accuracy on FEVEROUS test set augmenting the original train set with TENET and text augmentation examples. TENET-X stands for examples generated from X tables.

Train set	Augmented	Augmented Size	Accuracy
FEVEROUS	-	-	0.909
FEVEROUS	TENET-50	153	0.910
FEVEROUS	TENET-100	321	0.916
FEVEROUS	TENET-200	683	0.911
FEVEROUS	TENET-300	**1018**	**0.917**
FEVEROUS	TENET-400	1357	0.910

Table 3.8 reports results for a combination of *Human* and *Tenet* train examples on FEVEROUS. We augment the entire original training set with TENET's examples using an increasing number of seed tables. The best accuracy is obtained with 300 tables (1018 training examples) and an accuracy of 0.917. A larger number of generated examples has a smaller impact. We observe a similar pattern with the baseline text augmentation [140]: adding all augmented examples to the original human examples leads to a lower accuracy (0.908).

Figure 3.8 reports the results for the accuracy of the inference model for INFOTABS and TABFACT test datasets. For both datasets, the examples generated by human annotators do better than TENET examples. One difference from FEVEROUS is that these datasets have up to eight examples per table, therefore the accuracy grows faster with more tables compared to Figure 3.5. For TABFACT, the difference is a few points, while for INFOTABS is more significant. This is because the latter contains only entity tables derived from Wikipedia info-boxes. Those are equivalent to tables with a single row and many attributes, thus not suitable for s-query generation and our algorithm defaults to surface hypothesis for these cases. Finally, INFOTABS examples use the whole table as data evidence, which explains why we cannot derive e-queries for *TenetW* for it. However, we remark that our examples are generated without involving humans, therefore with a cost that it is a fraction of the one to obtain the original training datasets.

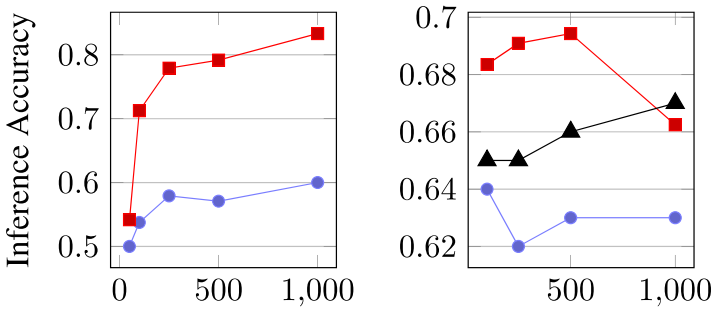

Figure 3.8: Inference accuracy for different training datasets over INFOTABS (left) and TABFACT (right) test data. The *x* axis is the number of tables in training datasets. The red curve corresponds to Humans, the blue curve to Tenet Cold, and the black curve to Tenet Warm.

3.6.2 Impact of Information from Pre-training

In this experiment we measure the impact of the knowledge stored in PLMs. Are the inference models really using the input evidence and tables? Or they rely on the information in the PLMs?

Indeed, PLMs have been trained with large amount of information, including dump of the Web and Wikipedia. With existing datasets derived from Wikipedia, it is not obvious how much of the inference decision comes from the information gathered in the weights of the large language model and how much is coming from the evidence and table passed as input for the TNLI task.

To enable such analysis, we use two test datasets: OUTOFDOMAIN and SWAPPEDFEVER. We design OUTOFDOMAIN with five tables from the UCI repository [143] (Abalone, Adults, Iris and Mushroom) and three sports tables used in NLP text generation challenges [144]. Hypothesis and labels are manually crafted by the authors with the generation process outlined in the Feverous paper [7].

For SWAPPED, the goal is to create hypothesis that contradict the information in Wikipedia. For this task, we create hypothesis that are supported by the tables given as input, but are in contradiction with the original Wikipedia tables, which are likely present as learnt information in the PLMs. To create this dataset, we take tables O from the Feverous corpus and create Supports hypothesis A with our methods. We then inject errors in the tables, obtain tables O', and create Refutes hypothesis B. We then swap the labels in the examples. We change the labels of the original Supports hypothesis A, as they are now Refutes for tables O', and do the same for B. The (now) Supports hypothesis in examples B are supported by the provided tables, but are in contradiction with the original Wikipedia tables that have been used in the pre-training of the PLMs.

For this experiment, we train the FEVEROUS inference predictor on TENET training data and on the original FEVEROUS datasets as in the previous section.

Table 3.9: Accuracy results for test datasets OUTOFDOMAIN, derived from non-Wikipedia tables, and SWAPPED, with examples contradicting information in Wikipedia tables. Training examples (5k) derived from FEVEROUS tables.

	Generated Train Set		FEVEROUS Train Set	
Test set	***TenetW***	***TenetC***	***Human***	***Human+Aug***
OUTOFDOMAIN	**0.84**	0.80	0.77	0.76
SWAPPED	**0.65**	0.65	0.64	0.61

The results in Table 3.9 show two important insights. First, accuracy results are lower compared to the original datasets from the literature. This is because those inference tasks are defined over concepts and entities that are already "known" to the PLMs used in the inference. This is evident with the SWAPPED dataset that contradicts the original knowledge in the Wikipedia tables used in the pre-training of the

PLM. Models that rely on the provided data evidence, rather than PLMs' knowledge, are more robust when executed on new domains.

Second, the model trained on TENET data outperforms the models trained with humans' examples. Our examples better steer the inference model into learning to use of the data evidence, rather than the internal information in the PLM. This is especially important for domain-specific tables that are covering entities not on the Web, with an improvement of 7 absolute points with *TenetW*'s model over the humans' model.

3.6.3 Ablation Study

In this section, we first measure the impact of the PLM on the quality of the generated examples. We then study the impact of parameters used across the data evidence and hypothesis generation.

Role of PLM. As a baseline for the first experiment, we report for the training data produced directly by a pretrained language model for this task (*PLM*). We use ChatGPT to automatically generate hypothesis from tables given only a prompt with the instructions and examples. For each table of a given dataset, ChatGPT generates (i) three Support and three Refuse hypothesis using data in the table, and (ii) the set of cells used to produce each sentence (evidence). Tables are presented using the same linearization of Figure 3.4.

Table 3.10: Accuracy results with different PLMs for example generation. Same inference model trained on examples from 300 tables from FEVEROUS train corpus.

	TenetW Train		*TenetC* Train		*PLM* Train
Test set	**T5**	**ChatGPT**	**T5**	**ChatGPT**	**ChatGPT**
FEVEROUS	0.79	0.80	0.81	**0.82**	0.70
TABFACT	0.60	**0.65**	0.56	0.62	**0.65**
INFOTABS	n.a.	n.a.	0.57	0.51	**0.63**
OUTOFDOMAIN	0.81	**0.84**	0.80	0.80	0.69
SWAPPED	**0.67**	0.65	0.63	0.65	0.58

Table 3.10 shows that TENET generates valid examples independently from the PLM used in the hypothesis generation. This applies with T5, fine tuned on ToTTo [131], and with ChatGPT, with in-context learning. *PLM* creates useful examples, but without the guide of the data evidence and the s-queries, it is prone to hallucinations that degrade the quality of the training data. In other words, generating examples out of the PLM is doable but TENET methods get higher quality. On average, TENET with ChatGPT has slightly better results because of its superior ability in text generation. However, using OpenAI API comes with its own issues, in terms both of data privacy, usage cost, and execution time (Section 3.6.4).

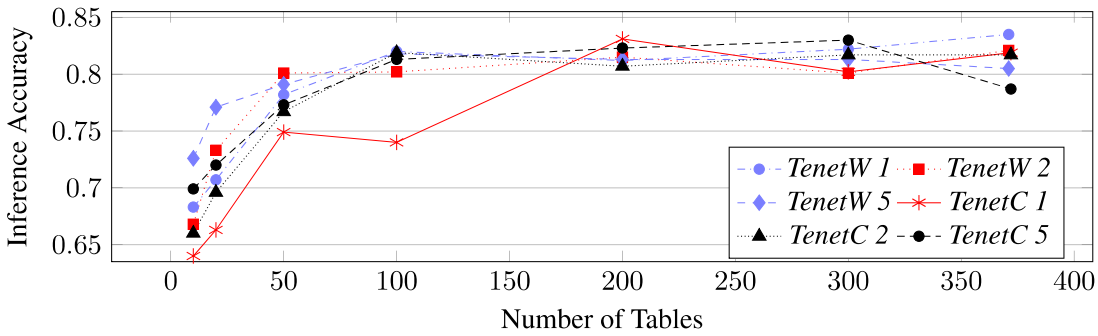

Figure 3.9: Impact of 1, 2, 5 data evidence per table in example generation. FEVEROUS test data, 3 s-queries per evidence.

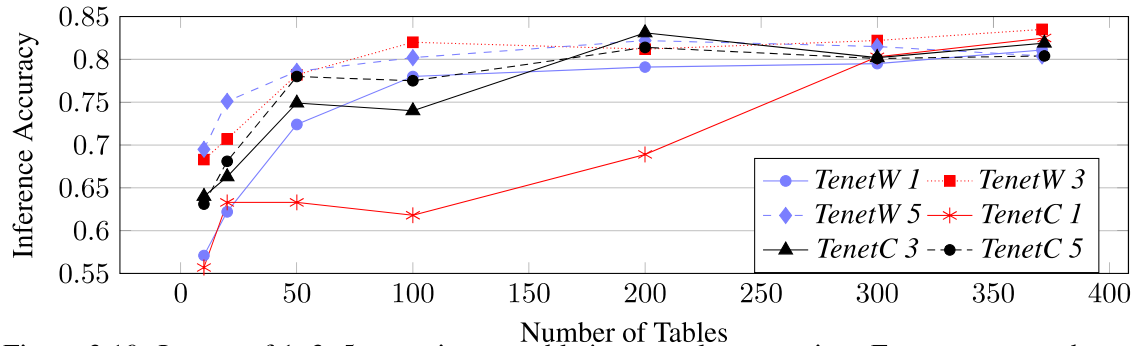

Figure 3.10: Impact of 1, 3, 5 s-queries per table in example generation. FEVEROUS test data, 1 data evidence per table.

Table 3.11: Accuracy results with different thresholds for the # of shuffled attributes in *Refutes* example generation.

Test set	Type	Quality with $\tau = 0.25$	Quality with $\tau = 0.5$	Quality with $\tau = 0.75$
FEVEROUS	Cold	0.74	**0.77**	0.69
FEVEROUS	Warm	0.74	**0.77**	0.74
INFOTABS	Cold	0.54	**0.57**	0.54
TABFACT	Cold	0.50	**0.62**	0.54
TABFACT	Warm	0.52	**0.65**	0.57

Impact of Parameters. Figure 3.9 shows the inference accuracy when varying the number of results taken from the evidence-query for every table. The experiment is run over the FEVEROUS test data, with tables in its training data, and with TENET models that use three s-query for every data evidence. Results show that a larger number of data evidence per table leads to better results with very few tables, but has marginal gain with an increasing number of tables in the training. We explain this behavior with the fact that using examples from more tables is more beneficial than using the multiple examples from the same table. For a trade off for quality and cost of example generation, we set one data evidence as default.

Figure 3.10 shows accuracy results when varying the number of s-queries executed for every data evidence. The experiment is over the FEVEROUS test data, with tables in its training data, and with TENET models using one result from the e-query. Results show that more hypothesis lead to better results on average, especially with small numbers of tables. As a trade-off between cost and quality, we set three s-queries as default.

Impact of different thresholds in refute example generation. To identify the right percentage of attributes to shuffle, we test different threshold τ values (Section 3.5). We use 200 FEVEROUS tables and generate positive and negative examples with TENET. We then train the model and measure the inference accuracy. Results in Table 3.11 show that 50% leads to the best quality.

Table 3.12: Accuracy results with two additional s-queries.

Test set	Type	Standard S-Queries	New S-Queries	Improvement
FEVEROUS	Cold	0.77	0.78	+0.01
FEVEROUS	Warm	0.81	0.83	+0.02
INFOTABS	Cold	0.59	0.59	0
TABFACT	Cold	0.65	0.63	-0.02
TABFACT	Warm	0.65	0.66	+0.01

Impact of new s-queries. To define the impact of adding new s-queries, we extend the set in Table 3.4 with two new s-queries: *ranked*, which uses the RANK() function in Postgres to craft examples such as "John is the second youngest person", and *percentage*, which calculates the difference in % for pairwise numerical values to generate examples such as "Bob earns a salary that is 50% higher than John's". These kinds of examples are present in a small percentage in TNLI corpora. We use such new s-queries over 200 Tables and extend the original TENET training data with the corresponding training examples. Accuracy results in Table 3.12 show that adding examples from the two new s-queries brings a small gain in quality.

3.6.4 Execution Time and Cost

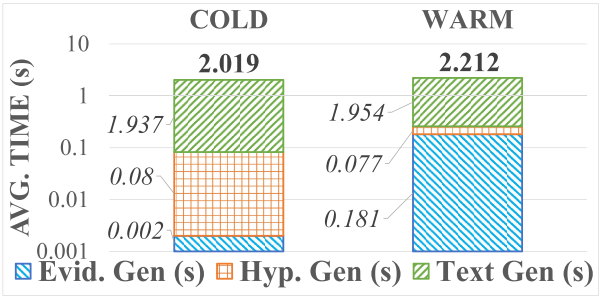

Figure 3.11: Average time for generating one example with Cold and Warm approaches. For each scenario is reported the time taken by each generation step, with total time on the top of each bar. Time in seconds and reported in log scale.

We measure TENET execution time to generate training data. We create five samples of 200 tables from FEVEROUS and execute the full pipeline with Cold and Warm approaches. We report in Figure 3.11 the average time in generating a single training example. We partition the overall time across the generation of the new evidence (blue, bottom), the hypothesis generation (orange, middle), and the text generation with ChatGPT (green, top). The average time does not change significantly between the cold and warm approaches. In the warm approach, more time is spent in the evidence generation. Indeed, generating and executing the e-queries takes more time than random selection. On the other hand, using a seed evidence in the warm approach leads in most cases to more compact evidence, involving a smaller number of attributes compared to random. The cold setting, due to its random nature, involve several attributes, and thus generates more s-queries to check. The most expensive step in our approach (97% of the execution time) is due to text generation. This heavily depends on the ChatGPT availability and it takes on average from 1.5 to 2.2 seconds per request.

Table 3.13: Costs of generating hypothesis with ChatGPT.

	# Tables	# Positives	# Negatives	Total #	Price ($)
Warm	200	1670	1536	3206	11.6
Cold	200	1655	1580	3245	11.7

Table 3.13 reports the costs of generating hypothesis with the OpenAI API and ChatGPT for 200 tables. The cost linearly depends on the number of generated examples, as ChatGPT calculates the costs based on the size of the input prompt together with the size of the generated output. On average the generation of one example costs 0.0037$. The total cost of all the experiments reported in this paper is about $130 for 36K generated examples. Using a smaller PLM, such as T5, on a local machine (Apple M1 Max laptop) does not have any API cost and takes on average 1 sec for the text generation step of one training example. However, the quality of the generated text is slightly lower than ChatGPT.

In conclusion, TENET generates a training example with a lower time and cost w.r.t. those required by human annotators.

3.7 Related works

TENET is a system spanning different problems. We start discussing augmentation and generation of text examples. We then focus on extracting SQL statements from NL text and from query results. Next, we cover text generation from tabular data. Finally, we discuss applications in Tabular Natural Language Inference (TNLI).

Augmentation of Textual Examples. In augmentation, the goal is to provide additional annotated data by modifying existing examples. In one baseline, we use a method that augment examples to create more hypothesis for the same tabular data evidence.

Augmentation can be performed on the data or on the feature space [133]. In the data space, several works operate at the *character* level [145, 146] by swapping, removing, adding letters; injecting common spelling mistakes; or replacing words with abbreviations, e.g., "I'm". Approaches that operate at the *word* level, use word swap/deletion [147, 148, 149, 150, 151] or replacement with synonyms, hypernyms, and antonyms [145]. At the *document* level, a popular method is round trip translation [152, 153]. New textual samples are also created with generative methods [147] and pre-trained language models (PLMs) [154, 138]. For example, by using GPT-2, as a generator, and reinforcement learning to guide it towards specific class labels in the decoding stage [155]. We also use PLMs (ChatGPT and T5) for text generation, but our generation is driven by the relational data.

Other works focus on transforming the feature space, rather than the raw data. Noise addition is used to create new examples by modifying the vectors with the injection of zeros [156] or updating them with random multiplications [157]. An alternative to noise is interpolation, such as combining similar vectors from examples with the same label [158, 159]. This line of work is not applicable in our case where the evidence table data is explicit in the example.

Generation of Textual Examples. For example generation in the unsupervised setting (no examples available), several works focus on exploiting PLMs to obtain textual claims. In *SuperGen* [160], an original text t is combined with a template prompt to obtain a positive, neutral or negative sentence from the PLM, e.g., given sentence t, the prompt for a new negative sentence is "t. However the truth is...". In the supervised setting, humans are asked for hints on the output, e.g., by annotating a taxonomy with related words to train a LSTM model that generates sentences [161]. In another direction, the classifier is trained with examples from a fine-tuned text generator [162] or with examples extracted from Wikipedia paragraphs with Bart models [57] that obtain pairs of (claims, label) [119]. While these works share some ideas with our approach, they cannot consume tables as input.

Semantic Parsing. In the supervised setting, we generate data evidence for new samples from a given example. As we want full control on the data (to distinguish Supports and Refutes), we derive a SQL query for every data evidence. Text-to-SQL (semantic parsing) methods that infer the query from the given hypothesis [163, 164, 165]. perform poorly when executed on factual claims. For instance, RAT-SQL [135] derives a query from a textual NL question and table pair. While it handles datasets with multiple tables and foreign keys [166], it assumes relational tables only, works on questions (not factual claims) as input, and mostly returns incorrect queries in our setting.

Query Reverse Engineering. In this problem, the goal is identify the query that generates a given output. Deriving surface-queries, that overfit on the input, is always possible, while for more general queries the complexity is exponential [137]. However, some methods focus on getting one query for the given example [137, 167], in this case the complexity is in P-time under some assumptions. This is not suitable for us, as we want to find a variety of s-queries to reflect the different kinds of reasoning needed in the inference. Moreover, some of these methods require both positive and negative output examples [137], while we have only (positive) data evidence. Related approaches for query-by-example also propose heuristics for the discovery of sets of possible queries, but the solution is for interactive use [168], while in our case we aim at full control over the variety of s-queries. Finally, given the nature of the corpora in NLTI problem, we do not focus on the inference of joins [169].

Text Generation from Tables. There are works on verbalizing tables to produce sentences that describe

them. *Data-to-text generation* has been traditionally tackled by leveraging domain knowledge and complex grammar rules [170, 171]. Recent breakthroughs in NLP, remove cumbersome sentence and content planning [131]. *R2D2* [172] combines a generator with a faithfulness discriminator for the produced text t to reduce "hallucinations" such as entities appearing in t that are not in the data. *DocuT5* [173] tackles the lack of context in describing data by manually adding table information and foreign keys. In our setting, we use table names, captions, and any document structural information as context. These works lead to fluent sentences, but only in the form of description of the tuples. In analogy to queries, they describe the output of *look-up* operations. We extend these approaches by generating textual claims that describe data retrieved with SQL operations beyond simple look-up, such as aggregates. *LogicNLG* [174] also discusses the requirement of logical operations in the generated text to go beyond the surface realization of a set of cell values. They create a dataset with more complex examples, such as math operations and comparisons, and test sentence generation with several methods. Our work introduces prompts based on few-shots for generative models, such as GPT-3, which perform better than the previous methods.

Tabular NLI. TNLI determines if a textual hypothesis is supported or refuted based on a given premise in tabular format. Applications include computational fact-checking [129], table understanding [127, 30] and assistance in data-centric fields such as finance and healthcare [175]. As an example of existing datasets, *Feverous* is a collection of labelled textual claims generated by a crowd starting from Wikipedia pages [7]. The pipeline for fact checking is composed of a cell retriever (given the claim and the tables) and a veracity predictor (given the claim and data evidence). Similarly to Feverous, the *SemEval-2021 Task 9* [176] has 2k tables on which claims are built for fact verification and cell evidence selection. From the tables, claims for the training set are generated using IBM Watson Discovery and test claims are written by annotators. The claim generation is based on templates and is poor in terms of variety. In *TabFact* [130], the examples rephrase table data with operation on cells, such as count and max, to obtain the claim. One checking method uses a linearized table with a BERT model, while a second method uses Latent Program Analysis. In *InfoTabs* [30], annotators build a dataset with 3 sentences for each table. They test various pre-trained NLI systems on their dataset and conclude that they do not perform well.

3.8 Conclusion

We proposed a generic solution that automatically constructs high-quality annotated datasets for TNLI. Experiments show that given only a table as input, TENET creates examples that lead to high accuracy when used as training data in the target tasks. Even in settings with a small number of tables for the training of the system, TENET produces examples with variety both in the pattern of the data and in the reasoning used to verify or refute the hypothesis.

While TENET is an important first step, there are several research directions still open. First, there are classes of examples in the long tail that are not represented in our generation process. Examples include mathematical operations, such as hypothesis "Mike is 27 years older than Anne". As the number of possible s-queries is large, we envision a solution where s-queries are inferred from hypothesis in annotated corpora, similarly to what we do for e-queries, with a new learning task that extends existing work on semantic parsing [164]. Second, existing corpora contain also examples that span multiple tables or even tables and text, but our e-query generation algorithm must be extended for such settings. In a similar direction, new algorithms for e- and s-queries are needed to generate examples than require joint reasoning over text and tabular data [177]. Third, once models have been bootstrapped with TENET, we could design active learning algorithms to solicit human-written examples that effectively improve performance on the test set.

Chapter 4

Unknown Claims: Generation of Fact-Checking Training Examples from Unstructured and Structured Data

Originally published as: Jean-Flavien Bussotti, Luca Ragazzi, Giacomo Frisoni, Gianluca Moro and Paolo Papotti. *Unknown Claims: Generation of Fact-Checking Training Examples from Unstructured and Structured Data*. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), 15 pages. Association for Computational Linguistics. [178]

We continue our diving, this time orienting our investigations on the mixture of tabular and textual evidence as sources for Synthetic Dataset creation. Taking advantage of multi-modality, we aim to build a framework to build examples that are qualitative, but also versatile. As in Chapter 3, we target the task of Fact-Checking to build and test our Fact-Checking model.

4.1 Introduction

The spread of false information on social media threatens public trust. For example, during the COVID-19 pandemic, misinformation led to vaccine hesitancy, straining public health systems and informed decision-making [179, 180, 181]. Computational fact-checking (FC) is a vital tool for verifying claims against diverse evidence types, including unstructured text and structured tabular data. Diversity increases task complexity, requiring advanced NLP methods to cross-reference information accurately [33].

Traditional FC models heavily rely on training samples manually annotated by experts, who meticulously review and pair claims with corresponding evidence, and intentionally modify claims to create refuting examples. Unfortunately, this process is labor-intensive and time-consuming, which significantly hinders the scalability of FC efforts in adapting to evolving misinformation scenarios [91]. Recent studies have attempted to mitigate these challenges by automating the generation of training examples using question-answering (QA) and entity replacement (ER) algorithms [182, 36]. However, as summarized in Table 4.1, they face limitations that restrict their practical utility:

1. *They fail to integrate precise tabular data with nuanced textual data*, which is often essential for verifying real-world claims [183, 184]; see Figure 4.2.
2. *They are confined to specific domains*, such as biomedicine, due to their reliance on vertical knowledge bases (KBs), compromising their ability to generalize across fields.

To overcome these limitations, we present UNOWN (Figure 4.1),[1] a novel approach that uses pretrained language models (PLMs) to generate synthetic training examples for FC systems, integrating both textual

[1]Pronounced "unknown", the name draws inspiration from the cryptic Pokémon hieroglyphs, reflecting the uncertain factuality label of undisclosed textual claims.

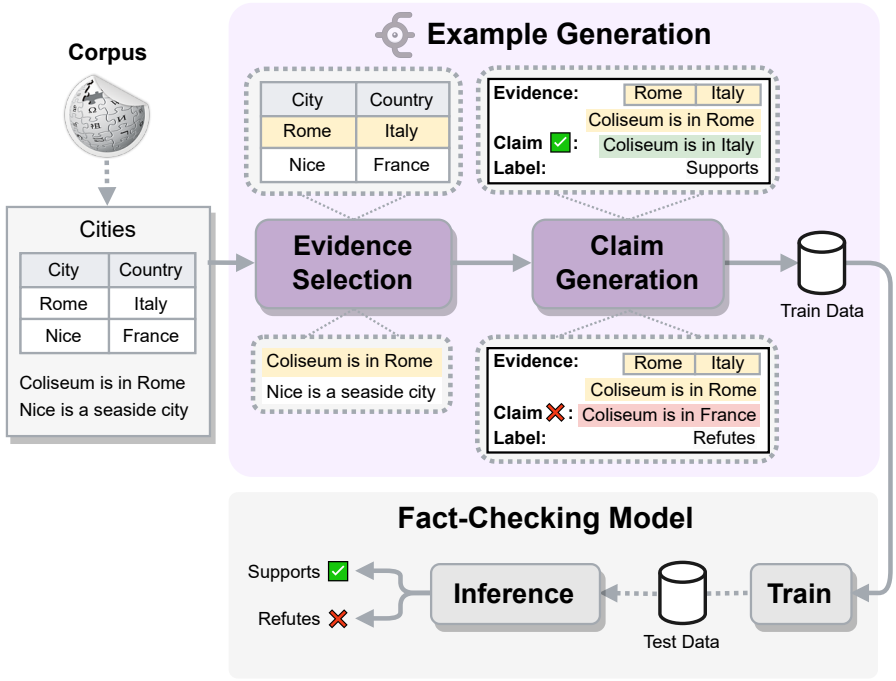

Figure 4.1: UNOWN pipeline. Given a corpus of documents, the *Example Generation* module (investigated in this work) outputs training instances.

Work	U+S[†]	Domain Agnostic	Tested Datasets	Human Eval[‡]
QACG[182]	✗	✓	FEVER [86]	✗
ClaimGen[36]	✗	✗ (biomed.)	SCIFACT [185]	✓
Ours	✓	✓	FEVEROUS [184] SCIFACT [185] MMFC (*new*)	✓

[†] The study combines unstructured and structured data as evidence.
[‡] The study includes human examination of the generated examples.

Table 4.1: Summary of works on the automatic generation of training samples for fact-checking systems.

and tabular evidence. Unlike prior work relying on ER methods and domain-specific data, UNOWN offers a flexible solution that supports multiple evidence selection and claim generation strategies, accommodating small and large language models (SLMs and LLMs). This versatility not only broadens the system's utility across real-world applications but also facilitates its deployment in diverse hardware environments, from low-power devices to advanced computing systems.[2]

We validate our approach by comparing the accuracy of FC models trained on examples generated by UNOWN versus those labeled by humans.[3] To achieve this, we conduct extensive experiments on text-only and text+table evidence scenarios using three public FC datasets targeting general and scientific content: FEVEROUS [184], SCIFACT [185], and MMFC, our new multi-modal and multi-domain fact-checking dataset.[4] MMFC complements FEVEROUS as the second existing corpus featuring textual and tabular evidence, distinguishing it from SCIFACT, which exclusively focuses on text.

The main findings of our study are as follows:

[2]We prioritize using SLMs due to their suitability for resource-limited environments, ensuring broad applicability.
[3]We employ state-of-the-art FC models as they existed at the start of this study (March 2023), without changing their original implementations and hyperparameters.
[4]The dataset is available in the HuggingFace hub: https://huggingface.co/datasets/jeflbu/MMFC.

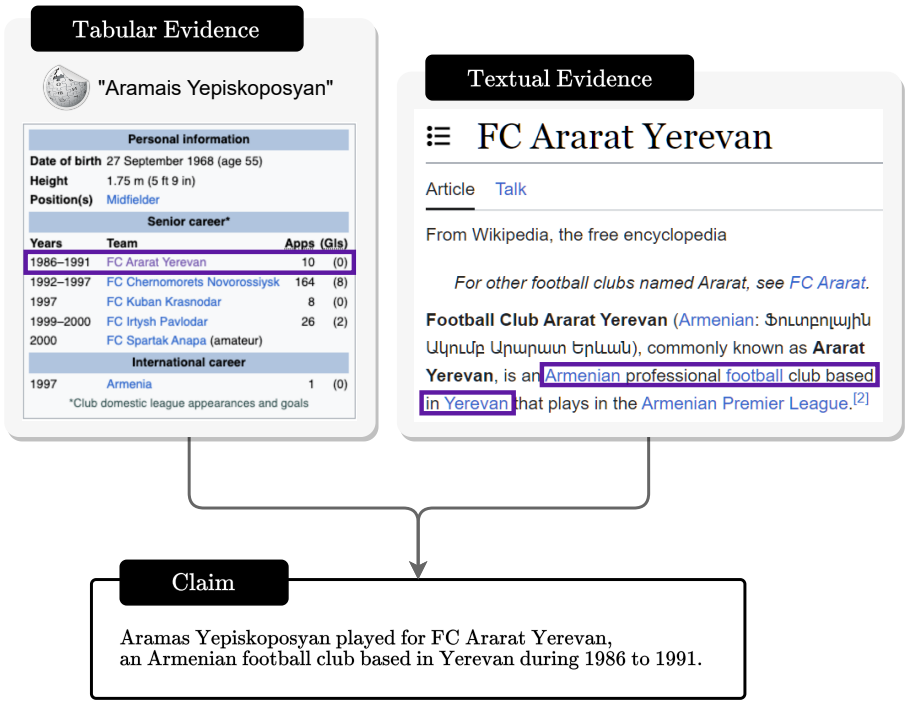

Figure 4.2: Example from the FEVEROUS dataset where the verification of dates reported in the claim requires reasoning above both textual and tabular information.

- In text-only evidence scenarios, training on UNOWN data yields lower accuracy, showing an 8% gap compared to human-labeled samples. However, this gap diminishes to just 2% with the inclusion of only 100 human-labeled instances. Conversely, in text+table scenarios, we achieve up to 5% higher accuracy.
- SLMs and LLMs produce synthetic data of comparable quality, with a 1% gap in downstream FC accuracy.
- By transcending traditional reliance on external KBs, UNOWN adeptly generates refuting claims with sophisticated negation artifacts.

4.2 Problem Formulation

Let $\mathbf{d}$ represent a semi-structured document (e.g., a Wikipedia page) containing n sentences and m tables. We define evidence $\mathbf{e} = \{\mathbf{e}_s, \mathbf{e}_t\}$ as a non-empty subset of sentences $\mathbf{e}_s = \{s_1, \ldots, s_{|\mathbf{e}_s|<n}\}$ and, optionally, cell values $\mathbf{e}_t = \{c_1, \ldots, c_{|\mathbf{e}_t|<p}\}$ extracted from a table within $\mathbf{d}$, where p is the total number of cells. A supervised FC model $\mathscr{F}$ evaluates whether a textual claim c is supported or contradicted by the given evidence $\mathbf{e}$. Specifically, $\mathscr{F}$ takes as input a data pair $<\mathbf{e}, c>$ and outputs a verdict from the set $\mathscr{L} = \{\textit{Supports}, \textit{Refutes}\}$.[5] Consequently, our goal is to automatically generate labeled examples $\mathscr{E} = <\mathbf{e}, c, l \in \mathscr{L}>$ to train $\mathscr{F}$.

Challenge: Refuting Claims There have been proposals to generate artificial claims by synthesizing $\mathbf{e}$ in a sentence. Abstractive summarization has been explored with text-only evidence [186, 36] and scenarios centered on cell values only [187]. In contrast, our goal is to create claims that incorporate evidence from both structured and unstructured data, as illustrated in Figure 4.1. However, while a *Supports* claim naturally aligns with the provided evidence, we also require examples with a *Refutes* label to train FC models effectively, which entails claims that are in conflict with $\mathbf{e}$. Technically, refuting samples should go beyond basic negations such as "Rome is not in Italy." They should instead be adept at capturing

[5]The label *Not Enough Information* is excluded due to its rarity, accounting for only 3% of instances in FEVEROUS.

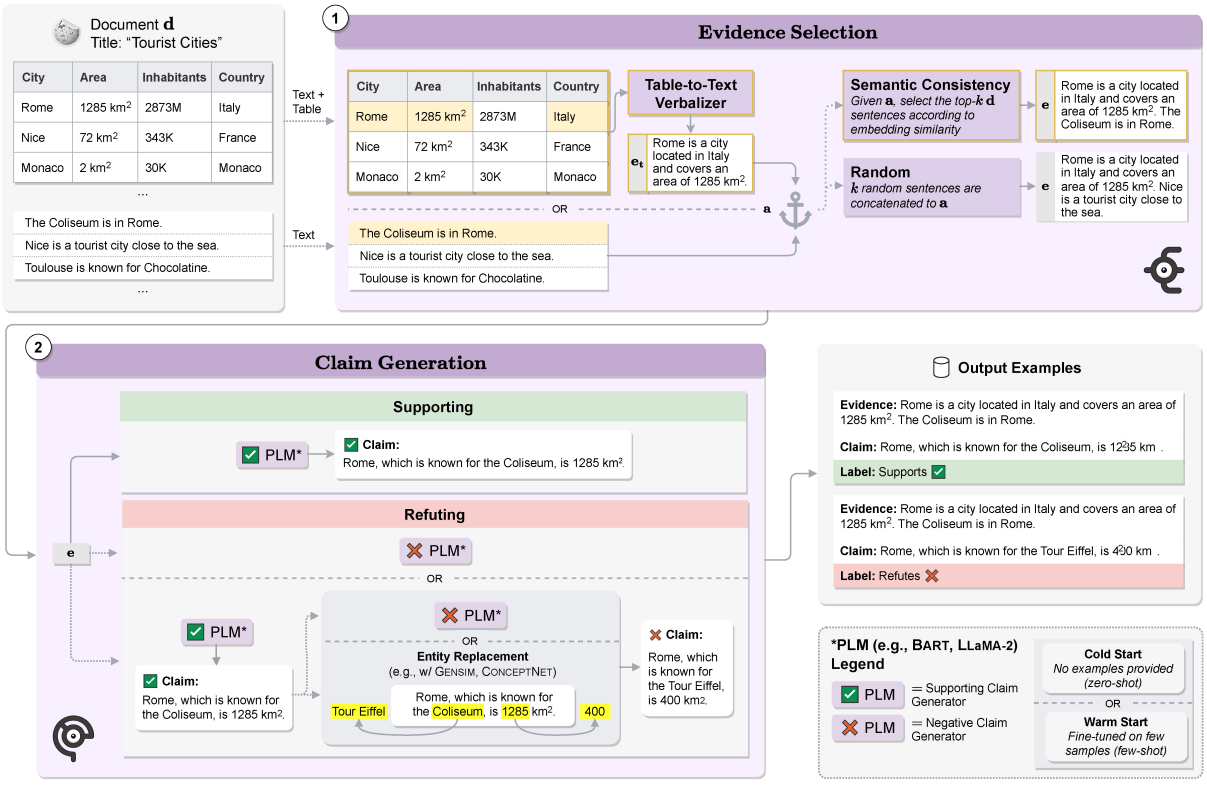

Figure 4.3: UNOWN pipeline. The input document $\mathbf{d}$ consists of sentences and optional tables. (1) When both modalities are used, we obtain $\mathbf{e_t}$ with a cell sampling and verbalization process. From $\mathbf{e_t}$, different strategies can be used to determine $\mathbf{e_s}$ and complete $\mathbf{e}$; in a text-only approach ($\mathbf{e_t} = \emptyset$), $\mathbf{e}$ is established after sentence sampling. (2) We generate supporting and refuting claims using PLMs. Non-continuous lines and arrows delineate alternatives.

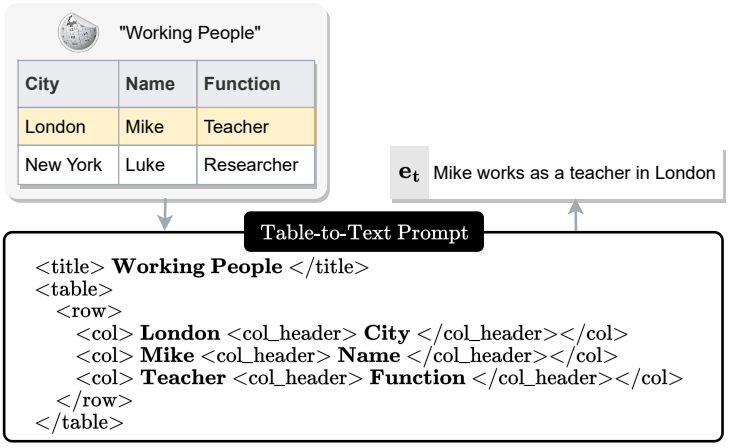

Figure 4.4: Verbalization of a subset of tabular cells.

nuanced factual contradictions, e.g., "Rome is in France", "There are two Coliseums in Rome." Obtaining such variety in claims remains an open research question.

Challenge: Low-Budget Environment In low-resource settings, restrictions such as commodity hardware infrastructure can affect model supervision and performance [188, 189, 190, 191]. In the era of LLMs, the investigation of flexible and scalable solutions is being neglected despite their high social impact [192]. Developing FC systems capable of scaling and adapting to diverse user needs and scenarios is imperative.

4.3 Method

We introduce UNOWN (Figure 4.3), a novel framework to automate the production of FC training data. In a first step, $\mathbf{e}$ is created from the input $\mathbf{d}$ (*evidence selection*). Then, $\mathbf{e}$ is used to generate supporting or refuting claims (*claim generation*).

4.3.1 Evidence Selection

The evidence construction process begins by creating a textual anchor $\mathbf{a}$. We distinguish two settings. **Text-only**: $\mathbf{a}$ is a randomly selected sentence from the document $\mathbf{d}$. **Text+table**: we combine textual and tabular data to determine $\mathbf{a}$. In alignment with the text-centric vision of previous works [193, 194, 195], we fine-tune T5-large (780M parameters) [196] on TOTTO [197], a table-to-text dataset. We sample table cells to generate $\mathbf{a}$ (text) by inference, unifying the data modalities.[6] The prompt uses cell values and includes contextual details such as table headers and the document title to maintain coherence (see Figure 4.4). This approach eases claim generation but still leaves the question of how to select evidence.

Once $\mathbf{a}$ is designated, we propose two alternative strategies to complete the evidence.

Random: we pick k random sentences from $\mathbf{d}$. Here, various topics may exist within $\mathbf{e}$, as the information chosen may not be aligned.

Semantic Consistency: $\mathbf{e_s}$ is constructed by concatenating the k sentences from $\mathbf{d}$ that semantically align the most with $\mathbf{a}$, preserving the topic coherence. As in [198], we use cosine similarity after T5 encoding.

We expand on important clarifications.

1. In text-only scenarios, $\mathbf{e_t}=\emptyset$ and $\mathbf{e}$ consists of a set of sentences. In text+table scenarios, $\mathbf{e}$ comprises sentences and a verbalized representation of tabular cells. We overwrite $\mathbf{e}$ by prefixing the $\mathbf{d}$ title for context with special `<title>` and `<evidence>` token delimiters. Concatenation enables cross-attention among the page title, cells, and sentences.
2. k is drawn randomly from a distribution of $[1, 1, 2, 2, 2, 3, 3, 4, 5]$, selected based on patterns observed in the FEVEROUS training set.
3. We emphasize that constructing $\mathbf{e}$ from $\mathbf{e_t}$ to $\mathbf{e_s}$ using a single verbalization step is the most practical approach, avoiding the complexities of reverse operations.

4.3.2 Claim Generation

Fine-tuning models on data aligned with the target task has proven effective in enhancing performance [76]. Practically, users can expect access to *external* data from related FC applications and a limited number (e.g., 10, 100) of *internal* human samples specific to the task. Given this context, we define the following concepts to guide our methodology.

Warm-start: *external* examples are available for preliminary training (i.e., $\mathbf{e} \rightarrow c$).

Cold-start: no *external* data is available.

Few-shot learning: *internal* examples are accessible for specialized fine-tuning (regardless of warm/cold start).

Refuting Claims Generating refuting claims comes with additional intricacies. We recognize two main paths to avoid introducing a strong lexical bias in artificial training samples, such as basic negation types. *Direct refusal*: we use a PLM that can directly transform $\mathbf{e}$ into a refuting claim, ensuring a direct and straightforward approach. *Two-step approach*: we summarize $\mathbf{e}$ into a supporting claim and then apply a targeted modification to flip its meaning. This involves either using *direct refusal* with the supporting claim or employing ER, where keywords are strategically swapped with antonyms or related terms from a KB.

[6] Cell extraction is a consolidated practice for evidence retrieval in table-based factuality predictors [184].

Dataset	Use Case	Veracity Labels†	Claim Length	Evidence Sent./Cells*
FEVER	Train	10K ✅ / 10K ❌	8.1	2.4/0
FEVEROUS 📄	Test	1.5K ✅ / 1.7K ❌	27.1	2.1/0
FEVEROUS 📄 + ⊞	Test	1.5K ✅ / 0.5K ❌	25.2	1.6/4.4
SCIFACT	Test	0.2K ✅ / 0.1K ❌	12.3	1.8/0
MMFC	Test	0.25K ✅ / 0.25K ❌	21.3	1.5/1.9

† ✅ = supporting claims; ❌ = refuting claims. *Average.

Table 4.2: Dataset statistics.

4.4 Experimental Setup

Our focus is on evaluating the veracity component of the FC process during test time, where models are provided with gold evidence alongside the claim for verification. To achieve this, we address the following research questions:

- Q1 Are the generated artificial examples effective for training FC models?
- Q2 Which evidence selection strategy yields the best performance?
- Q3 What method is recommended for generating refuting claims?
- Q4 To what extent does the efficacy of synthetic examples generalize across various domains?
- Q5 How many internal dataset-specific samples are necessary for few-shot learning to bootstrap the downstream FC model successfully?

Datasets In warm-start scenarios, we use human examples from FEVER [86], a collection of claim–evidence pairs based on Wikipedia, with a balanced of 10K positive and 10K negative instances. As the leading FC benchmark, we take **FEVEROUS** [184], an extension of FEVER with more complex claims enriched with tabular evidence (there is no overlap between the corpora). To assess generality, we include **SCIFACT** [185], a dataset of expert-written claims paired with evidence from scientific papers abstracts. Finally, we release **MMFC**, a new multi-modal FC corpus. Mechanically, we sample 2000 instances from MULTIMODALQA [199], a QA dataset requiring joint reasoning over text, table, and images. In our sampling procedure, we filter out instances requiring visual grounding. Then, we transform each question–answer pair into a claim paired with table+text evidence by performing few-shot in-context learning with GPT-4-TURBO. We carefully review all examples to guarantee their highest quality. Dataset statistics are provided in Table 4.2. More precisely, supporting claims are generated by prompting GPT-4 (`gpt-4-turbo-2024-04-09`) as detailed in Figure 4.12 in Appendix. The examples employed in the few-shot learning process are structured as follows:

- *input* contains the question–answer pair.
- *not optimal output* shows a type of answer to avoid.
- *better output* provides the reference claim.

Refuting claims are generated with the prompt reported in Figure 4.13 in Appendix. A *why* field clarifies the expected negation behavior and makes explicit the difference between the *not optimal output*, *incorrect output*, and *better output* fields.

We conducted in-depth prompt engineering and assessed the claims produced by manual checking of the claims obtained.

Metrics We assess FC predictions using accuracy and F1 scores ($[0, 1]$; higher is better), distinguishing between *Supports* and *Refutes* labels. We validate models on the test sets after training with artificial and human examples. We finally evaluate the logical relationship between each evidence–claim pair with a DEBERTA cross-encoder [200] pretrained on natural language inference (NLI) tasks to classify pairs as *Entailment*, *Contradiction*, or *Neutral*.

Claim Generation Models As SLM, we use models built on BART [57]. For supporting claims, we employ the large version (400M parameters). For refuting claims, we utilize two variants: BART-large

and BARTNEG [58], a specialized BART-base model (140M parameters) trained on parallel and opposing claims from the WIKIFACTCHECK dataset [201].[7] As LLM, we operate with LLAMA-2-7B [66], opting for QLoRA [202] adapter fine-tuning.

Prompt tuning experiments proved the marginal role of few-shot in-context learning. We then opted for a simpler and reproducible zero-shot approach, also fairer to small models, as reported in Figure 4.5.

Claim Generation Prompt

Write a claim that uses the following evidence.
Write a negative claim, i.e., false with regard to the following evidence.

Evidence:
<title> {{**d** title}} <evidence> {{**e**}}

Claim:
y

Figure 4.5: Instruction tuning prompt template for claim generation. The highlighted part is used for loss computation.

We stress that refuting claim generation can be obtained by: (i) running these models directly on **e**; (ii) applying these models to the claim returned by a supporting model. Training is done independently for the two claim types. We run each experiment on a cluster of OS Linux workstations with a single Nvidia GeForce RTX3090 Turbo GPU of 24 GB VRAM. UNOWN is developed using PyTorch [203] and the HuggingFace library [204] (seed set to 42 for reproducibility).

To train BART, we set the following hyperparameters: learning_rate=$1e^{-4}$, batch_size=16, and epochs=20; for LLAMA-2, we use 4-bit nested quantization, r=8, α=32, batch_size=1, and epochs=3. For inference, we adopt beam search (num_beams=5) and nucleus sampling (top_p=0.01, top_k=40, temp=0.15) for BART and LLAMA-2, respectively.

Entity Replacement Methods As a baseline method, and to show the generality of our framework, we adapt the pipeline proposed by [36] to domain-agnostic resources, studying three alternative refuting claim generation procedures. **(1)** We prompt FLAN-T5-large (780M parameters) [205] with *"Answer the following question. Can you give me an antonym of* {{**w**}}*?"*, where **w** is a word of length ≥ 4 characters randomly chosen for replacement. **(2)** We use the GENSIM library [206] to calculate a similarity matrix between the words in the supporting claim.

The matrix is subsequently used to build a frequency ranking, aiding in deciding which word to replace (least common, most common, random). Denoting the chosen item as **w**, words having similarity > 0.7 to **w** are substituted with a similar but distinct word as per WORDNET [207]. **(3)** We use CONCEPTNET [208] to identify a set of concepts closely related to each word in the claim. We build a claim-level frequency ranking on the intersection of word-level concepts. Then, we replace **w** according to `antonym` relationships.

Fact-Checking Models We assess the impact of our synthetic training examples on accurately predicting the verdict label of an input claim given a set of evidentiary sentences. To achieve this, we choose optimal classification models for the benchmarks at hand, keeping their weights and hyperparameters unchanged. For FEVEROUS and MMFC, we use ROBERTA [141] with a linear layer on top. For SCIFACT, we employ MULTIVERS [209] with a shared encoding of the claim and input context.

4.5 Results and Discussion

4.5.1 Quality of Generated Claims

SLMs. We measure how well the UNOWN examples generated with small models contribute to training a downstream FC system on the FEVEROUS test set (Figure 4.6). In the worst-case scenario (cold-start,

[7] Although BARTNEG has already undergone a warm-start process, applying warm start with FEVER is still necessary to deal with multi-sentence input and language style adaptation.

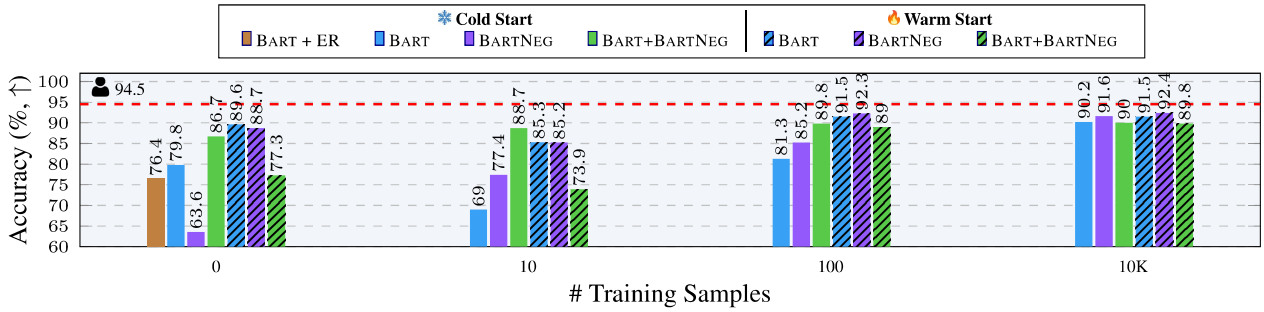

Figure 4.6: Accuracy scores on FEVEROUS by varying the number of its training samples. Dashed bars indicate the use of fine-tuning on FEVER. The horizontal red dashed line represents the accuracy obtained by human data.

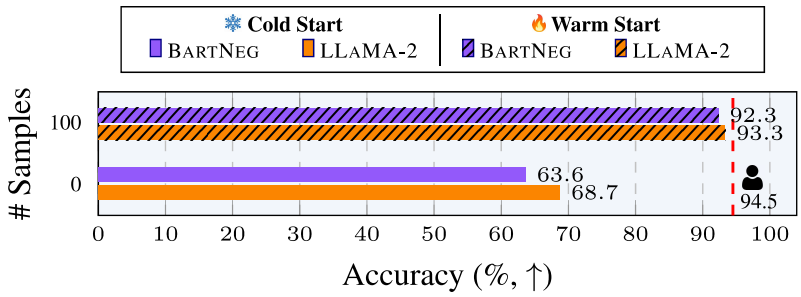

Figure 4.7: Accuracy scores on FEVEROUS with training examples generated by LLAMA-2.

zero-shot learning), the highest accuracy achievable by UNOWN is 86.7 with BART-large used for the generation of both supporting and refuting claims. The results show a consistent boost in performance by leveraging human training instances. In fact, accuracy climbs to 92.3 with warm start and just 100 internal target examples, using BART-large for supporting claims and BARTNEG for direct refusal—close to the accuracy achievable with human-annotated data (94.5).

LLMs Figure 4.7 looks at how the claims generated by LLAMA-2 stack up against those inferred by the best SLM setup. The accuracy propelled by LLAMA-2 claims, after training on 100 internal examples, is 93.3, outperforming the small solution by a single point. Therefore, incorporating LLMs does not appear essential in the UNOWN pipeline, favoring BART-based models for their superior effectiveness–efficiency trade-off.Table 4.3 reports the train and inference time per claim for the claim generation task. The benefit of smaller models is evident during inference. We also report the average time required to generate an example in terms of evidence selection. The total time of about 6 seconds per claim is in contrast to the time and effort required by a human to craft a comparable example.

NLI To gain additional insight into the generated claims, we compute the NLI prediction score between claims and evidence. Table 4.4 shows that, for supporting claims, UNOWN's examples closely resemble the score distribution in their human-written counterparts. Yet, in the refuting examples generated by UNOWN, the percentage of entailed claims surpasses that of human-generated ones, highlighting the greater difficulty in creating refuting examples compared to supporting ones. We observe that the ER baseline performs the worst.

Human Evaluation We perform a qualitative analysis to investigate the quality of the claims generated by UNOWN. We randomly sample 50 instances from the FEVEROUS training data (25 supporting, 25 refuting). Taking into account the expense associated with careful human evaluation and the central role

Model	Task	sec/Claim
Claim Generation		
BART	`Train/Infer.`	1.92 / 0.12
BARTNEG	`Train/Infer.`	1.01 / 0.08
LLAMA-2	`Train/Infer.`	1.98 / 2.10
Table-to-Text		
T5-TOTTO	`Infer.`	0.75
Evidence Selection (Semantic Consistency)		
T5	`Tokeniz.` + `Distance`	5.43

Table 4.3: Time consumption for different tasks.

	Method	Entail.†	Contrad.‡	Neutral
	☑ `Supporting`			
👤	HUMAN	75.00	4.00	21.00
🤖	BART	74.57	4.90	20.53
🤖	LLAMA-2	71.92	3.95	24.12
	✖ `Refuting`			
👤	HUMAN	3.00	77.00	30.00
🤖	BART	10.43	56.00	33.57
🤖	LLAMA-2	11.23	37.82	50.95
🤖	ENTITY REPLACEMENT	36.05	40.70	23.26

† $[0, 100]$. ☑ : ↑ (higher is better). ✖ : ↓ (lower is better).
‡ $[0, 100]$. ☑ : ↓ (lower is better). ✖ : ↑ (higher is better).

Table 4.4: The quality of the generated claims in FEVEROUS based on NLI scores (text-only scenario).

of text as our unified modality, we accord priority to text-only evidence. Each instance is presented with its original human-selected evidence and the corresponding claim. To maintain fairness, we condition our models on this evidence and generate synthetic claims using our best-performing models: the warm-started BART-large for supporting claims and BARTNEG for refuting claims. After manually verifying the correctness of the assigned label, which were accurate for all 50 claims, we enlist the expertise of three external annotators with strong NLP and FC backgrounds to evaluate the claims. In a blind review process, we provide them with the evidence and the two claims (original and generated) in randomized order. Following a direct comparison assessment, which has proven to be more reliable and sensitive than rating scales [210, 211], we ask the annotators to determine which claim is the best with respect to two dimensions: *clarity* (effective communication of the intended meaning with a good sentence structure, fluency, and English precision) and *coherence* (semantic connection to the evidence). They are also given the option to declare a tie if they perceive the quality of the claims to be comparable. To aggregate the annotations, we employ a majority voting approach and calculate Cohen's κ coefficient to gauge the agreement between annotators and the majority voting label. The coefficient value of 0.613 indicates a substantial level of agreement, enhancing the reliability of our analysis. As illustrated in Figure 4.8, the results reveal an interesting landscape. Out of the 50 paired claims, annotators found 35 to be of comparable quality. In 10 cases, the original FEVEROUS claims were deemed superior, while in 5 cases, the claims generated by UNOWN were judged to be of higher quality.

Overall, the generated examples prove to be sufficiently effective for training FC models, yielding quantitative results in a 2-point margin in absolute accuracy compared to those achieved by a crowd of annotators.

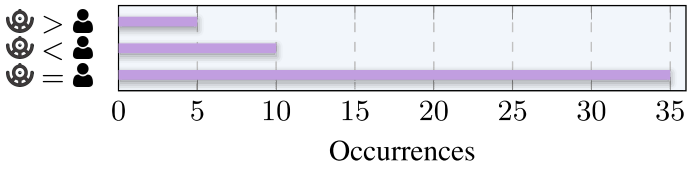

Figure 4.8: Human evaluation results on 50 claims.

Method	Text Acc.	Text F_1 ✅	Text F_1 ❌	Text+Table Acc.	Text+Table F_1 ✅	Text+Table F_1 ❌
FEVEROUS						
EVID. + CLAIM	94.50	94.92	95.30	82.09	87.83	66.12
EVID. + C	92.40	92.10	92.70	84.59	90.53	58.69
RAND. w/ GOLD + C	92.30	91.70	92.70	81.81	87.96	62.89
RAND. w/o GOLD + C	85.11	86.07	84.01	84.43	89.90	66.03
SEM. CONSIST. + C	88.33	88.21	88.44	86.83	91.73	67.65
MMFC						
EVID. + CLAIM	.	.	.	87.60	88.17	86.97
EVID. + C	.	.	.	76.00	75.61	76.38

Table 4.5: Evidence selection comparison in FEVEROUS and MMFC. Methods use BART and BARTNEG to create supporting and refuting claims, respectively. Models fine-tuned on FEVER and then on 100 dataset samples.

4.5.2 Q2 : Evidence Selection

We study the impact of alternative evidence selection methods. We report two experiments using FEVEROUS training examples: one with text-only evidence and another with text+table evidence; test datasets are filtered according to the scenario. For every human example, referred to as "gold," we execute our best BART model with four alternative evidence selection strategies. **Human evidence**, where we use the original evidence handpicked by the annotators. **Random with gold**, where the number of selected sentences matches the human example, but the actual cells and sentences are chosen randomly from $\mathbf{d}$. **Random without gold**, where the number of retrieved sentences k, after anchor definition, is drawn from the distribution presented in Section 4.3.1. **Semantic consistency**, where textual evidence is retrieved using embedding similarity to the table verbalization (see Figure 4.4).

Table 4.5 shows accuracy and F1 results. The influence of evidence is evident. The use of human evidence allows UNOWN to produce examples that match nearly the human upper bound. In the text+table setting, we achieve even higher scores for supporting claims, confirming the quality of our claim generator. In the text-only scenario, performance is optimal when guided by the cardinality of human gold evidence, with random selection surpassing semantic consistency. In text+table, semantic consistency outperforms both random selection and original human examples in all metrics. We observe that human annotators struggle to annotate tabular data accurately, making mistakes that mislead the classifier. This is also reflected in the generally lower results for text+table compared to the text-only scenario.

Table 4.5 also shows the results for MMFC. In this dataset, all claims involve text and tabular data and we only have human gold evidence for the original claim. We explain the lower quality results for UNOWN because the warm start includes examples from FEVER, which are different from those in MMFC. Tables 4.7, 4.8, and 4.6 report examples of textual claims generated by our system with different models given the same original evidence. The human-written claim is provided for comparison. We note that many claims generated by BARTNEG with *Refutes* labels do not contain the word "never". To illustrate:

Evidence (d title = Stefan Lessard)	
Lessard was born in Anaheim, California, to musician Ron and Jacqueline Lessard. During the 2007 European Tour, Lessard *also began* a video blog on YouTube. In October 2005 he married Jaclyn Burton, and together they have *three daughters*, Hazel Bailey (born March 6, 2004), Flora Jude (born July 13, 2006), and India (born November 2015).	
✗ Refuting Claims	
HUMAN	Stefan Kahil Lessard (born on June 4, 1974 **outside** Anaheim, California) completed his junior year of high school, passed the GED examinations in the summer of 1991, and was accepted into the Jazz Studies program at Virginia Commonwealth University.
LLAMA-2	Stefan Lessard has been married to Jaclyn Burton since October 2005. They have **three sons** named Hazel Bailey (born March 6, 2004), Flora Jude (born July 13, 2006) and India (born November 2015).
BART	Stefan Lessard **never began** a video blog on YouTube.

Table 4.6: Example from FEVEROUS of UNOWN's refuting claims, highlighting negation artifacts (bold) with respect to evidence excerpts (underline).

Evidence (d title = Windecker Eagle)	
The Eagle AC-7 Eagle 1 (USAF designation YE-5) is an aircraft that was manufactured by Windecker Industries Data from Jane's All the World's Aircraft 1971.	
☑ Supporting Claims	
HUMAN	The Eagle AC-7 Eagle 1 is a military aircraft that was manufactured by Windecker Industries.
LLAMA-2	Windecker Eagle is listed in Jane's All the World's Aircraft 1971-72.
BART	In 1896, Friedrich Baume established a legal practice with A. E. Whitaker, a son of Frederick Whitaker, and won the election.

Table 4.7: Example from FEVEROUS of UNOWN's supporting claims with textual evidence.

Tabular Evidence (d title = 2010 FedEx Cup Playoffs)	
Bubba Watson [Header : Player]	
Textual Evidence (d title = Bubba Watson)	
Watson was born and raised in Bagdad, Florida, near Pensacola. He played on the golf team at Milton High School, which had featured future PGA Tour members Heath Slocum and Boo Weekley just before he attended. Watson played golf for Faulkner State Community College in nearby Baldwin County, Alabama, where he was a junior college All-American. He transferred to the University of Georgia, the defending NCAA champions, and played for the Bulldogs in 2000 and 2001. As a junior, Watson helped lead the Bulldogs to the SEC title in 2000.	
☑ Supporting Claims	
HUMAN	The player who had 1,498 points in the regular season of the 2010 FedEx Cup Playoffs attended Milton High School.
BART	Bubba Watson Watson played in the 2010 FedEx Cup playoffs. Watson was born and raised in Bagdad, Florida, near Pensacola. He played on the golf team at Milton High School, which had featured future PGA Tour members Heath Slocum and Boo Weekley

Table 4.8: Example from MMFC of UNOWN's supporting claims, with tabular and textual evidence.

- *"In the 2006-07 San Jose Sharks season, the team scored 107 goals, 183 assists, and 1 Shutout."* Here, the real numbers are 107, 283 and 5.
- *"Karyn Kupcinet, who died on June 2, 1963, appeared on The Donna Reed Show and The Gertrude Berg Show, 1999."* Here, the actual day is November 28, 1963.
- *"Rihanna had a live performance at the Super Bowl in 2012."* Here, the actual singer is Madonna.

These examples showcase the variability of our generated claims, ensuring that the models trained on our data must learn robust patterns beyond simple negations and manage hard negative cases from a semantic viewpoint. Additionally, we acknowledge the presence of several generated claims with *Supports* labels that contain the word "never", further requiring the ability to capture diverse linguistic patterns. For instance *"Bruce Johnston's song 'I Write the Songs' never charted."*

We conduct an ablation study aimed at evaluating the importance of each evidence modality for table+text FC instances (Table 4.9). When text or cells are excluded from the evidence in the test data, accuracy and F1 scores for the FC model drop significantly.

4.5.3 Q3 : Refuting Claims

We show how the FC performance varies with different types of *Refutes* generated claims in a quantitative analysis and then in a qualitative user study.

Test set	Accuracy	F_1 ✅	F_1 ❌
STANDARD TEST DATA	86.9	91.7	67.6
ABLATED TABLES	57.6	66.0	43.9
ABLATED SENTENCES	62.8	71.5	46.3

Table 4.9: Results on three different test sets: the gold test set, the same test set with ablated tables in evidence, and the same test set with ablated sentences in evidence. Training data is always based on the warm start and the BART/BARTNEG combination.

Quantitative Figure 4.9 shows how we identified FLANT5 and random selection as the best combination for the ER method used as our baseline approach.

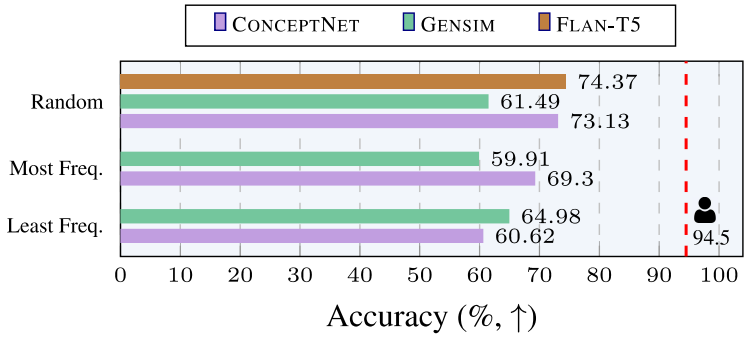

Figure 4.9: Comparison of different entity replacement methods in FEVEROUS.

Unless otherwise specified, we use ER to denote this baseline approach. Figure 4.6 includes the impact of various negation strategies on the accuracy of the target task. In cold start, the combination of BART and BARTNEG using the two-step approach is effective, while the results are subpar with ER, which fails to make refuting claims, possibly due to limitations in content replacement without adequate rewording. As anticipated, starting with a warm start is beneficial, resulting in the highest accuracy with 0 and 100 training samples.

Qualitative We perform a human analysis to evaluate the negation techniques used to refute claims. We adhere to the negation taxonomy outlined in previous studies [212, 213]. Rigorously, we use two main negation types, namely *Verbal Negation* (V) and *Noun Phrase Negation* (NP). Each is classifiable in three subclasses, including *Lexical* (L), where the negation is expressed with new words or phrases that alter the sentence meaning (e.g., 10 papers→**more than** 10 papers), *Morphological* (M), where the form of the word is modified through morphemes (e.g., legal→**il**legal), and *Replacement* (R), where a phrase is swapped for another with a different meaning (e.g., 1995→1997). Given these classes, three annotators (selected among the authors) evaluated 30 refuting claims from the original FEVEROUS training dataset and 30 refuting claims generated by UNOWN. The final category is identified by majority voting over the three suggested labels; the Cohen's κ coefficient is 0.91, which shows very high agreement among annotators. The results of the study are illustrated in Figure 4.10, allowing a comparison of annotation distributions between the two sets of examples (UNOWN vs. human). UNOWN produces an even distribution of refuting claims, encompassing both noun phrases and verbal structures, whereas humans tend to prefer noun phrases. Both UNOWN and humans favor the replacement strategy for noun phrases and the lexical strategy for verbs. In both scenarios, the ranking of classes and subclasses remains consistent, indicating that UNOWN produces a range of negation types comparable to those observed in a human-crafted corpus.

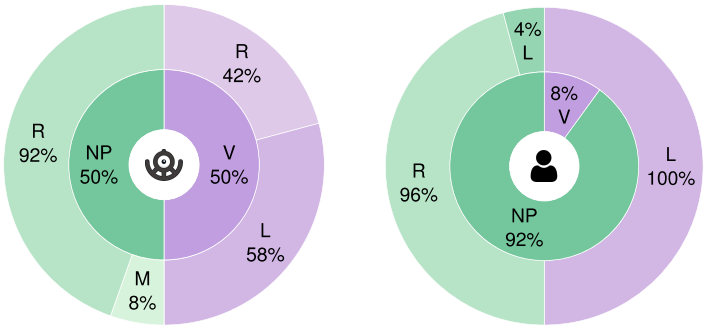

Figure 4.10: Human annotation on negation artifacts.

	Method	Accuracy	F_1 ✅	F_1 ❌
	HUMAN TRAINING DATA	84.62	81.05	85.71
	ENTITY REPLACEMENT	55.33	57.27	16.51
	BARTNEG	65.08	53.79	65.96
	ENTITY REPLACEMENT	54.00	57.63	1.98
	BARTNEG	74.50	73.53	73.45

Table 4.10: Strategies for refuting claim generation on SCIFACT; models use BART to create supporting claims. In warm scenarios, models are fine-tuned on FEVER.

4.5.4 Q4: Checking Scientific Claims

We measure the quality of the FC system trained with UNOWN examples in a different domain. Due to the lack of heterogeneous datasets such as FEVEROUS, we use the text-only scientific corpus SCIFACT. Table 4.10 confirms the analysis outcome on FEVEROUS. Human data achieve the best results, followed by UNOWN with the warm-started BART. We explain the greater result gap between humans and UNOWN because the warm start includes only examples from FEVER. Again, BARTNEG leads to better results with respect to ER. We posit that low F1 refuting scores (i.e., 1.98, 16.51) stem from FLAN-T5's pre-knowledge bias, which may not adequately align with scientific subjects.

4.5.5 Q5: Bootstrapping: Cold vs. Warm Start

We measure the impact of the examples used to fine-tune the models. As shown in Figure 4.6, Figure 4.7, and Table 4.10, a warm-start approach improves the quality of the generated data. More precisely, Figure 4.11 shows the average Δ accuracy improvement when shifting from cold to warm in FEVEROUS. We observe a decrease in Δ as the number of internal samples from the target dataset increases, highlighting the beneficial contribution of using external related data as a guide source of knowledge. Also SCIFACT exhibits an increase in accuracy for BARTNEG in the warm approach.

4.6 Related works

Computational FC has been an active area of research for decades [214, 33]. Recently, the rise of LLMs has advanced the development of FC pipelines [215], but their effectiveness is still inferior to human experts [216, 217]. Specialized models are currently the most effective approach [218], despite requiring for large labeled datasets for training. Existing approaches to automatically generating FC training examples has been approached through both unsupervised and supervised methods. Unsupervised

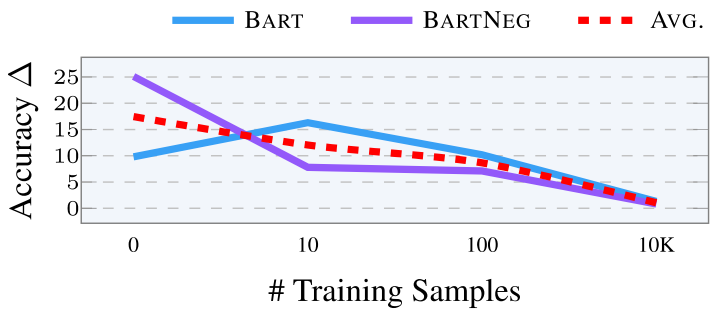

Figure 4.11: The FEVEROUS's average Δ accuracy improvement when shifting from cold to warm.

solutions, typically employed in the absence of labeled data, leverage PLMs to create textual claims from a given text, e.g., by using template prompts [219]. Supervised approaches rely on specific resources, e.g., an annotated taxonomy to train an LSTM model for sentence generation [220]. Several works have investigated the generation of claims from textual evidence (see Table 4.1). [182] produce question–answer pairs using answer replacement to assemble the refuting claim. [36] create supporting claims with a generative PLM and ER over a domain-specific KB for evidence refusal in the biomedical field. Research on generating claims specifically from tabular data remains limited. While some studies have explored template-based methods [176, 221], [187] demonstrated improved results by generating claims based on human-provided examples. To our knowledge, no existing work has addressed the generation of training examples by integrating both structured and unstructured data as input.

4.7 Conclusion

We introduced UNOWN, a domain-agnostic framework to automatically generate training examples for fact-checking systems, bypassing the costly task of manually annotating large volumes of data. UNOWN fits both structured and unstructured data to compile textual claims that support or refute the evidence provided. It also accommodates several solutions for evidence selection and claim generation to adapt to different scenarios. We evaluated our framework using three datasets that deal with general-domain and scientific contexts. The results indicate that our synthetic examples exhibit a quality comparable to that of expert-labeled data, showing the practicality and efficacy of our framework. Quantitative and human evaluation also register that our refuting examples have high variety, comparable to human-generated ones.

Limitations

Although UNOWN is a promising step forward, some research directions remain unexplored. First, our generation process lacks coverage of certain examples within the long tail, e.g., mathematical operations, such as the premise "Paul is 2 years younger than Mary." We consider using a solution in which more intricate patterns are generated as queries over relational tables [187]. Second, once models have been trained with instances from UNOWN, we could set up active learning algorithms to guide our methods in generating examples that effectively enhance performance on the test set [222]. Finally, the considered datasets include well-crafted claims, but real claims can be incomplete (i.e., lacking context)—with ambiguity in the text with respect to the evidence [223]—or require multi-modal evidence that goes beyond text and tables [100].

Question/Answer to Claim Prompt

SUPPORTS

Can you make a claim out of this Question/Answer pair? Your answer should only contain the claim. You should add no other information.
Here are some examples of things not to do :
Input : Is the religion with 16.27% of the Canadian Census of 1871 the same religion as the Church of England? No
Not optimal output : The religion constituting 16.27% of the Canadian Census of 1871 is not the Church of England.
Better output : The religion constituting 16.27% of the Canadian Census of 1871 is a religion other than the Church of England.

Input:Which team was Sebastian Svärd on in 2004-05 that played in the 2017 FA Cup final? Arsenal
Not optimal output : Sebastian Svärd was on the Arsenal team in 2004-05.
Better output : Arsenal, the team Sebastian Svärd was on in 2004-05, played in the 2017 FA Cup final

Input:
Question/Answer
Claim:
y

Figure 4.12: Prompt for the generation of supporting claims from question–answer pairs in MMFC.

Question/Answer to Claim Prompt *REFUTES*

Can you make a refuted claim out of this Question/Answer pair? Your answer should only contain the claim. The claim should not be based on basic negation

Here are some examples of things not to do and why:
Input : Is mobil 1 the official sponsor for the constructor that had a time/retired of electrical in the Australian Grand Prix race of 2016 f1 team? Yes
Not optimal output : Mobil 1 was not the official sponsor for the constructor that had a time/retired of electrical in the Australian Grand Prix race of the 2016 F1 team
Why : The boolean answer should not cause a poor negation, containing a simple negation
Better output : Google was the official sponsor for the constructor that had a time/retired of electrical in the Australian Grand Prix race of the 2016 F1 team

Input : in the Season victories of 2017 Astana season, where was the grand depart for the 2017 Race when the Location was La Planche des Belles Filles city and country? Düsseldorf, Germany
Not optimal output : The 2017 Race grand depart from Astana season in La Planche des Belles Filles was in Düsseldorf, Germany.
Why : the way the claim is refuted is too subtle
Better output : The 2017 Race grand depart from Astana season was in Paris, France.

Input : When did the home team that had a score of 20-34 in round 7 of the 2018 NRL season enter the NRL? 1988
Incorrect output : The home team that scored 20-34 in round 7 of the 2018 NRL season entered the NRL in 1988.
Why : the generated claim is not false with regards to the question/answer pair
Better output : The home team that scored 20-34 in round 7 of the 2018 NRL season entered the NRL in 1975.

Input : For the religion that has 3,304 females in the Moscow Governorate, what is its primary literary work? Talmud
Incorrect output : The religion with 3,304 female adherents in the Moscow Governorate predominantly follows the Talmud as its primary literary work.
Why : the generated claim is not false with regards to the question/answer pair
Better output : The religion with 3,304 female adherents in the Moscow Governorate predominantly follows the Bible as its primary literary work.

In any case, the text you generate must be false in light of the initial question/answer pair.

Input:
Question/Answer
Claim:
y

Figure 4.13: Prompt for the generation of refuting claims from question–answer pairs in MMFC.

Chapter 5

Explaining The Role of Evidence in Data-Driven Fact Checking

Under submission at ARR (NAACL 2025): Jean-Flavien Bussotti, Andrea Baraldi, Francesco Guerra and Paolo Papotti. *Explaining The Role of Evidence in Data-Driven Fact Checking*.

After the exploration of Synthetic Datasets in a uni modal and in a multi modal setting, we now target explainability of models. Instead of creating training data for models,we want to understand how fact-checking models utilize them to make decisions. Explaining reasons behind a decision are important to convince a user. More than simply understanding, we craft an explainable module for existing fact-checking models.

5.1 Introduction

Online misinformation endangers democracy and trust in news sources. While fact-checking mitigates this, human-based methods are resource-intensive and cannot scale with the rapid content influx on social media [224].

Automated methods have been proposed to address this scalability challenge, offering cost-effective alternatives. Computational fact-checking involves establishing the relationship between a textual claim and a body of evidence to determine if the evidence supports, refutes, or is insufficient [179]. Evidence can be structured tabular information, unstructured text, or both [224]. Evidence-based solutions are the most common and effective approaches for this problem.

Such computational solutions rely on a ML pipeline, where, given a claim to check, a first model (*retriever*) identifies relevant evidence from a corpus of documents and a second model (*verifier*) evaluates the veracity of the claim using such evidence. Table 5.1 illustrates an annotated example from a fact-checking dataset. The retriever returns both evidence that support the claim (two sentences in green) and refuting evidence (the one in red, stating that the production rights and certification are held by Wei Hang, not Windecker Industries). It also retrieves noisy content that is not useful to the verifier's final decision (in gray).

Correct labeling is crucial but insufficient for combating misinformation. Most solutions act as black-boxes: they provide accurate inferences but do not explain the reasoning behind the decisions. However, fact-checkers reject solutions that do not expose the evidence justifying the model's output [129]. Transparency is essential for users to verify the evidence and decide if they agree with the model [224].

Our goal is not just to identify relevant evidence for the given claim, as done by re-rankers in the retrieval stage [40], but to determine how pieces of evidence influence the verifier's claim classification. Our solution aims at labeling evidence similarly to the colored annotations in Table 5.1. Inference is not enough: it is essential to explain the reasoning behind a verifier by determining the key piece of evidence in supporting, refuting, or stating that a claim lacks sufficient information for verification. In

CLAIM	The Eagle AC-7 Eagle 1 is a military aircraft that was manufactured by Windecker Industries.
EVIDENCE	The Eagle AC-7 Eagle 1 (USAF designation YE-5) is an aircraft. Windecker Industries was an American aircraft manufacturer founded in 1962. It was manufactured by Windecker Industries. [...] Wei Hang holds the rights and the type certificate to produce the aircraft.
LABEL	SUPPORTS

Table 5.1: Annotated example from the FEVEROUS dataset with a claim, its retrieved evidence (supporting in green, noise in gray and refuting in red), and the corresponding ground truth label.

this direction, we propose a framework that explains the workings of a fact-checking model by scoring individual pieces of evidence based on their contribution to the decision-making process.

The framework enables users to better understand the results of a fact checking tool. In particular, we derive explanations of the individual examples that can be aggregated in a global explanation of the model. Explanations allow us to address the following three questions:

- Q1: What are the evidences that steer the decision? Can the verifier discern noise, identifying pieces of evidence that are important in the claim evaluation?
- Q2: To what extent the verifiers rely on the evidence and the claim to perform the decision? Is this reliance determined by whether a claim is supported, refuted, or deemed to have insufficient evidence?
- Q3: How is distributed the impact of the different pieces of evidence on the verifier? Do verifiers use all the evidence?

This work explores these questions using post-hoc methods explanations methods (Section 2) over four popular datasets and three verifier models, two of which have their own intrinsic explainer (Section 3). Our experimental campaign demonstrates that post-hoc methods enable analysis of every verifier, including those based on fine tuning pre-trained language models (PLMs), with quality of the evidence-based justifications comparable to intrinsic explainers (Section 4).

Related Work. Unlike approaches focused on creating inherently explainable verification models [225] (intrinsic explainability), our work aims to provide explanations for decisions made by black-box models (post-hoc explainability). In contrast with some attempts to provide textual justification for the verdict [41], we emphasize the need to explicitly enable the users to check how the input evidence contributes to the prediction.

Several verifiers come with intrinsic explainers. Some systems decide each piece of evidence's usefulness before making a claim label prediction [44]. The evidence label is thus not based on how a fact-checking model used it, but on a LLM's decision on how pertinent the evidence could be to label the claim. Other approaches imply the training of an explainer[226, 227], while the post-hoc methods do not require training. These systems cannot provide information on the role played by the evidence in the model decision, i.e., our approach measures how an evidence piece steers a claim's labels towards *'Refutes'*, while another piece steers it toward *'Supports'*. Neither they can be used out of the box on any verifier.

We focus on popular verification models based on pre-trained language models, which have demonstrated competitive performance in practice and the best trade off when considering also cost/energy issues [106].

For our study, we use two local post-hoc attribution methods, LIME [120] and SHAP [228], which explain a prediction by measuring how different parts of the input (the evidence, in our setting) contribute to the prediction. The first randomly removes features of an input example and trains a linear interpretable

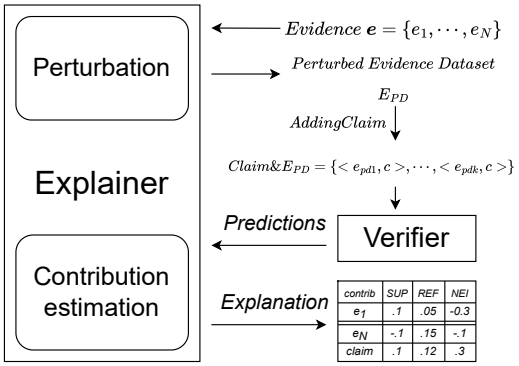

Figure 5.1: The explanation framework.

surrogate model to predict the model's output for these perturbations. The second attributes the contribution of each feature to the model's prediction, via dataset perturbations created with cooperative game theory. Both methods enables practitioners to explain and interpret machine learning models [229].

5.2 The Framework

The Fact-checking Problem. We define evidence $\mathbf{e} = \{\mathbf{e}_s, \mathbf{e}_t\}$ as a non-empty subset of sentences $\mathbf{e}_s = \{s_1, \ldots, s_{|\mathbf{e}_s|<n}\}$, where n is the total text count, and cell values $\mathbf{e}_t = \{t_1, \ldots, t_{|\mathbf{e}_t|<p}\}$ arising from a table, where p is the total cell count. A supervised verifier model $\mathscr{F}$ assesses whether a textual claim c finds support or contradiction w.r.t. a given evidence $\mathbf{e}$. Formally, $\mathscr{F}$ takes as input a data pair $<\mathbf{e}, c>$ and outputs a verdict in the space $\mathscr{L}$. We name *gold evidence* the evidence selection from the fact-checking dataset. Those evidence are the clean ground-truth of evidence to be retrieved. They are required to deduce $\mathscr{L}$, in contrast with *noisy* evidence. In this work, we consider two settings. In the binary (2-label) setting $\mathscr{L} = \{Supports, Refutes\}$, while in the full (3-label) setting $\mathscr{L} = \{Supports, Refutes, Not\ Enough\ Information\}$. The *Not Enough Information* (NEI) label is associated to cases where in the corpus there is not enough evidence to state if the claim is supported or refuted. Some verifier proposal focus on the binary case because of the infrequency of datasets with three labels, e.g., NEI examples account for only 3% of FEVEROUS's instances.

Our Proposal. Figure 5.1 shows our framework[1] to understand the *contribution* of each evidence in predicting the label for a claim. It adapts a local post-hoc explainer (LIME and SHAP in our study) to evidence and claim. Local post-hoc explainers rely on "variations" of the input to compute the contributions of the features (every example's evidences) in the decision (for a claim). In our framework, a perturbed dataset is generated for the example by removing pieces of evidence. The explainers use the predictions of the verifier over the perturbed dataset coupled with the original claim to build the explanation. It assigns for each example a contribution to each feature (evidence) and to an *intercept*, defined as the average difference between the overall prediction and the total contribution of all individual evidences in the perturbed dataset. This can be interpreted as the contribution that the explainer attributes to the claim within the context provided by the evidence. The prediction is thus approximated expressed as a function of the contributions defined by the explanation:

$$\text{Pred} \approx \sum_{i=1}^{N} \text{contribution}(e_i) + \text{Intercept} \tag{5.1}$$

where contribution(e_i) (contribution of evidence i) is the contribution over the model prediction detected by an explainer for an evidence.

[1]The code is available on the project github https://anonymous.4open.science/r/explainable-fact-checking-268D/

Dataset	#Claim	T.	|Claim|	#Ev.	#Noise
Fev.3L	71.2k(27.2k/2.2k/41.8k)	train	25.3	1.6	16.8
	7.9k(3.5k/0.5k/3.9k)	test	24.9	1.4	17.6
Fev.2L	69k(27.2k/0/41.8k)	train	25.3	1.6	16.8
	7.4k(3.5k/0/3.9k)	test	24.9	1.4	17.5
SciFact	0.8k(0.2k/0.3k/0.3k)	train	12.3	1.3	9.0
	0.3k(0.1k/0.1k/0.1k)	test	12.5	1.3	8.7
AVTC	2.8k(1.7k/0.3k/0.8k)	train	17.1	2.6	5.6
	0.5k(0.3k/0.04k/0.1k)	test	14.4	2.6	5.5
FM2	10.4k(5.3k/0/5.1k)	train	13.7	1.3	9.1
	1.4k(0.7k/0/0.7k)	test	13.8	1.3	9.1

Table 5.2: Statistics of datasets used for training and testing the fact-checking models. The number of claim details stand from left to right for *'Supports'*, *'Not Enough Information'*, and *'Refutes'*.

Dataset	Precision	Recall	F1	Sufficient
Fev. 2L	0.07	0.36	0.12	✗
Fev. 3L	0.07	0.36	0.12	✗
SciFact	0.13	1	0.23	✓
AVeriTeC	0.32	1	0.48	✓
FM2	0.12	1	0.22	✓

Table 5.3: Performance metrics for the retrievers used to build every dataset (Precision, Recall, F1).

5.3 Datasets and Models

We first introduce the datasets in our study and how we inject noise in the evidence set for every example. We then present the different models in our study and the configuration of the explainers.

5.3.1 Datasets

We select four fact-checking datasets. All examples consist of a claim written by a human, a label, and the golden evidence used to classify the claim. Statistics about the datasets after pre-processing are in Table 5.2. When the original test set is private, we use the original validation set as test set.

Feverous [230] is an extension of Fever [231] with more complex claims. Claims are crafted by humans from textual and tabular evidence from Wikipedia. We linearize tabular evidence in the format: $Cell_{Value}$ *<context>*$Cell_{Header}$*</context>*. We create variants of the dataset with and without the label *'Not Enough Information'*, namely **Fev.3L** and **Fev.2L**.

SciFact [232] contains expert-written claims paired with evidence from scientific papers. The evidence is from textual sources only. One challenge of this dataset is that its content is scientific rather than general.

FM2 [233] is obtained from an online multiplayer game where users write claims from a list of evidence from Wikipedia. To gain points, claims must be hard to fact-check by other players. Evidence is only textual and the *'Not Enough Information'* label is not present.

AVeriTeC (**AVTC**) [104] contains real-world claims to verify with Web evidence. Each claim has evidence in the form of question-answer pairs supported by online content. We treat each pair as one textual evidence. This dataset has a fourth label, *'Conflicting Evidence/Cherry-picking'*, we do not report it in our results as no verifier returns it.

Controlled Noisy Evidence. Datasets come with golden evidence for every claim. However, we must include noisy evidence in the examples to enable our experiments. Whenever possible, we obtain the evidence with retrievers, as this is how they arise in practice. When no corpus is available, we mix the

golden evidence with noise. We show an overview of retrieval performances in Table 5.3. We detail next how we add noise to every dataset.

For Feverous, the corpus of Wikipedia pages and the retriever are provided in the pipeline, so we run it on the train and test datasets to obtain their noisy versions. We retrieve 5 documents per claim, and in each documents 5 sentences and 3 tables. The retriever selects an arbitrary number of cells per table. Crucially, golden evidence is missing after the retrieval step (0.36 Recall) and a lot of noise is selected in the dataset (0.07 Precision). In the Feverous datasets, recall is lower than 1, i.e., some golden evidence are not picked by the retriever. This may lead the labels of the verifier to be less accurate, as it may lack sufficient information to verify the claim.

SciFact authors included for each example five "distractor abstracts" that cover topics mentioned in the original article. We append sentences from these abstracts to the original evidence, up to a total of 20 sentences.

To write a claim, FM2's users use one to two sentences out of ten sentences selected from Wikipedia on a given subject, the remaining sentences are used as noise.

In AVeriTeC, gold evidence are human-created question-answer pairs. The authors provide a question-answer generator to obtain retrieved evidence. From the generator, we pick the least relevant pairs measured by BM25 ranking against the claim. We add an average of 5.5 pairs per example. For this dataset, the retrieval performance are higher than the other datasets.

5.3.2 Models and their training

For the verifier models, the objective is to infer a label from a claim and the retrieved evidence. Before testing models on the test set, we fine-tune them on the corresponding train set.

RoBERTa [141] is a transformer-based language model. We fine-tune the model on relevant NLI datasets [105] as in Feverous, with learning rate $1e^{-5}$ for the dataset with 3 labels, and $1e^{-7}$ for the datasets with 2 labels. We run the training for 1 epoch on Feverous and AVeriTeC, and 3 epochs on FM2 and Scifact. For Scifact, we further improve results by reducing the skew in labels in the train set.

GFCE [41] jointly trains veracity prediction and explanation generation using a fine-tuned version of DistilBERT. It leverages both claim texts and supporting evidence to produce justifications. We use a learning rate of $1e^{-5}$ for every dataset. We train FM2, Feverous and SciFact on 3 epochs, and AVeriTeC on 2. We reduce the imbalance in AVeriTeC at train time by equalizing the number of *'Supported'* and *'Refuted'* claims.

We report an LLM-prompting verifier based on LLaMa 3.1 70B [234]. The prompt includes the description of the task and the claim with its evidence. The model outputs a veracity label as well as a list of the evidence used to take the decision. The inference takes up as 2 hours for a thousand predictions. To enable analysis with SHAP and LIME, we also use an alternative prompt without the evidence-labeling task, reducing both input and output lengths, and consequently the running time. We observe a comparable behavior with this prompt in claim labeling, but with inference time reduced by 75%.

5.3.3 Configurations of the Explainers

We use LIME and SHAP to analyze verifier outputs. For the GFCE and RoBERTa models, we set 500 perturbation samples per explanation, while for LLaMa, we reduced it to 100 due to its computational cost. We explain a subset of the datasets: we exclude examples exceeding 512 tokens (too long for RoBERTa) and sample up to 1,000 examples from each dataset.

5.4 Experiments

We address the research questions presented in the introduction. For Q1, we start by investigating the ability of SHAP/LIME-based explanations to distinguish Noise from Useful Evidence by contrasting

them against intrinsic explainer models. Then, for Q2, we analyze how every model uses the input evidence and the claim in outputting a label. Finally, to address Q3, we investigate the importance given by verifiers to the useful evidence for each label individually.

5.4.1 Experimental Setup

The experiments were conducted on a system running Ubuntu 20.04.6 LTS with a kernel version of 5.4.0-177-generic. The hardware configuration consisted of an AMD EPYC 7272 12-Core Processor with 128GB of RAM, and a NVIDIA GeForce RTX 3090 GPU with 24GB of memory. The software setup included Python 3.8.10, LIME 0.2.0.1, SHAP 0.45.0. To run inference on LLaMa 3.1 70B, we used a more powerful server running Ubuntu 20.04.6 LTS with a kernel version of 5.15.0-122-generic. The hardware configuration consisted of an AMD EPYC 7742 64-Core Processor with 512GB of RAM, and a NVIDIA A100-SXM with 80GB of memory. The software setup included Docker 23.0.3 where we ran in a container LLaMa 3.1 using Python 3.9.

The runtime performance of the explainability methods was measured with comparable configurations: 500 perturbations per single explanation, explaining 3 set of 1000 examples (thousand per each dataset type) and applied to the two models in consideration.

- LIME: between 1:55h and 2:18h to explain 1,000 examples for the 3-label model; between 1:31h to 1:41h with the same configuration for 2-label model.
- SHAP: between 2:48h to 3:21h to explain 1,000 examples for the 3-label model; between 2:59h to 3:43h with the same configuration for 2-label model.

5.4.2 Configurations of the Explainers

To analyze the outputs of the verifier models, we utilized two explainability methods: LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations). These methods were employed with consistent configurations across different models, except for LLaMa 3.1 70B, which required adjustments due to computational constraints. For both the GFCE and RoBERTa models, we used 500 perturbation samples to generate each explanation. This number of samples ensures explanation stability and fidelity to the model's behavior. Given that the number of explanation units (evidence) is not greater than 10 it follows that we are always covering at least half of all the possible combinations or evidence. Considering that explainers weigh more samples close to the original item the chosen number of perturbations guarantees fidelity. However, for LLaMa 3.1 70B, generating explanations was more time-consuming due to the model's larger size and complexity. To accommodate this, we reduced the number of perturbation samples to 100. This reduction allowed us to obtain results in a reasonable time frame without sacrificing too much on the quality of explanations. Another important adjustment was related to the number of examples we explained from each dataset. We first filtered out examples that were too long for RoBERTa (over 512 tokens), as these would exceed the model's input length limit and make the explanations less comparable. From the remaining examples, we selected up to 1,000 examples to explain for each dataset. Some datasets had fewer than 1,000 examples available for explanation after filtering, in which case we explained all the available examples. This approach ensured that the explainers were applied to a manageable and representative subset of the data.

5.4.3 SHAP and LIME based Noise Detection

We measure the ability of SHAP and LIME to act as an evidence labeling system, when used to observe a verifier model. To assess the quality of our proposal, we measure it against explanations provided by intrinsic explainer models (GFCE and LLaMa), that output both a label for the claim and for each input evidence piece.

Evidence Labeling GFCE and LLaMa output a binary label for each evidence piece, together with an *importance in decision* score. However, this information does not label evidence piece in terms of their role in the decision, i.e., if they steer the verifier towards a refutes or supports decision.

For SHAP and LIME, we model the noise detection problem as a binary classification task using the sum of absolute contributions of the explanations as score. We hypothesize that any noisy evidence should leave the prediction unaltered, i.e., the explainer recognizes that a noisy evidence does not change the state of the claim. From the methods, we obtain scores for every evidence piece relative to its importance in a claim label decision. To label each evidence piece as noise or useful, we infer a threshold that separate them. The best values for each dataset and observed model are different, making this threshold unique to every setting. To infer each threshold, we use the explanation scores obtained for 100 evidence pieces from the train set executed on the model currently observed, as follows. As we are working on the train set, gold labels for evidence usefulness are available. We choose the threshold that maximize the sum of the F1 score for useful evidence and the F1 score for noisy evidence. We then categorize each evidence as Useful or Noise using the threshold obtained.

Table 5.4: Performance comparison across different models and datasets in terms of claim verification (left), explanations from the intrinsic explainers (middle), and explanations from the post-hoc explainers (right). Best explanation result for every dataset and observed model pair in **bold**.

DS	Model	Verifier				Intrinsic Explainer			SHAP/LIME		
		Acc.	F1 Sup	F1 Nei	F1 Ref	Acc.	F1 Useful	F1 Noise	Acc.	F1 Useful	F1 Noise
Fev.2l	Roberta	0.61	0.73	-	0.35	-	-	-	**0.88**/0.83	0.24/**0.30**	**0.93**/0.90
	GFCE	0.48	0.03	-	0.64	0.78	0.31	0.87	0.56/**0.83**	0.27/**0.40**	0.69/**0.90**
	LLaMA	0.69	0.70	-	0.68	**0.88**	**0.47**	0.93	**0.88**/0.86	0.21/0.20	**0.94**/0.92
Fev.3l	Roberta	0.60	0.71	0	0.41	-	-	-	**0.89**/0.88	0.35/**0.40**	**0.94**/0.93
	GFCE	0.59	0.69	0	0.46	0.79	0.33	0.88	0.80/**0.82**	**0.36**/0.34	0.88/**0.89**
	LLaMA	0.57	0.70	0.17	0.44	**0.88**	**0.46**	**0.93**	0.83/0.84	0.33/0.26	0.90/0.91
SciFact	Roberta	0.67	0.69	0.75	0.49	-	-	-	0.83/**0.86**	**0.28**/0.23	0.91/**0.92**
	GFCE	0.37	0.52	0	0.23	0.38	0.20	0.50	**0.66**/0.58	**0.21**/0.20	**0.79**/0.72
	LLaMA	0.76	0.80	0.71	0.72	0.80	**0.51**	0.88	**0.87**/0.85	0.29/0.26	**0.93**/0.92
FM2	Roberta	0.70	0.72	-	0.68	-	-	-	0.87/**0.88**	**0.27**/0.16	**0.93/0.93**
	GFCE	0.58	0.59	-	0.56	0.60	**0.32**	0.71	**0.87**/0.83	0.09/0.20	**0.93**/0.90
	LLaMA	0.87	0.88	-	0.87	0.85	**0.58**	0.91	**0.88**/0.85	0.40/0.38	**0.93**/0.92
AVTC	Roberta	0.79	0.86	0.52	0.78	-	-	-	**0.78/0.78**	**0.49/0.49**	**0.86/0.86**
	GFCE	0.59	0.50	0.50	0.69	0.37	**0.49**	0.17	**0.70**/0.69	0.48/0.44	**0.79/0.79**
	LLaMA	0.76	0.77	0.44	0.85	**0.88**	**0.81**	**0.92**	0.77/0.75	0.63/0.54	0.84/0.83

Useful Evidence Identification Table 5.4 presents our results. Each row represent a dataset and the verifier run on it – this is the observed model for our approach. Columns under *Verifier* report the associated verifier performance on claim labeling. Columns under *Intrinsic Explainer* report the the Accuracy, F1 Useful and F1 Noise for labeling evidence for the PLM-based models that can justify their decisions, i.e., GFCE and LLaMa. Columns under *SHAP/LIME*, report the same metrics using our post-hoc methods. All scores for a single dataset are computed by running the models on the associated test set - both for claim and evidence labeling.

Overall, SHAP provides more accurate results than LIME, independently of the metric analyzed. SHAP wins in 62%, 69%, and 54% of the cases when looking at Accuracy, F1 Useful, and F1 Noise, respectively. For most dataset/model pairs, the SHAP and LIME based explainers perform on par with the intrinsic explainer - GFCE or LLaMa - regardless of the metric. Post-hoc methods outperforms GFCE's explainer on 13 over 15 results when using such observed model as verifier. If we compare across all explainers for a single dataset (both intrinsic and post-hoc), the accuracy of evidence labeling metric for SHAP/LIME is the highest on 4 out of 5 datasets. In noise detection and accuracy, post-hoc methods

outperform intrinsic ones in all cases expect two, where LLaMA shines in terms of identification of useful evidence. The results show that post-hoc explainers can compete with PLMs that are specifically trained on this task. Finally, our proposal is the only one that can explain a Roberta model, which lack an intrinsic explaining.

Restraining to explanations based on SHAP and LIME, in all cases, the worst accuracy for evidence labeling is obtained when the observed model is GFCE's verifier. This is consistent with the performance of the verifier, that is in average for all datasets 14 absolute points under the worst scores among the two other verifiers.

When looking at results for a same model and a same dataset, we remark that GFCE's explanations have scores that are not concordant with its verifier's scores. The origins of this behavior relies in the separation of the explainer and verifier in this model. Indeed, even if they are trained with a joint-loss, the GFCE's explainer prediction are not directly linked to the verifier's decisions.

The performance of SHAP and LIME being highly dependant on the verifier's performance, we would expect the models to struggle on evidence labeling for GFCE. Most of the results of our explainer for this verifier are only a few points behind its alternatives. We make the hypothesis that GFCE's verifier is able to detect correctly noise, but still cannot infer the right claim label from the useful evidence, thus resulting in the verifier's scores observed. The analysis performed in Section 5.4.4 supports our reasoning.

Through the experiments, it appears that post-hoc methods are able to discern noise. Results show that in some situation they can identify them better than intrinsic explainers tailored on this task. By extension, we can conclude that fact-checking models effectively distinguish between useful and noisy evidence. This provides an answer to Q1.

5.4.4 Model Reliance on Claim and Evidence

We analyze how the verifier model depends on the (gold) evidence, noise and claim to make decisions over (1) whether the model relies on the claim and the evidence for predictions, (2) how contributions are distributed among the evidences.

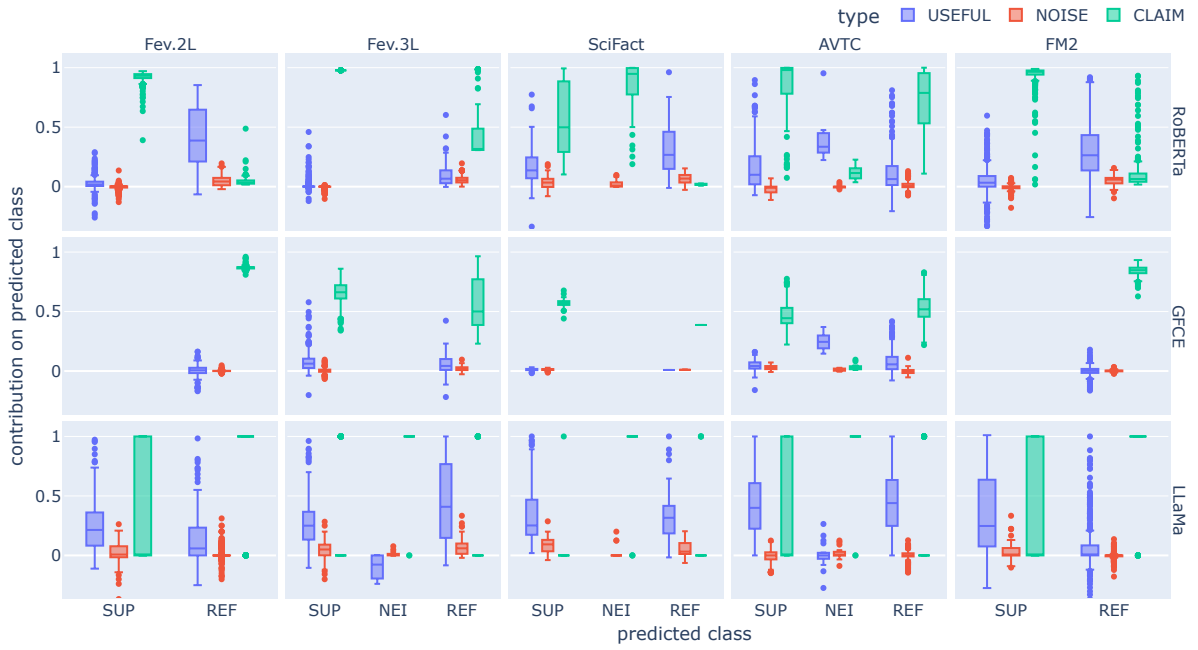

Figure 5.2: Distribution of contribution scores obtained with SHAP of the useful evidence, noisy evidence, and claim on the predicted classes (Supports, Refutes, NEI). We spread horizontally the five datasets and vertically the three observed models. Every plot reports the predicted class over the x and the prediction contributions for useful evidence, noise and the claim itself.

Importance of claim and evidence. We analyze separately the examples labeled by a model as *Supports*, *Refutes* or *Not Enough Information*. For each example, we compute three contributions: the one from the useful evidence (obtained averaging the contributions for all useful evidences), the one from the noisy evidence (by averaging them), and the one from the intercept (the contribution from the claim). The distribution of these contributions for all examples in the test set is shown in Figure 5.2 as USEFUL (evidence), NOISE (evidence), and CLAIM. Computing the average contribution for useful and noise lets us analyze their contribution independently from their unbalance in quantity. Even though models may assign greater contributions to useful evidence than over noise, in some experiments the higher number of noisy evidence makes the sum of their contribution greater than the contribution of the useful evidence.

We report the analysis on every dataset and every verifier. The behavior is consistent between SHAP and LIME, so we report results for the former, as it perform slightly better in the previous study.

From Figure 5.2, we see that regardless of the model or dataset, the average contribution of useful evidence in claim labeling is greater than contribution from noisy ones. The exception is label *Not Enough Information*, where Noise contributes as much as useful evidence. For this label, verifiers predominantly exploit the claim. This behavior indicates a lack of sufficient useful evidence, coherently with the results in Table 5.4.

Observing RoBERTa on the Feverous 2 labels dataset, we see that most of the contribution for *Refutes* come from evidence, especially the useful ones. In the *Supports* case, the median of the contributions of the claim is the only score deviating from zero, i.e., the claim is the only factors that contribute to the claim labeling - evidence being discarded. This may suggest that the model decides to label a claim as *Supports* when it cannot find any evidence to refute it. On the Feverous 3 labels, the model displays a similar behavior than on Feverous 2 labels. In the AVeriTec dataset, RoBERTa predominantly bases its decision on the Claim across all labels. We observe the same behavior for GFCE across all models. This reliance on the Claim, rather than on useful evidence, undermines the verifier's trustworthiness, as reflected in the non excellent results in terms of labeling of the claims in Table 5.3.

For LLaMa, the post-hoc methods show that the majority of the contributions always come from the evidence. However, among the three verifiers, it is the only one that does not count on evidence to label AVeriTec *Not Enough Information*.

To answer Q2, the analysis show that the three verifiers rely on evidence and claim very differently to perform their decision. The contribution of claim and evidence over label is very different across models.

Contribution spread on the evidence. We measure whether the contributions are evenly distributed across evidence pieces. We rank the evidences by decreasing contribution in all examples. Then, we compute the mean for all the evidences with the same rank.

Results in Figure 5.3 confirm observations made for answering Q2. We see that the verifiers rely on evidence very differently. The GFCE verifier is only lightly influenced by the evidence to take a decision, with a maximum contribution of the most useful evidence of 0.2 across every datasets on *Supports* and *Refutes* labels. Also, in the case where the predicted label is *Not Enough Information*, GFCE and RoBERTa on AVeriTeC have a high contribution score for the most useful evidence, showing that the model could switch its decision to *Not Enough Information* from a single evidence piece. This leads to a natural question: how can adding an evidence make the model switch the prediction toward lack of information? Such result may indicate that the verifier did not learn correctly the implications of this label, probably due to the very small percentage of such claims in the training data.

Finally, we look specifically at contribution ranking for evidence across models and datasets. LLaMa appears to make the most reasonable usage of evidence as, both for *Supports* and *Refutes*, it spreads the contribution scores more than the other verifiers over the available evidence. This verifier is also the only one that use the evidence on every dataset – with first contribution always around 0.5. As a non fine-tuned LLM, not tailored for the fact-checking task, we could expect RoBERTa and GFCE to take the advantage here. Furthermore, one could expect that the LLM's extensive knowledge base could enable it to disregard the provided evidence.

Splitting by label, evidence play a key role for RoBERTa when the model predicts *Refutes*, while this

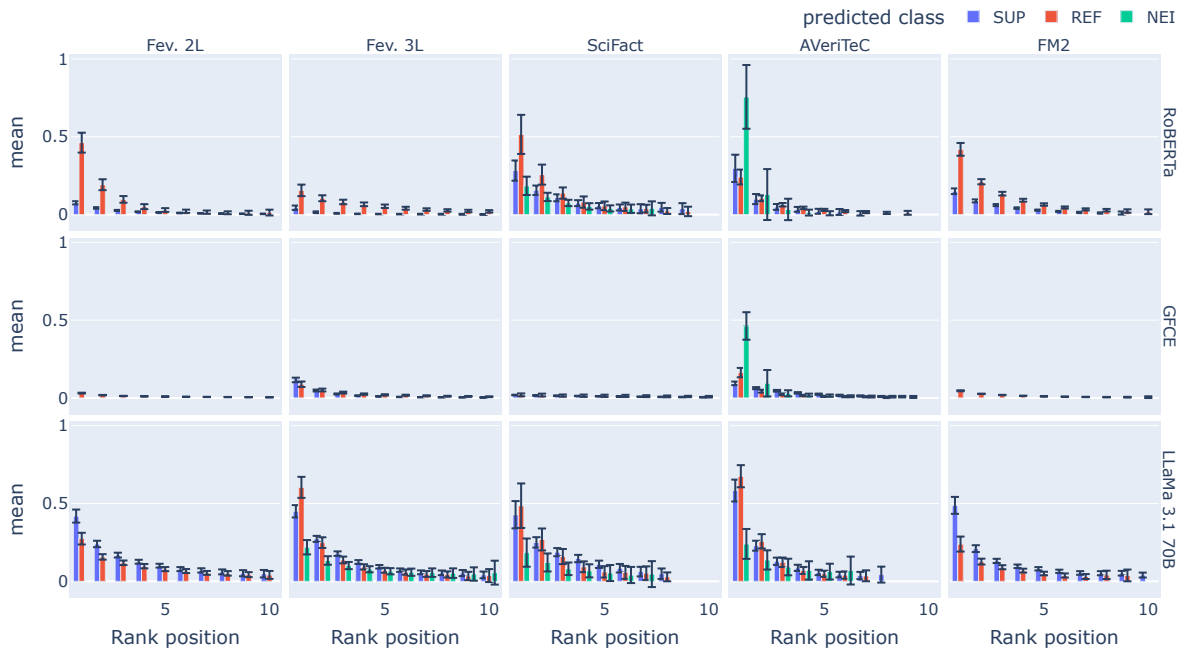

Figure 5.3: Mean and 95% confidence interval of contribution score over the predicted class grouped by predicted class and rank of the evidence piece in the set of evidence of an example, removing evidence pieces contributing negatively on the predicted class. Experiment run on each verifier and each dataset.

behavior is less visible for *Supports*. On the other hand, LLaMa exhibits a more balanced performance between the two labels. As for GFCE, the low contribution of each evidence concerns all labels. By exploring the distribution of the evidence's impact on the model, we have thoroughly addressed Q3.

5.5 Conclusion

We address the black-box nature of neural fact-checking models by applying post-hoc XAI methods to identigy relevant useful and its contribution to claim classification across four established datasets. Our findings suggest that models effectively rely on evidence for supported claims and can discern noisy evidence, thus enhancing transparency. We demonstrate that post-hoc methods like LIME and SHAP can sometimes provide higher-quality explanations than intrinsic explainers across three fact-checking systems. With a running time of 6 seconds on some model per one claim, post-hoc methods can be considered as a slower but effective alternative to intrinsic explainers. For future work, we envision a verifier model that inherently provides both labels and evidence explanations, aiming to leverage the insights of post-hoc methods while significantly reducing computational costs.

5.6 Limitations

We acknowledge the limitations in our study.

Our approach relies on the assumption that the evidence retrieved by the system is contextually relevant to the claims being checked. While our framework effectively identifies and explains relevant evidence, its performance is contingent on the quality of the evidence retriever. In practice, if the retriever retrieves predominantly noisy or irrelevant evidence, the effectiveness of the explainability methods (LIME and SHAP) in attributing correct contributions may be compromised. This assumption might not always hold true, especially in diverse or dynamic datasets where context relevance is harder to ascertain. Our empirical evaluation is confined to the Feverous dataset and the verifier models trained with it. While Feverous is a comprehensive benchmark, our findings' generalizability to other datasets, domains, or

languages remains unexplored. Consequently, the performance and robustness of our approach may vary when applied to different fact-checking scenarios or datasets with distinct properties. Future research should extend these evaluations to a wider array of datasets and contexts to better understand the scope of our method.

The effectiveness of our proposed framework is inherently linked to the capabilities of LIME and SHAP post-hoc explainability methods. These methods have their own limitations, such as sensitivity to input perturbations and assumptions about local linearity around the data points being explained. Should these assumptions fail, the explanations provided may not be reliable. But by aggregating a large number of explanations (3k) we consider more trustful the aggregated results rather than each single explanation. Moreover, both methods can be computationally intensive, which might limit their applicability for real-time or large-scale fact-checking tasks.

Chapter 6

Applying Computational Fact-Checking to Identify Health Misinformation on Social Media

Originally published as: Grazia Cecere, Jean-Flavien Bussotti, Clara Jean, Nessrine Omrani, and Paolo Papotti. *Digital Divide and Artificial Intelligence for Health*. Digital Transformation Society (May 2024), 35 pages. [11]

In this thesis, we explored technical solutions aimed at enhancing inference models. Our investigation focused on data-driven AI, specifically the generation of synthetic datasets, and we expanded our research to incorporate multi-modal approaches. Additionally, we examined the impact of input on output through the lens of explainability. While our work was primarily applied to the fact-checking use case, it had not yet been implemented in real-world scenarios. In this chapter, we support economic researchers by providing customized models for analyzing Facebook ads. In doing so, we leveraged our dataset generation techniques to address the imbalance in their training dataset, which originally contained uneven distributions of different labels.

6.1 Introduction

Artificial intelligence (AI) defined as machines, software, and algorithms that act by recognizing and responding to their environment [235] has caused significant transformations in a range of industries and sectors [236, 237, 238, 239, 240, 241]. It has triggered innovations in areas such as more sustainable production [242] and resource optimization [243], and is fundamentally changing how humans engage with technology. For these reasons, AI is often studied from a socio-technical perspective [244].

At the same time, past and current socio-economic development challenges persist around the world [245, 246, 247]. The 2030 Sustainable Development Agenda, an international agreement signed by 193 United Nations members in 2015, was created to respond to these challenges. It includes 17 Sustainable Development Goals (SDGs) and 169 targets and is aimed at promoting prosperity while preserving the environment and ensuring decent living standards. The growing influence of AI has prompted research into whether AI could become one of the tools used to achieve these SDGs [248, 249, 250]. Most of the existing research highlights the positive impacts of AI use in various sectors [251, 252, 253], and tends to underexplore its dark-side [254, 255].

One area where use of AI is particularly evident is online digital platforms and especially social media. Social media platforms are considered primary sources of information. Based on a 2023 Pew survey, 50% of users in the US always or sometimes use social media as a source of information. Meta is the most popular platform (30% of users) while TikTok has experienced the largest growth in the

last three years [256].[1] Compared to traditional media, the value added of social media is the ability of AI-driven algorithms to match content to individuals based on their personal characteristics or interests which results in significantly lower user search costs [257]. Alongside user-generated content, ads on social media are important sources of information. They serve commercial purposes and also are posted by a range of different institutions such as government and non-government organizations (NGOs), to inform individuals about issues such as health emergencies, or increase awareness of green developments.

However, the huge volume of content offered on social media platforms requires more than algorithmic matching; it also requires robust AI-driven algorithms to curate and manage this content. Curation of the massive volume of advertising is crucial for the platform management and it is achieved through use of AI-driven algorithms which evaluate a very large mass of content, and especially sensitive content related, for example to health.[2] Previous studies show that social media platforms are often subject to polarization in relation to political content [258], and to echo chambers [259] and potential algorithmic bias in the diffusion of information on job offers and education [260, 261]. A recent surge in (mis)information related to health prompted the World Health Organization (WHO) to coin the term "infodemic" and call for urgent research on the issue [262].

The problems caused by the spread of misinformation online has resulted in regulators and policymakers putting increased pressure on social media management platforms in terms of the content published on their platforms which now is subject to the recently implemented Digital Service Act (DSA). Several studies that adopt an SDG perspective have suggested that AI as a non-neutral technology, could hinder achievement of the SDGs [263, 264]. However, there is a lack of empirical evidence to substantiate these claims. In this paper, we explore whether AI and use of social media platforms as a primary source of information could threaten progress towards the SDGs related to reducing inequalities (SDG10) and increasing health and well being (SDG3).[3] We focus in particular on how AI-driven algorithms might exacerbate economic and social differences. The research questions addressed are: *Do AI-driven algorithms efficiently control the type of content diffused on digital platforms? Do AI-driven algorithms reduce or exacerbatesocio-economic inequality by reducing the digital divide?*

We collected ads displayed in the US on both Meta and Instagram from the Meta ad library for the period January to June 2020. We consider only ads displayed in all US states.[4] During the period analyzed which included the COVID-19 pandemic, many of the ads were health related. For each of the ads in our dataset, we have detailed information on the ad text, the total number of impressions, and the distribution of impressions by age and by gender. We also had information on the total number of ad impressions displayed by individual US states, and socio-economic demographic data for individual US states.

To address the research questions, we employed a method that involved use of a fact-checking algorithm which exploits a range of different technologies to detect misinformation related to health issues during the pandemic. The fact-checking algorithm enabled evaluation of the likelihood that an ad contained false or unsupported claims designed as misinformation [265]. Our approach combined broad understanding of complex textual claims enabled by pre-trained language models (PLM) with domain-specific knowledge allowed by the fine-tuning of these models with curated datasets.To train the model, we used human annotations and relied on synthetic data produced by a generative AI tool to increase the training sample size. Use of the fact-checking algorithm was aimed at identifying whether a given ad in our dataset included misinformation or not.

We combine data collected from the Meta ad library and the results of the fact-checking algorithm with US state-level administrative data. We were interested in whether algorithmic display of health-related ads reduces existing socio-economic inequalities by reducing the digital divide. To our knowledge, there is an absence of research on whether ad content distributed by AI-driven algorithms on social media affect inequality among individuals in terms of accessing health information.

[1] https://www.pewresearch.org/journalism/fact-sheet/social-media-and-news-fact-sheet/, last accessed April 2024.

[2] https://medium.com/@ComicRealmReports/metas-latest-findings-experiments-suggest-algorithms-have-no-impact-on-political-opinion-3bdf7
last accessed April 2024.

[3] https://www.pewresearch.org/journalism/fact-sheet/social-media-and-news-fact-sheet/, last accessed April 2024.

[4] see Section 6.3 section for more details

Our findings based on fact-checking show that very few of the health-related ads displayed on the Meta platform contained misinformation. Although the fact-checking algorithm categorized 15.41% of the ads as ambiguous only 0.2% were seen as providing misinformation. However, we found evidence of a digital divide in the context of ad display based on AI-driven algorithms. First, health-related ads classified as misinformation or ambiguous were more likely to be displayed in US states characterized by a low GDP per capita. Second, we observed higher display of health-related ads containing misinformation in US states with high percentages of individuals with no health insurance cover.

This paper has several managerial and policy implications. First, our fact-checking algorithm directly addresses the need for robust, automated online content moderation tools. In information-heavy environments such as newsrooms, the appropriate tools could help regulators, policy makers, and journalists to swiftly and accurately verify the validity of claims and data [266] by significantly reducing the pressure on human fact-checkers and promoting more timely debunking of misleading claims. This is in line with EU recommendations related to harnessing new technologies such as AI to tackle misinformation [267]. Problems related to managing misinformation were considered in 2024 by the World Economic Forum to be the most prominent risk over the coming two years[5]. Our study has made some initial steps towards bridging the gap between the research community and targeted markets, and offers some potential solutions to this issue. Second, platforms need to be aware of the dark-side of using AI-driven algorithms. For example, they can reinforce existing inequalities by conveying certain types of health-related information to certain population categories but not others. If this occurs, then AI-driven algorithms will be an obstacle to the achievement of the 2030 Agenda SDGs. Although the regulatory authorities are imposing increased scrutiny of information and content shared on social media platforms, our results highlights the need for better control of algorithmic ad displays and especially those related to health information.

Our paper is organized as follows. Section 6.2 reviews the current literature on online misinformation, algorithmic decision-making and internet technologies inequality. Section 6.3 introduces the context of the study and describes the data collection via the Meta Ad Library API and external data sources. Section 6.4 details the construction and training of the fact-checking algorithm. Section 6.5 offers preliminary analysis through descriptive evidence. Section 6.6 presents our main results. Section 6.7 discusses the potential mechanisms explaining our results and the underlying implications. Section 6.8 highlights some limitations and concludes.

6.2 Literature Review

The article relies on three streams of literature. First, we refer to the literature on online misinformation. Second, we contribute to the literature related to algorithmic decision-making. Third, we build on the literature that invests how internet technology affects inequality.

6.2.1 Online Misinformation

Numerous studies highlight social media platforms, including Meta and Twitter, as widespread channels for disseminating misinformation online [268, 269, 270]. In particular, [271] identifiy a robust correlation between Meta use and the consumption of false news articles. To complement this finding, [272] demonstrates that misinformation and rumors proliferate more easily in social networks featuring an echo chamber. An explanation is related to people who often share misinformation because their attention is focused on factors other than accuracy of the news they read [273].

While misinformation might seem insignificant in terms of impacts on individuals, the reality contradicts this assumption [274]. [275] discovers that the impact of misinformation depends on the sophistication of the individuals exposed to it. For example, [276] demonstrate, in the context of search engines, that online searches to verify the truthfulness of false news articles can actually increase the

[5] https://www.weforum.org/publications/global-risks-report-2024/, last retrieved January, 2024.

probability of believing them. In the electoral perspective, [277] reveal that overconfidence exacerbates the impact of widespread misinformation on democratic decision-making. During the COVID-19 pandemic, a plethora of information has circulated, occasionally leading to adverse effects on public health [278] with direct effect on vaccination intent [279].

Addressing the necessity to combat online misinformation, [280] explore the use of crowdsourcing as a potential solution. While the authors showcase crowdsourcing as an effective fact-checking strategy in specific settings, the inconsistency and lack of actionable results prompt the exploration of alternative solutions for more effective fact-checking of online misinformation. As an illustration, [281] apply transformer-based models[6] (CT-BERT) and node embedding techniques[7] (node2vec) to address COVID-19-related conspiracy theories using tweet text and user interaction graphs, and show this is a viable approach to address this challenge. [282] advocates Natural Language Processing (NLP) as a non-human innovation intermediary, enhancing decision-making by expanding information analysis and reducing costs through automation. In the same vein, [283] underscore the efficiency of NLP tools, particularly the F-term approach, in classifying patent documents based on technical attributes.[8]

Until now, there persists an ongoing discussion regarding the potential of AI in addressing the issue of detecting online misinformation [267]. Our article differs from previous studies with the development of a fact-checking algorithm specifically tailored to advertising content on social media platforms. Notably, our focus diverges from traditional emphasis on posts or tweets, commonly targeted in previous efforts. Moreover, the aim of this algorithm is to aid fact-checkers in more effectively debunking the classification of ad content containing misinformation.

6.2.2 Algorithmic Decision-Making

Given the large volume of content available online, digital platforms are increasingly relying on AI-driven algorithms to process content and data. AI defined as general-purpose technology is of great interest to an increasing number of private and public organizations [285] because AI-driven algorithms are inherently scalable. Another advantage of AI-driven algorithms is that they are able to perform human-alike task without human intervention or with little human involvement.

Given the need for personalization to ensure optimal content-user matches [286], AI-driven algorithms may lead to unintended side effects [287]. [288] show, through a comprehensive dataset, the intrinsic connection between news consumption, the news shared by friends, and how algorithms rank news. In further exploration, [289] find that content-based and collaborative filtering recommendation algorithms could contribute to the formation of filter bubbles. From a theoretical standpoint, [290] demonstrate that recommendation systems blend consumer interests with content. However, they warned that if the platform recommends a content type diverging from a consumer's optimal mix, the algorithm might introduce recommendation bias.

We contribute to this body of literature by examining whether and how misinformation related to health issues disseminates on social media platforms according to individual US states socio-economic characteristics.

6.2.3 Digital platforms and internet technology inequality

The third strand of literature focuses on examining the impact of internet technologies within the digital economy on societal inequality. The early diffusion of internet has been associated with a wider gap in the access and use of information and communication technologies (ICT) creating the so called digital divide. This has contributed to increase already existed inequalities [291]. There is a large literature that have documented the digital divide in the internet usage [292] and in the geographical dispersion of internet diffusion increasing inequality between urban and rural area [293]. Recently, a new form of digital divide

[6] A transformer model is a type of neural network that learns context and consequently, meaning, by discerning connections within sequential data.

[7] Node Embeddings are vectors that reflect properties of nodes in a network.

[8] The F-term [284] approach is a method used to classify patent documents using the k-nearest neighborhood method.

has been observed by [294] through the ability of individuals to navigate on the Internet while being aware of the role of algorithms. The fact that not everyone benefits equally from the technology requires more investigation towards establishing clear technology needs that could lead to the development of more coherent frameworks and policies to bridge digital gaps [295]. We enhance this body of literature by investigating how the nature of information disseminated through social media, facilitated by algorithmic decision-making, may exacerbate geographical digital divides and inequalities. This is achieved by providing varying levels of information quality based on individual and geographical characteristics.

6.3 Empirical Section

6.3.1 Data Collection

We gathered data from the Meta ad library to empirically test the diffusion of online misinformation and its algorithmic display according to socio-demographic characteristics of US states. We chose Meta due to its prominence as the main player in algorithmic news distribution channels [267]. Given the crucial importance of managing health-related misinformation, we focused on pandemic related ad content. Figure 6.1 shows an example of health-related ads related to pandemic. We collect the data using the Meta API.[9]

Figure 6.1: Health-related Ads Related to Pandemic

Our final sample included 149,734 ads from 1,102 advertisers. We limit our sample to only ads displayed in all US states. As advertiser can decide to target the whole country or only a given set of states, we consider only ads published in whole country to study the intensity of ad distribution in each state. This aims at mitigating advertiser targeting bias and to focus on algorithmic ad display.

Online advertising is an essential communication tool. While online advertising may be for commercial purpose, it is also used by public institutions and NGOs to inform individuals. Thus, algorithmic

[9]API is Application Programming Interface.

detection of information is triggered by the coexistence of different types of information. The urgency surrounding the COVID-19 pandemic meant that platforms had to be able to discern whether advertisements were appropriate or potentially contained inaccurate information [296].

Before publication, all paid ads on Meta are reviewed by an automated ad-screening system to ensure compliance with Meta's advertising policies. This algorithmic ad screening helps the platform to identify ad content that might harm users through what the platform terms 'unacceptable content'. Examples of unacceptable content include ads promoting child sexual exploitation, abuse, and nudity, discriminatory practices, hate speech, inaccurate health information, and anti-vaccine content.[10] Ad that includes any of these types of content violates Meta's advertising policy and is removed from the platform. Figure 6.2 shows an example.

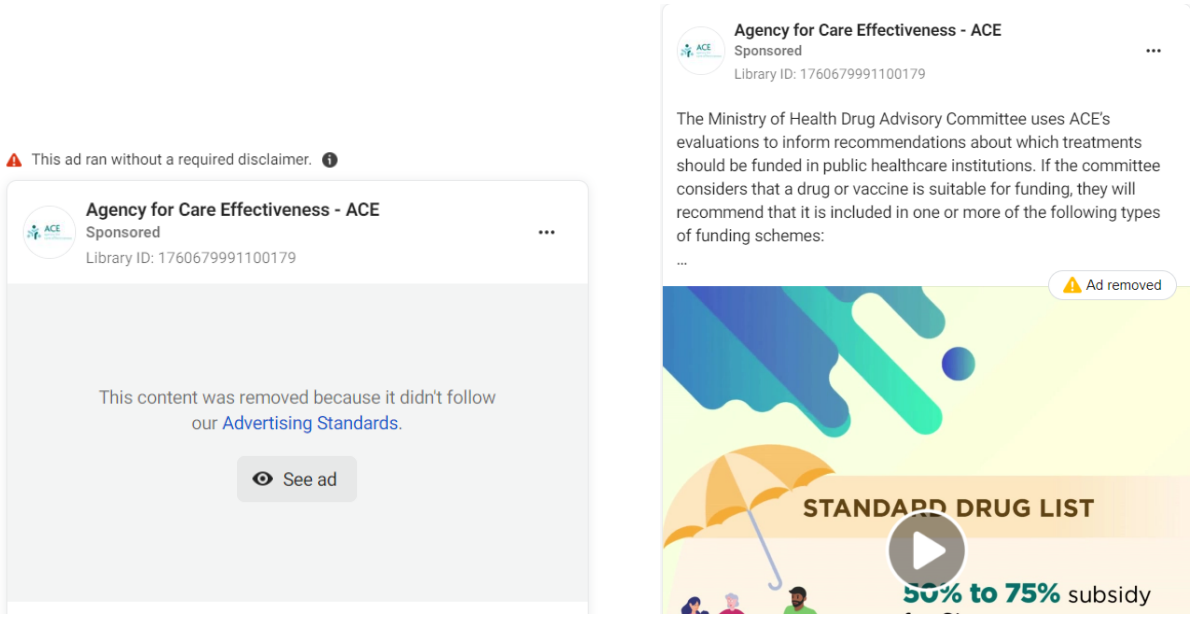

Figure 6.2: Example of Ad being Removed

6.3.2 Measuring Ad Distribution Inequality

To assess inequality in ad display, we augmented the Meta ad-related data with three sources of US state-level administrative data. First, we collected open data from the US Census Bureau for year 2020 on the percentages of people without health insurance in the whole population by individual US state. Second, we collected US Centers for Disease Control and Prevention (CDC) open data on number of COVID-19 cases per state and per month in 2020, and matched them with our study period. Third, we collected data on state GDP for the year 2020 from the US Bureau of Economic Analysis (BEA). We aggregated ad-level data to the state level enabling us to match ad-related data with US administrative data.

6.4 Fact-checking using a Deep Learning Model

To analyse the ad content, we use a fact-checking method that relies on a deep learning model. We rely on a fact-checking model to verify and annotate the content of real ads published on Meta's ad library platform. To annotate the whole dataset, we use a fine-tuned CT-BERT [297] pre-trained model to predict a label for each ad. CT-BERT is a deep learning model trained on crisis-related text data, including social media posts and domain-specific vocabulary. It aims to better understand and interpret language used during various types of emergencies. We adopt it to build a classifier to automatically label our examples

[10] See https://transparency.fb.com/policies/ad-standards/ for a full list, last retrieved February, 2024.

because, given the purpose of its creation, it is well adapted to our task. While CT-BERT is a relatively small model, we find that its performance are on-par or better than much larger models, such as LLama2 7b [298] fine tuned for classification and ChatGPT[299] instructed with prompting.

As any encoder model, CT-BERT relies on fine-tuning to be instructed for our dataset and labels. We further train the original CT-BERT model using custom examples. We use human annotation to classify the content of a random sample of 2,600 ads including both health-related ads and not health-related ads ads. We annotate this set of ads using three categories: "Not misinformation", "Ambiguous" and "Misinformation". We follow the same approach used by [300] based on the guidelines provided by the statement of the Australian Competition and Consumer Commission (ACCC) to distinguish between false and misleading claims.[11] We also rely on the FTC statement related to unproven COVID-19 health related claims.[12]

However, this manually annotated dataset cannot be directly provided to the model as the training data is significantly skewed across the distribution of the labels, with "Not misinformation" being the most represented label. The class unbalance is an important problem in the training set, as this may lead the model to predict only the popular class. We therefore address this issue with two combined approaches.

First, we augment the data in the "Misinformation" class by generating 400 synthetic examples [301, 302]. We generate such examples by using ChatGPT [299] with two methods based on "in context" learning. The first exploits the availability of misinformation definitions discussed above (from ACCC and FTC). We include in the LLM prompt the definition of misinforming text to steer the generation [303]. A second method is also based on prompting LLMs, but instead of descriptions, we use human labeled misinformation examples following the "few shots" approach [304]. The newly generated data set is then evaluated by humans. Ultimately, only 250 examples are considered high quality and added to the training data.

We report the three prompts that we used to generate additional training examples (labeled as "Misinformation") to fine tune the fact-checking model.

In the first prompt, we give to the model guidelines for creating claims. These guidelines are sourced from the FTC and ACCC. No examples are given to the model. Therefore, we ask the model to generate claims which violates at least one of the provided guidelines.

> I'm building a fake news detection model. For this I need misinformation claims.
> Given the following rules, can you write me some claims that would be classified as misinformation : -Product and service claims must be truthful, substantiated, and include accurate information on price, quality, and benefits. -Misrepresentations, withholding key information, or making false origin claims are illegal. -Exaggerations are often permissible, but objectively false claims, especially about prices and comparative advantages, are not. -COVID-19 related advertising must avoid false or unproven treatment claims; comparative claims require accurate information. -Claims about product quality, safety, and environmental impact must be factual and specific. -All claims, including those about COVID-19 prevention or treatment, need credible scientific support and transparent pricing. -The FTC enforces against deceptive COVID-19 claims; platforms monitor and penalize such misinformation.
> Each claim should break one or multiple rules above. The claim should not be too obvious. For example, avoid writing in the claim 'there are no evidence that this is true'. Also, the claim should not contain the evidence that prooves it is misinformation

In the second prompt, we only provide the model a small list of examples from our dataset, and an instruction on what to do. The task is to supplement the list with additional items.

[11] See the definition: https://www.accc.gov.au/consumers/advertising-and-promotions/false-or-misleading-claims, February, 2024

[12] https://www.ftc.gov/business-guidance/blog/2021/04/advertisers-stop-unproven-covid-claims-or-face-penalties-under-new-law, February, 2024

Here is a list of covid misinformation :
['whoa... we just ignore that aliens are real?! watch the first episode of 'real america' with graham allen!in turning point usa's brand new series, graham cuts through the fake news & tackles the biggest stories of the week. join us on youtube & facebook every thursday at 4 est! #realamerica', 'selinexor killed the virus in a petri dish. next steps?', "the false chinese government propaganda against our president has gone too far. as usual, china is trying to brainwash the rest of the world while throwing america under the bus.it's time to put an end to their communist propaganda once and for all, but we can't do it without you. president trump is calling on every american to step up and defend their country against the chinese communist party lies!please sign our official petition demanding an end to the chinese government propaganda against our president!", "some claim it could cure covid-19. here's the story of the controversy surrounding the antiviral drug avigan in italy.", [...]]
Can you add other items to the list. They should be written in a similar style

The third and last prompt is a combination of the two previous prompts: We provide the model both examples and rules to generate claims.

Your task is to generate short sentences that contains misinformaion technique in order to train a fact checking model. The definition of misinformation technique is the following: Product and service claims must be truthful, substantiated, and include accurate information on price, quality, and benefits. Misrepresentations, withholding key information, or making false origin claims are illegal. Exaggerations are often permissible, but objectively false claims, especially about prices and comparative advantages, are not. COVID-19 related advertising must avoid false or unproven treatment claims; comparative claims require accurate information. Claims about product quality, safety, and environmental impact must be factual and specific. All claims, including those about COVID-19 prevention or treatment, need credible scientific support and transparent pricing. The FTC enforces against deceptive COVID-19 claims; platforms monitor and penalize such misinformation.
Here are some examples: - 'have you seen more people wearing face masks in response to the coronavirus? turns out, they might not be as effective as they think.', [...]
Please generate sentences that contains the misinformation technique, as detailed above, similar to the examples, on similar topics.

Second, at the training time, we also optimize the process by giving more importance to human data as well as taking consideration of class unbalance with a custom loss function. A loss function is a mathematical method used to measure how well the model's predictions match the actual known values, helping to guide the improvement of the model during training. Example of an under represented class, such as the "Misinformation" class, are given more importance individually during the training step, so that the model is more likely to recognize them, even if they are rare. The higher importance given to ads of such class permits to obtain better predictions for the under represented classes.

6.4.1 Results of the Fine-tuning for CT-BERT

The model we use for the final annotation of the unlabeled texts is trained on 2.1k manually annotated examples. To assess the quality of the model, we keep out of training 500 of the original human-annotated examples for test. We also generated using ChatGPT 400 synthetic ads for the "Misinformation" category and use in the training set the 250 ads considered of high quality after human evaluation.

To use the CT-Bert classification model in our setting, we need to adjust some of its functionalities and parameters. In the training process, we give more importance to human data with respect to ChatGPT generated data, and we also take into consideration class unbalance. For this goal, we use a custom loss function to train the model. The custom loss takes into consideration the percentage of each label to

compute the loss. In our case, it gives higher importance to texts with label "Misinformation", as they are rare in the dataset. The custom loss also gives human training examples a coefficient two times superior to the one for generated examples. This choice reflects the intuition that the original examples are more representative of reality, thus it is preferable to privilege them in the training.

We run the training and inference of our CT-BERT model on a cluster of OS Linux workstations. The experiments were run on a single GPU, a NVIDIA TITAN Xp that has a VRAM capacity of 12GB. This is a low-resource setting that allows most people to run this model without needing a high-end GPU. The fact-checking models relies on PyTorch [203] and on the HuggingFace library [204]. To train the CT-BERT model, we set the following hyperparameters: 10 epochs, a batch size for train and evaluation of 8. We define the learning rate to be $5e^{-5}$.

Table 6.1 reports the results of the model evaluation. Overall, the model is able to effectively predict both "Misinformation" and "Not misinformation" categories, with an overall accuracy of 0.923. This number is obtained by computing the average of F1 scores available in Table 6.1 between "Not misinformation" and "Misinformation" categories. At inference time, 76.3% of the original *Misinformation* claims are either labeled as *Misinformation* or *Ambiguous*, confirming the quality of the predictions of the model.

Overall, the analysis of the deep learning model indicates that our dataset includes 0.2% published ads that contain misinformation and 15.41% published ads being ambiguous. This represents a low percentage suggesting AI-driven algorithms filter ads containing misinformation.

Table 6.1: ML Model Evaluation on 500 Manually Labeled Examples.

Label	Precision	Recall	F1 Score	Support
Not misinformation	0.8795	0.9733	0.9241	375
Ambiguous	0.3333	0.0385	0.0690	52
Misinformation	0.8861	0.9589	0.9211	73

6.5 Descriptive Statistics

The final sample includes 145,272 ads published in all US states from January to June 2020. As our objective is to measure whether algorithmic distribution of ads reduces or exacerbates inequalities, we aggregate the ad display at the state level. We compute the average number of impressions by state and by ad category, i.e. health related or not, and classification made by the fact-checking algorithm. Therefore, we end up with 50 observations corresponding to the 50 US states, District of Columbia being not available in the data. Table 6.2 presents the descriptive statistics of the main variables used in the empirical analysis. The data includes the average number of impressions by types of ads and state characteristics including population, GDP per capita, percentage of people without health insurance and total COVID-19 cases. Overall, we observe a significantly higher number of impressions made for health-related ads classified as misinformation compared to non health-related ads in the same misinformation category. This is in line with previous work that show that misinformation proliferates faster on social media [269]. This first evidence raises concerns towards the equality in terms of access towards information quality online.

Table 6.2: Summary Statistics at State Level

Variable	Mean	Std. Dev.	Min	Max
Non-health-related ads				
# Impressions - non-misinformation ads	923.653	959.826	93.779	5013.228
# Impressions - ambiguous ads	942.829	930.913	96.863	4325.648
# Impressions - misinformation ads	469.249	463.901	51.075	2256.227
Health-related ads				
# Impressions - non-misinformation ads	1034.148	1190.013	74.253	6281.785
# Impressions - ambiguous ads	932.576	948.202	90.95	3700.076
# Impressions - misinformation ads	2351.356	2252.841	249.882	10949.464
State characteristics				
Population	6,652,315	7,451,008	581,381	39,029,342
GDP per capita	58,981	10,975	39,157	88,467
% of people w/o health insurance	9.722	3.541	3.2	19.9
Tot. COVID-19 cases	1,0748,450	11,639,839	500,900	53,548,352
Observations	50			

To account for population differences, we normalize the data by dividing the number of impressions in each state by its population giving us a ratio of impressions per capita. Since our primary interest is in the algorithmic display of health-related ads, we focus on the sub-sample of ads related to the pandemic.

6.6 Empirical Analysis

We explore the potential correlation between ad distribution and socio-economic measures at the state level.

In this section, we adopt an econometric approach to study whether algorithmic ad display exacerbates digital divide by increasing inequality in access to information. Our approach relies primarily on ordinary least squares (OLS) estimates. We use as dependent variable the number of impressions displayed to a given state *i*. We have three X_i explanatory variables: 1) the GDP per capita, 2) the percentage of people without health insurance, 3) total COVID-19 cases. Equation (1) captures our main econometric specification. Standard errors are clustered at the US state level. The equation, we estimate, is as follows:

$$Impressions_i = \beta_0 + \beta_1 X_i + \epsilon_i. \tag{6.1}$$

6.6.1 Is There a Link Between Misinformation Ad Display and GDP per Capita?

Table 6.3 illustrates the correlation between the number of impressions and the GDP per capita of US states. We create a dummy variable *High GDP States* which takes the value of 1 if the GDP per capita in a given state is above the median value equals to 58007.85 dollars and 0 otherwise. Columns (1) to (3) provide estimates for the sub-sample of non-health ads and columns (4) to (6) provide estimates for the sub-sample of health-related ads. While there is no significant relationship between the number of impressions and high GDP states for ads classified as not being misinformation for both non-health and health ads (columns 1 and 4), we observe a different pattern for ads classified as ambiguous or misinformation. Column (3) shows that, among non-health related ads, states with a high GDP per capita are significantly less likely to see ads with misleading claims compared to low GDP per capita

states. On the other hand, Column (6) shows the same trend where states with a high GDP per capita are significantly less likely to see ads with misleading claims compared to low GDP per capita states; however the magnitude of the coefficient is larger for health-related ads compared to non-health ads. This result suggests that AI-driven algorithms can exacerbate socio-economic inequality where wealthier states, captured by GDP per capita, are likely to receive information of better quality when it comes to health-related ads.

Table 6.3: Ad Distribution and GDP per Capita

	Non-Health Ads			Health Ads		
	Not Misinformation	Ambiguous	Misinformation	Not Misinformation	Ambiguous	Misinformation
	(1)	(2)	(3)	(4)	(5)	(6)
High GDP States	-8.142	-110.643**	-53.780***	92.066	-132.244	-274.715**
	(34.206)	(54.802)	(17.15)	(77.852)	(116.175)	(127.756)
Constant	79.327**	178.589***	83.201***	-27.223	222.892**	510.539***
	(33.150)	(52.355)	(19.378)	(74.503)	(98.378)	(121.746)
R-squared	0.979	0.954	0.975	0.929	0.820	0.948
Observations	50	50	50	50	50	50
Population	Yes	Yes	Yes	Yes	Yes	Yes

Notes: OLS estimates. The dependant variable is the number of impressions displayed in a given state. Errors clustered at the state level. We include state population. Significance at 1%; 5% and 10% levels indicated respectively by ***,** and *.

6.6.2 Are Health-Related Ads with Misleading Claims Displayed More in States with Higher Rates of Uninsured Individuals?

This section aims to investigate whether the display intensity of ads classified as misinformation is likely to be correlated with the percentage of people without health insurance at state level. Figure 6.3 shows the percentage of people without health insurance by state on the left, and health-related ad impressions displayed by state on the right.

We observe that states with a small percentage of people without health insurance are likely to have a higher ad display about health-related ads. In other words, states with a high percentage of people without health insurance are less exposed to health-related ads.

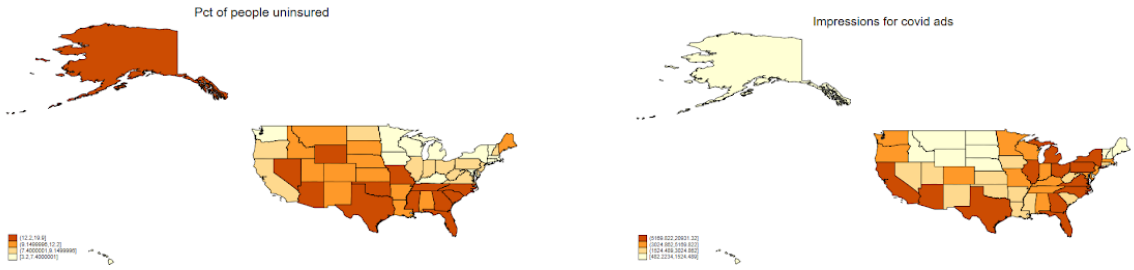

Figure 6.3: Percentage of People without Health Insurance per State and Health-related ad Impressions

We conduct an empirical analysis to study the previous evidence showing a correlation between state health insurance coverage and algorithmic ad display across the different ad categories. Table 6.4 shows the results. The dependant variable is the number of impressions at state level. Columns (1) to (3) show estimates for the sub-sample of non-health-related ads and columns (4) to (6) provide estimates for the sub-sample of health-related ads. We observe a significant and negative correlation between the percentage of people without health insurance and the ad display intensity per state for ads classified as "Not misinformation" (column 1 and 4) and the coefficient is larger for health related ads in Column (4). Conversely, Columns (3) and (6) show that there is positive and significant correlation between the percentage of people without health insurance and the intensity of ad displayed classified

as "Misinformation". The coefficient is larger in Column (6). Therefore, living in a state where the percentage of people without health insurance is high is associated to a higher exposure to low-quality health related information compared to other states. This suggests that the percentage of health insurance coverage in a state is likely to shape the quality of information received by users living in this state, perpetuating social inequalities among individuals.

Table 6.4: Disparities in Health Ad Display linked to Health Insurance Coverage

	Non-Health Ads			Health Ads		
	Not Misinformation	Ambiguous	Misinformation	Not Misinformation	Ambiguous	Misinformation
	(1)	(2)	(3)	(4)	(5)	(6)
% people w/o health insurance	-18.720**	-0.784	8.927***	-56.161***	-9.641	50.466**
	(7.763)	(8.884)	(3.145)	(15.658)	(17.075)	(24.024)
Constant	246.654***	139.874	-20.497	523.068***	255.986	-63.536
	(86.760)	(85.449)	(28.835)	(160.876)	(160.709)	(255.128)
R-squared	0.983	0.950	0.977	0.954	0.817	0.951
Observations	50	50	50	50	50	50
Population	Yes	Yes	Yes	Yes	Yes	Yes

Notes: OLS estimates. The dependant variable is the number of impressions displayed in a given state. Errors clustered at the state level. We include state population. Significance at 1%; 5% and 10% levels indicated respectively by ***,** and *.

6.7 Discussion and Implication

Social media are widely used worldwide as primary sources of information. Given the proliferation of misinformation on social media platforms, policymakers and regulators urge social media to improve the curation of content available on their platforms. The widespread dissemination of online misinformation poses challenges to the integrity of various markets, including media, cybersecurity, and social networks. In particular, promoting equality in accessing reliable health-related information is crucial from a public health perspective. However, given the increased role of algorithms in matching ad content to users, little is known about how algorithmic curation can reduce digital divide by increase the access to health-related information via digital platforms.

Our paper aims to bridge the gap between the use of AI-driven algorithms and reducing inequalities (SDG 10), including access to good health and well-being (SDG 3). Through the construction of an innovative fact-checking algorithm and the analysis of ad-related data collected from the Meta ad library, we provide evidence that the use of AI-driven algorithms can contribute to an increase in the digital divide, conflicting with the goal of reducing inequalities as stated in SDG 10. Moreover, even if our findings show that very few ads published by Meta's platform include misinformation, we provide evidence of disparate ad displays made by the algorithm according to US state characteristics, preventing equal access to health-related information as stated in SDG3.

These findings have significant implications, both nationally and internationally. Countries with high percentages of uninsured health people may experience exacerbated inequalities, given that uninsured population often comprises working-age adults with lower education and income levels. Additionally, such implications extend to countries with high income and wealth inequality indices. For example, India has seen a surge in misinformation queries during the pandemic, ranking highest in the risk of disinformation and misinformation dissemination.[13] This has contributed to a widening global digital divide between countries, with a new challenge emerging in the form of the use of health misinformation for geopolitical purposes.[14] Therefore, companies, especially social media platforms, need to strengthen

[13] https://www.who.int/images/default-source/digital-health/google-data-insights.jpg?sfvrsn=16b5b112_5, https://www.statista.com/chart/31605/rank-of-misinformation-disinformation-among-selected-countries/, last accessed April 2024.

[14] https://www.europarl.europa.eu/RegData/etudes/ATAG/2020/649369/EPRS_ATA(2020)649369_EN.pdf, last retrieved April 2024.

their efforts when it comes to health-related information displayed by algorithms, where the reinforcement of inequalities and a digital divide is heightened.

Currently, understanding the role of social media network structures in misinformation diffusion remains imperative [305, 306]. Our research reinforces this necessity, emphasizing the significance of algorithmic transparency and accountability. This also aligned with the increasing regulatory pressure on social media platforms to regulate the ad content they distribute. While the Digital Service Act represents an initial step, forthcoming regulations such as the AI Act may further enhance algorithmic transparency. From a practical perspective, regulatory bodies could aid social media platforms by providing ad-focused fact-checking tool for scrutinizing social media algorithmic behavior. Our fact-checking tool directly addresses this market need since it permits checking the validity of ad claims related to health issues during emergencies. The main difference with previously built tools is that it is tailored to ads on social media, which was not the case before. Additionally, platforms could implement strategies such as detecting and flagging disproportionate ad displays given a set of pre-determined attributes, use health-related ad content run by reputable organizations to assess the veracity of other contents on a similar topic, standardizing geotargeting practices in crisis context, and expanding user feedback mechanisms to combat misinformation effectively. These measures collectively aim to enhance transparency, combat misinformation, and foster a safer online environment.

6.8 Conclusion and Limitation

We collect data for January to June 2020 from the Meta ad library, on ads displayed in all US states. We augmented these data with US government information on GDP per capita, percentage of uninsured people, and numbers of COVID-19 cases in a given state in 2020. To evaluate the platform's curation, we developed an innovative fine-tuned fact-checking algorithm based on a deep learning model trained on human annotated data and synthetic data generated by means of LLM which we applied to the textual content available in the ads.

Our fact-checking tool showed that only a small fraction of the ads in the sample were classified as containing misinformation. This suggests that the platform's curation is effective in terms of assessing the content available on the platform. However, we found evidence of a digital divide in the algorithmic ad display. We observed that US states with low levels of per capita GDP were less likely to receive high-quality information and received a larger proportion of ads containing misinformation. We found also that US states with a higher proportion of health uninsured were more likely to receive misinformation about pandemic, and that states with high rates of COVID-19 cases were more likely to be displayed ambiguous health ads.

Our study has some limitations. First, the six-month time span may not be enough to account for potential learning and adjustments by the algorithm over time. Second, despite our efforts to rectify class imbalances in our fact-checking tool we acknowledge the potential for false positives. As is the case when studying any form of AI, distortions can occur due in particular to unrepresentative data. For instance, training the algorithm exclusively on pandemic-related ads might have restricted its ability to generalize effectively within the health domain. Also, training the algorithm on synthetic data could lead to bias amplification. Bias amplification occurs if the synthetic data inherit biases in the original training data. On the other hand, use of high quality synthetic data can produce economies of scale by reducing the volume of human-generated data needed for training with direct cost implications for businesses.

Despite these limitations, we believe our work should be helpful to policymakers and help to debunk misleading claims. It also highlights the need for platforms to actively address and mitigate the inequalities that are exacerbated by AI-driven algorithm ad displays.

Chapter 7

Future Work and Conclusions

In this thesis, we presented a comprehensive framework that automates the generation of training examples for Tabular Natural Language Inference as outlined in Chapter 3, and we extended this approach to multi-modal fact-checking in Chapter 4. Our systems, TENET [187] and UNOWN [178], have successfully demonstrated that high-quality datasets can be generated with minimal human annotation, effectively addressing the critical bottleneck in fine-tuning fact-checking models. Across multiple datasets, our experiments affirmed that models trained on synthetic data generated by TENET and UNOWN perform comparably to those trained on manually curated datasets. Additionally, our multi-modal system, UNOWN, effectively incorporates both tabular and textual inputs. The applicability of our systems is illustrated through our work showcasing the potential to support research in socio-economic domain [11] in Chapter 6.

Moreover, we tackled the challenge of explainability in fact-checking models in Chapter 5. By leveraging state-of-the-art explainable AI (xAI) methods such as SHAP and LIME, we identified the critical evidence used by fact-checking models to generate predictions. This approach not only allows users to rely on the model's outputs but also enhances their understanding of the reasoning behind each decision. The experiments evidenced that these xAI techniques provide meaningful insights into the relevance of evidence, ultimately fostering greater trust in the predictions of fact-checking systems.

Open Challenges

While our contributions advance the automation of fact-checking and its model explainability, several future research directions remain open.

1. Improving Diversity and Complexity of Generated Claims A critical area for future research is the enhancement of the diversity and complexity of the claims generated by our systems. While our current frameworks yield high-quality synthetic examples, there is still considerable room for improvement in capturing a wider variety of claim structures and types. Specifically, it is essential to develop the capability to generate claims that incorporate sophisticated mathematical and logical operations [108]. For instance, the current models mainly focus on generating straightforward logical assertions, but many real-world claims require nuanced reasoning, such as comparative statements, conditional assertions, or claims reliant on statistical data and calculations [307]. Existing systems assist fact-checkers in verifying numerical data [308]; however, it does not automate the full verification process. Addressing this challenge would not only make the generated examples more representative of real-world scenarios but could also significantly impact the training and robustness of fact-checking models, making them more capable of handling complex queries in diverse domains.

2. Extending Systems to Handle Multi-Table and Complex Multi-Modal Inputs Further advancing our systems to manage multi-table structures and more intricate multi-modal inputs is another challenge.

Current implementations focus primarily on leveraging single tables and textual evidence. However, real-world scenarios [309], particularly in domains like finance, health, and social media, often present information across multiple tables or combined with different types of media, including images and videos [310]. Thus, developing methodologies and architectures capable of effectively processing and integrating these varied forms of evidence will enhance the versatility and applicability of our models [311]. For instance, in the context of fact-checking on platforms like X.com, where visual elements often accompany textual claims, the ability to jointly analyze and interpret both textual and visual inputs will lead to more comprehensive and accurate verifications.

3. Lightweight Explainable Models for Integration into Fact-Checking Systems Lastly, fostering the development of lightweight explainable models that can be seamlessly integrated into existing fact-checking frameworks is essential for scaling the deployment of our systems. While our research on xAI methods such as SHAP [228] and LIME [46] has laid a foundational understanding of how to interpret model decisions, the challenge remains to create explanation systems that are computationally efficient and capable of operating in real-time alongside the main predictive models. Incorporating these explainable components directly into the fact-checking models would not only streamline the architecture but also reduce the computational overhead associated with generating separate explanations [312, 41, 44, 43]. As a result, users would gain immediate insights into the reasoning behind predictions without any latency, thereby enhancing their trust and confidence in automated decision-making systems. This approach could also pave the way for more interactive user experiences, wherein stakeholders can query the model for clarifications on specific decisions, ultimately fostering a more informed user base.

In conclusion, this research presents advancements in minimizing the manual effort needed to construct fact-checking systems while enhancing the transparency of their predictions. The innovations introduced in this work lay the foundation for the development of scalable, interpretable, and more reliable computational fact-checking systems, making them suitable for deployment across a diverse range of domains and applications. As a result, these advancements not only contribute to the efficiency of fact-checking processes but also bolster user trust in automated decision-making systems, facilitating more informed and responsible information dissemination in an increasingly complex information landscape.

Bibliography

[1] Esteban Ortiz-Ospina. The rise of social media. *Our World in Data*, 2019.

[2] Le Monde. Bedbug panic was stoked by russia, says france, March 2024.

[3] WEF. Global risks 2024: Disinformation tops global risks 2024 as environmental threats intensify, January 2024.

[4] I Cheng, Johannes Heyl, Nisha Lad, Gabriel Facini, and Zara Grout. Evaluation of twitter data for an emerging crisis: an application to the first wave of covid-19 in the uk. *Scientific Reports*, 11, 09 2021.

[5] Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers, 2021.

[6] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 1803–1812, New York, NY, USA, 2017. Association for Computing Machinery.

[7] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: fact extraction and verification over unstructured and structured information. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[8] Fine-tune a pretrained model. https://huggingface.co/docs/transformers/training.

[9] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain,

Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[10] Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:*, 2024.

[11] Clara Jean, Jean-Flavien Bussotti, Grazia Cecere, Nessrine Omrani, and Paolo Papotti. Digital divide and artificial intelligence for health. In *DTS24*, 2024.

[12] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[13] Vinay Setty. Surprising efficacy of fine-tuned transformers for fact-checking over larger language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2842–2846, New York, NY, USA, 2024. Association for Computing Machinery.

[14] Manuel Escobar. Memory requirements for llm training and inference. https://medium.com/@manuelescobar-dev/memory-requirements-for-llm-training-and-inference-97e4ab08091b, 2024.

[15] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts:

Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9, 2023.

[16] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to unseen domains: A survey on domain generalization, 2022.

[17] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[18] Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. Domain adaptation with BERT-based domain classification and data selection. In Colin Cherry, Greg Durrett, George Foster, Reza Haffari, Shahram Khadivi, Nanyun Peng, Xiang Ren, and Swabha Swayamdipta, editors, *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83, Hong Kong, China, November 2019. Association for Computational Linguistics.

[19] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1), oct 2021.

[20] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims, 2020.

[21] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification, 2020.

[22] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter, 2020.

[23] Longwei Wang, Chengfei Wang, Yupeng Li, and Rui Wang. Explaining the behavior of neuron activations in deep neural networks. *Ad Hoc Networks*, 111:102346, 2021.

[24] Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to graph: Interpreting language model neurons at scale, 2023.

[25] Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. Factuality challenges in the era of large language models, 2023.

[26] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. Benchmarking large language models in retrieval-augmented generation, 2023.

[27] Rahul Pandey, Hemant Purohit, Carlos Castillo, and Valerie L. Shalin. Modeling and mitigating human annotation errors to design efficient stream processing systems with human-in-the-loop machine learning. *International Journal of Human-Computer Studies*, 160:102772, 2022.

[28] Landing AI. Data-centric ai. https://landing.ai/data-centric-ai, 2024.

[29] Gilbert Badaro and Paolo Papotti. Transformers for tabular data representation: A tutorial on models and applications. *Proc. VLDB Endow.*, 15(12):3746–3749, 2022.

[30] Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. INFOTABS: Inference on tables as semi-structured data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online, July 2020. Association for Computational Linguistics.

[31] Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification. *Proc. VLDB Endow.*, 13(11):2508–2521, 2020.

[32] Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset, 2020.

[33] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, 2022.

[34] Barry Menglong Yao, Aditya Shah, Lichao Sun, Jin-Hee Cho, and Lifu Huang. End-to-end multimodal fact-checking and explanation generation: A challenging dataset and models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2733–2743, New York, NY, USA, 2023. Association for Computing Machinery.

[35] Xingyu Liu, Li Qi, Laurent Wang, and Miriam Metzger. Checking the fact-checkers: The role of source type, perceived credibility, and individual differences in fact-checking effectiveness. *Communication Research*, 10 2023.

[36] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. Generating scientific claims for zero-shot scientific fact checking. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2448–2460. Association for Computational Linguistics, 2022.

[37] Ana Luize Corrêa Bertoncini and Mauricio C. Serafim. Ethical content in artificial intelligence systems: A demand explained in three critical points. *Frontiers in Psychology*, 14, 2023.

[38] Zhisheng Chen. Ethics and discrimination in artificial intelligence-enabled recruitment practices. *Humanities and Social Sciences Communications*, 10:1–12, 2023.

[39] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection, 2022.

[40] Mohammed Saeed, Giulio Alfarano, Khai Nguyen, Duc Pham, Raphael Troncy, and Paolo Papotti. Neural re-rankers for evidence retrieval in the FEVEROUS task. In Rami Aly, Christos Christodoulopoulos, Oana Cocarascu, Zhijiang Guo, Arpit Mittal, Michael Schlichtkrull, James Thorne, and Andreas Vlachos, editors, *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 108–112, Dominican Republic, November 2021. Association for Computational Linguistics.

[41] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Generating fact checking explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online, July 2020. Association for Computational Linguistics.

[42] Jiasheng Si, Yingjie Zhu, and Deyu Zhou. Exploring faithful rationale for multi-hop fact verification via salience-aware graph learning, 2022.

[43] Jiasheng Si, Yingjie Zhu, and Deyu Zhou. Consistent multi-granular rationale extraction for explainable multi-hop fact verification, 2023.

[44] Preetam Prabhu Srikar Dammu, Himanshu Naidu, Mouly Dewan, YoungMin Kim, Tanya Roosta, Aman Chadha, and Chirag Shah. Claimver: Explainable claim-level verification and evidence attribution of text through knowledge graphs, 2024.

[45] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[46] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[47] David Magerman and Mitchell Marcus. Parsing a natural language using mutual information statistics. 05 1999.

[48] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing, 2018.

[49] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[51] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory, 2024.

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[53] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022.

[54] Guillermo Marco, Luz Rello, and Julio Gonzalo. Small language models can outperform humans in short creative writing: A study comparing slms with humans and llms, 2024.

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[56] Enzo Veltri, Donatello Santoro, Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Pythia: Unsupervised generation of ambiguous textual claims from relational data. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 2409–2412, New York, NY, USA, 2022. Association for Computing Machinery.

[57] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, 2019.

[58] Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheon-Eum Park, and Kyomin Jung. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3181–3185. ACM, 2021.

[59] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[60] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[61] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2024.

[62] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[63] Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. Expertprompting: Instructing large language models to be distinguished experts, 2023.

[64] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[65] Nvidia, :, Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report, 2024.

[66] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[67] Ronit Singhal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. Evidence-backed fact checking using rag and few-shot in-context learning with llms, 2024.

[68] Anthropic. Introducing claude, 2024. Accessed: 2024-09-25.

[69] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn,

Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna

Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam

Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer

Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024.

[70] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan

Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

[71] Sean Wu, Michael Koo, Lesley Blum, Andy Black, Liyo Kao, Fabien Scalzo, and Ira Kurtz. A comparative study of open-source large language models, gpt-4 and claude 2: Multiple-choice test taking in nephrology, 2023.

[72] Simone Papicchio, Paolo Papotti, and Luca Cagliero. QATCH: benchmarking sql-centric tasks with table representation learning models on your data. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[73] Xu Guo and Yiqiang Chen. Generative ai for synthetic data generation: Methods, challenges and the future, 2024.

[74] Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting llm quantization, 2024.

[75] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[76] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[77] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners, 2021.

[78] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Natural instructions: Benchmarking generalization to new tasks from natural language instructions, 04 2021.

[79] Lucas Graves. *Deciding what's true: The rise of political fact-checking in American journalism*. Columbia University Press, 2016.

[80] Caroline Jack. Lexicon of lies: Terms for problematic information. *Data & Society*, 3(22):1094–1096, 2017.

[81] Shujaat Mirza, Labeeba Begum, Liang Niu, Sarah Pardo, Azza Abouzied, Paolo Papotti, and Christina Popper. Tactics, threats and targets: Modeling disinformation and its mitigation. In Usenix, editor, *Usenix*, San Diego, 2023.

[82] Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. Claimbuster: The first-ever end-to-end fact-checking system. *Proc. VLDB Endow.*, 10(12):1945–1948, 2017.

[83] Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. That is a known lie: Detecting previously fact-checked claims. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online, July 2020. Association for Computational Linguistics.

[84] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[85] Morgan Scheuerman, Jialun Jiang, Casey Fiesler, and Jed Brubaker. A framework of severity for harmful content online, 08 2021.

[86] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[87] Naser Ahmadi, Joohyung Lee, Paolo Papotti, and Mohammed Saeed. Explainable fact checking with probabilistic answer set programming, 2019.

[88] Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. Scrutinizer: fact checking statistical claims. *Proceedings of the VLDB Endowment*, 13(12):2965–2968, 2020.

[89] Joseph E. Uscinski and Ryden W. Butler. The epistemology of fact checking. *Critical Review: A Journal of Politics and Society*, 25(2):162–180, 2013.

[90] B. Borel. *The Chicago Guide to Fact-Checking, Second Edition*. Chicago Guides to Writing, Editing, and Publishing. University of Chicago Press, 2023.

[91] Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4551–4558. ijcai.org, 2021.

[92] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[93] Mohamed H. Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. Exfakt: A framework for explaining facts over knowledge graphs and text. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 87–95, New York, NY, USA, 2019. Association for Computing Machinery.

[94] Tsvetomila Mihaylova, Preslav Nakov, Lluis Marquez, Alberto Barron-Cedeno, Mitra Mohtarami, Georgi Karadzhov, and James Glass. Fact checking in community forums, 2018.

[95] Max Glockner, Ieva Staliūnaitė, James Thorne, Gisela Vallejo, Andreas Vlachos, and Iryna Gurevych. Ambifc: Fact-checking ambiguous claims with evidence, 2023.

[96] Enzo Veltri, Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Pythia: Generating Ambiguous Sentences From Relational Data. https://www.eurecom.fr/~papotti/pythiaTr.pdf, 2021.

[97] Gordon Pennycook, Jonathon McPhetres, Yunhao Zhang, Jackson G Lu, and David G Rand. Fighting covid-19 misinformation on social media: Experimental evidence for a scalable accuracy-nudge intervention. *Psychological Science*, 31(7):770–780, 2020.

[98] Max Glockner, Yufang Hou, Preslav Nakov, and Iryna Gurevych. Missci: Reconstructing fallacies in misrepresented science. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 4372–4405, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[99] Max Glockner, Yufang Hou, and Iryna Gurevych. Missing counter-evidence renders NLP fact-checking unrealistic for misinformation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5916–5936, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[100] Mubashara Akhtar, Michael Schlichtkrull, Zhijiang Guo, Oana Cocarascu, Elena Simperl, and Andreas Vlachos. Multimodal automated fact-checking: A survey. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5430–5448. Association for Computational Linguistics, 2023.

[101] David W Test, Amy Kemp-Inman, Karen Diegelmann, Sara Beth Hitt, and Lauren Bethune. Are online sources for identifying evidence-based practices trustworthy? an evaluation. *Exceptional Children*, 82(1):58–80, 2015.

[102] Jad Doughman and Wael Khreich. Gender bias in text: Labeled datasets and lexicons. *arXiv preprint arXiv:2201.08675*, 2022.

[103] Julian Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan Boyd-Graber. Fool me twice: Entailment from Wikipedia gamification. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 352–365, Online, June 2021. Association for Computational Linguistics.

[104] Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. Averitec: A dataset for real-world claim verification with evidence from the web, 2023.

[105] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[106] Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. Fact-checking complex claims with program-guided reasoning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6981–7004. Association for Computational Linguistics, 2023.

[107] Kezhi Kong, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Chuan Lei, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Opentab: Advancing large language models as open-domain table reasoners, 2024.

[108] Shuo Yin, Weihao You, Zhilong Ji, Guoqiang Zhong, and Jinfeng Bai. Mumath-code: Combining tool-use large language models with multi-perspective data augmentation for mathematical reasoning, 2024.

[109] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[110] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[111] Giulia Bonino, Gabriele Sanmartino, Giovanni Gatti-Pinheiro, Paolo Papotti, Raphael Troncy, and Pietro Michiardi. Fine tuning a large language model for socratic interactions. In *AI for Education (AI4EDU)*, 2024.

[112] Tal Schuster, Adam Fisch, and Regina Barzilay. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*, 2021.

[113] Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic, 2021.

[114] Connor Shorten, Taghi Khoshgoftaar, and Borko Furht. Text data augmentation for deep learning. *Journal of Big Data*, 8, 07 2021.

[115] Akbar Karimi, Leonardo Rossi, and Andrea Prati. AEDA: An easier data augmentation technique for text classification. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2748–2754, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[116] Enzo Veltri, Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Data ambiguity profiling for the generation of training examples. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 450–463, 2023.

[117] Nancy X. R. Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. SemEval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (SEM-TAB-FACTS). In Alexis Palmer, Nathan Schneider, Natalie Schluter, Guy Emerson, Aurelie Herbelot, and Xiaodan Zhu, editors, *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online, August 2021. Association for Computational Linguistics.

[118] Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. Generating scientific claims for zero-shot scientific fact checking. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[119] Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. Zero-shot fact verification by claim generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 476–483, Online, August 2021. Association for Computational Linguistics.

[120] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *HLT-NAACL Demos*, pages 97–101. The Association for Computational Linguistics, 2016.

[121] W Samek. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[122] Zachary C. Lipton. The mythos of model interpretability. *Commun. ACM*, 61(10):36–43, sep 2018.

[123] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 395–405, New York, NY, USA, 2019. Association for Computing Machinery.

[124] Yi-Ju Lu and Cheng-Te Li. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online, July 2020. Association for Computational Linguistics.

[125] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[126] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[127] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. TaPas: Weakly supervised table parsing via pre-training. In *ACL*, pages 4320–4333. Association for Computational Linguistics, 2020.

[128] You Wu, Pankaj K. Agarwal, Chengkai Li, Jun Yang, and Cong Yu. Computational fact checking through query perturbations. *ACM Trans. Database Syst.*, 42(1):4:1–4:41, 2017.

[129] Preslav Nakov, David P. A. Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. In *IJCAI*, pages 4551–4558. ijcai.org, 2021.

[130] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *ICLR*, 2020.

[131] Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. In *EMNLP*, pages 1173–1186. ACL, 2020.

[132] Vivek Gupta, Riyaz A. Bhat, Atreya Ghosal, Manish Shrivastava, Maneesh Kumar Singh, and Vivek Srikumar. Is my model using the right evidence? systematic probes for examining evidence-based tabular reasoning. *Trans. Assoc. Comput. Linguistics*, 10:659–679, 2022.

[133] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, jun 2022.

[134] Immanuel Trummer. From BERT to GPT-3 codex: Harnessing the potential of very large language models for data management. *Proc. VLDB Endow.*, 15(12):3770–3773, 2022.

[135] Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *ACL*, pages 7567–7578, July 2020.

[136] Anish Das Sarma, Aditya G. Parameswaran, Hector Garcia-Molina, and Jennifer Widom. Synthesizing view definitions from data. In *ICDT*, pages 89–103. ACM, 2010.

[137] Yaacov Y. Weiss and Sara Cohen. Reverse engineering spj-queries from examples. In *PODS*, page 151–166. ACM, 2017.

[138] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[139] Hyunsoo Cho, Hyuhng Joon Kim, Junyeob Kim, Sang-Woo Lee, Sang goo Lee, Kang Min Yoo, and Taeuk Kim. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners. In *AAAI*, 2022.

[140] Julian Eisenschlos, Syrine Krichene, and Thomas Müller. Understanding tables with intermediate pre-training. In *EMNLP*, pages 281–296, November 2020.

[141] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[142] Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on Freebase with weak supervision. In *ACL*, pages 23–33, 2017.

[143] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[144] Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In *EMNLP*, pages 2253–2263. ACL, 2017.

[145] Steven Y Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. Genaug: Data augmentation for finetuning text generators. *arXiv preprint arXiv:2010.01794*, 2020.

[146] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

[147] Siyuan Qiu, Binxia Xu, Jie Zhang, Yafang Wang, Xiaoyu Shen, Gerard De Melo, Chong Long, and Xiaolong Li. Easyaug: An automatic textual data augmentation platform for classification tasks. In *Companion Proceedings of the Web Conference 2020*, pages 249–252, 2020.

[148] Markus Bayer, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics*, pages 1–16, 2022.

[149] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*, 2020.

[150] Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Do not have enough data? deep learning to the rescue! In *AAAI*, volume 34, pages 7383–7390, 2020.

[151] Thien Ho Huong and Vinh Truong Hoang. A data augmentation technique based on text for vietnamese sentiment analysis. In *International Conference on Advances in Information Technology*, pages 1–5, 2020.

[152] Milam Aiken and Mina Park. The efficacy of round-trip translation for mt evaluation. *Translation Journal*, 14(1):1–10, 2010.

[153] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020.

[154] Vincent Claveau, Antoine Chaffin, and Ewa Kijak. Generating artificial texts as substitution or complement of training data. *arXiv preprint arXiv:2110.13016*, 2021.

[155] Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. Data boost: Text data augmentation through reinforcement learning guided conditional generation. *arXiv preprint arXiv:2012.02952*, 2020.

[156] Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*, 2020.

[157] Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and William Campbell. A closer look at feature space data augmentation for few-shot intent classification. *arXiv preprint arXiv:1910.04176*, 2019.

[158] Congcong Wang and David Lillis. Classification for crisis-related tweets leveraging word embeddings and data augmentation. In *TREC*, 2019.

[159] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. *arXiv preprint arXiv:2004.12239*, 2020.

[160] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. *CoRR*, abs/2202.04538, 2022.

[161] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised hierarchical text classification, 2018.

[162] Yu Meng, Martin Michalski, Jiaxin Huang, Yu Zhang, Tarek F. Abdelzaher, and Jiawei Han. Tuning language models as training data generators for augmentation-enhanced few-shot learning. *CoRR*, abs/2211.03044, 2022.

[163] Orest Gkini, Theofilos Belmpas, Georgia Koutrika, and Yannis E. Ioannidis. An in-depth benchmarking of text-to-sql systems. In *SIGMOD*, pages 632–644. ACM, 2021.

[164] George Katsogiannis-Meimarakis and Georgia Koutrika. A deep dive into deep learning approaches for text-to-sql systems. In *SIGMOD*, pages 2846–2851. ACM, 2021.

[165] Nathaniel Weir, Prasetya Utama, Alex Galakatos, Andrew Crotty, Amir Ilkhechi, Shekar Ramaswamy, Rohin Bhushan, Nadja Geisler, Benjamin Hättasch, Steffen Eger, Ugur Çetintemel, and Carsten Binnig. Dbpal: A fully pluggable NL2SQL training pipeline. In *SIGMOD*, pages 2347–2361. ACM, 2020.

[166] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *EMNLP*, pages 3911–3921, 2018.

[167] Wei Chit Tan, Meihui Zhang, Hazem Elmeleegy, and Divesh Srivastava. Reverse engineering aggregation queries. *Proc. VLDB Endow.*, 10(11):1394–1405, 2017.

[168] Hao Li, Chee-Yong Chan, and David Maier. Query from examples: An iterative, data-driven approach to query construction. *Proc. VLDB Endow.*, 8(13):2158–2169, sep 2015.

[169] Meihui Zhang, Hazem Elmeleegy, Cecilia M. Procopiuc, and Divesh Srivastava. Reverse engineering complex join queries. In *SIGMOD*, page 809–820. ACM, 2013.

[170] Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3, 03 2002.

[171] Karen Kukich. Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics*, pages 145–150, Cambridge, Massachusetts, USA, June 1983. Association for Computational Linguistics.

[172] Linyong Nan, Lorenzo Jaime Yu Flores, Yilun Zhao, Yixin Liu, Luke Benson, Weijin Zou, and Dragomir Radev. R2d2: Robust data-to-text with replacement detection. *arXiv preprint arXiv:2205.12467*, 2022.

[173] Elena Soare, Iain Mackie, and Jeffrey Dalton. Docut5: Seq2seq SQL generation with table documentation. *CoRR*, abs/2211.06193, 2022.

[174] Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. Logical natural language generation from open-domain tables. In *ACL*, pages 7929–7942, 2020.

[175] A. Neveol, Dalianis, and S. Velupillai. Clinical natural language processing in languages other than english: opportunities and challenges. *Journal Biomed Semantic*, 9(12), 2018.

[176] Nancy XR Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. Semeval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (sem-tab-facts). *arXiv preprint arXiv:2105.13995*, 2021.

[177] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. Tabert: Pretraining for joint understanding of textual and tabular data. In *ACL*, pages 8413–8426, 2020.

[178] Jean-Flavien Bussotti, Luca Ragazzi, Giacomo Frisoni, Gianluca Moro, and Paolo Papotti. Unknown claims: Generation of fact-checking training examples from unstructured and structured data. 2024.

[179] Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. Covid-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2116–2129. Association for Computational Linguistics, 2021.

[180] John M. Carey, Andrew Markus Guess, Peter John Loewen, Eric Merkley, Brendan Nyhan, Joseph B. Phillips, and Jason Reifler. The ephemeral effects of fact-checks on covid-19 misperceptions in the united states, great britain and canada. *Nature Human Behaviour*, 6:236 – 243, 2022.

[181] Vincenzo Carrieri, Sophie Guthmuller, and Ansgar Wübker. Trust and covid-19 vaccine hesitancy. *Scientific Reports*, 13, 2023.

[182] Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. Zero-shot fact verification by claim generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 476–483. Association for Computational Linguistics, 2021.

[183] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *ICLR*. OpenReview.net, 2020.

[184] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: fact extraction and verification over unstructured and structured information. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[185] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7534–7550. Association for Computational Linguistics, 2020.

[186] Ozan Tonguz, Yiwei Qin, Yimeng Gu, and Hyun Hannah Moon. Automating claim construction in patent applications: The cmumine dataset. In Nikolaos Aletras, Ion Androutsopoulos, Leslie Barrett, Catalina Goanta, and Daniel Preotiuc-Pietro, editors, *Proceedings of the Natural Legal Language Processing Workshop 2021, NLLP@EMNLP 2021, Punta Cana, Dominican Republic, November 10, 2021*, pages 205–209. Association for Computational Linguistics, 2021.

[187] Jean-Flavien Bussotti, Enzo Veltri, Donatello Santoro, and Paolo Papotti. Generation of training examples for tabular natural language inference. *Proc. ACM Manag. Data*, 1(4):243:1–243:27, 2023.

[188] Shantipriya Parida and Petr Motlícek. Abstract text summarization: A low resource challenge. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5993–5997. Association for Computational Linguistics, 2019.

[189] Gianluca Moro and Luca Ragazzi. Semantic self-segmentation for abstractive summarization of long documents in low-resource regimes. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11085–11093. AAAI Press, 2022.

[190] Gianluca Moro and Luca Ragazzi. Align-then-abstract representation learning for low-resource summarization. *Neurocomputing*, 548:126356, 2023.

[191] Taehun Huh and Youngjoong Ko. Efficient framework for low-resource abstractive summarization by meta-transfer learning and pointer-generator networks. *Expert Syst. Appl.*, 234:121029, 2023.

[192] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *CoRR*, abs/2102.02503, 2021.

[193] William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. Towards language models that can see: Computer vision through the LENS of natural language. *CoRR*, abs/2306.16410, 2023.

[194] Wang-Chiew Tan, Yuliang Li, Pedro Rodriguez, Richard James, Xi Victoria Lin, Alon Y. Halevy, and Wen-tau Yih. Reimagining retrieval augmented language models for answering queries. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6131–6146. Association for Computational Linguistics, 2023.

[195] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S. Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[196] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

[197] Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1173–1186. Association for Computational Linguistics, 2020.

[198] Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Vera: A general-purpose plausibility estimation model for commonsense statements. *CoRR*, abs/2305.03695, 2023.

[199] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. Multimodalqa: complex question answering over text, tables and images. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[200] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics, 2019.

[201] Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. Automated fact-checking of claims from wikipedia. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 6874–6882. European Language Resources Association, 2020.

[202] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314, 2023.

[203] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, December 2019.

[204] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[205] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[206] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[207] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[208] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975, 2016.

[209] David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. Multivers: Improving scientific claim verification with weak supervision and full-document context. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 61–76. Association for Computational Linguistics, 2022.

[210] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics, 2018.

[211] Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1074–1084. Association for Computational Linguistics, 2019.

[212] Salud María Jiménez Zafra, Roser Morante, María Teresa Martín-Valdivia, and Luis Alfonso Ureña López. Corpora annotated with negation: An overview. *Comput. Linguistics*, 46(1):1–52, 2020.

[213] Radina Dobreva and Frank Keller. Investigating negation in pre-trained vision-and-language models. In Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad, editors, *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2021, Punta Cana, Dominican Republic, November 11, 2021*, pages 350–362. Association for Computational Linguistics, 2021.

[214] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché-Buc, editors, *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer, 2005.

[215] John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright, Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, and Christopher Hesse. Introducing chatgpt, 2022.

[216] Mohammed Saeed, Nicolas Traub, Maelle Nicolas, Gianluca Demartini, and Paolo Papotti. Crowdsourced fact-checking at twitter: How does the crowd compare with experts? In Mohammad Al

Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1736–1746. ACM, 2022.

[217] Kevin Matthe Caramancion. News verifiers showdown: A comparative performance evaluation of chatgpt 3.5, chatgpt 4.0, bing ai, and bard in news fact-checking. *CoRR*, abs/2306.17176, 2023.

[218] Miaoran Li, Baolin Peng, and Zhu Zhang. Self-checker: Plug-and-play modules for fact-checking with large language models. *CoRR*, abs/2305.14623, 2023.

[219] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[220] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised hierarchical text classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6826–6833. AAAI Press, 2019.

[221] Enzo Veltri, Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. Data ambiguity profiling for the generation of training examples. In *39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3-7, 2023*, pages 450–463. IEEE, 2023.

[222] Zhisong Zhang, Emma Strubell, and Eduard H. Hovy. A survey of active learning for natural language processing. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6166–6190. Association for Computational Linguistics, 2022.

[223] Max Glockner, Ieva Staliūnaitė, James Thorne, Gisela Vallejo, Andreas Vlachos, and Iryna Gurevych. AmbiFC: Fact-Checking Ambiguous Claims with Evidence. *Transactions of the Association for Computational Linguistics*, 12:1–18, 01 2024.

[224] Zhijiang Guo, Michael Sejr Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. *Trans. Assoc. Comput. Linguistics*, 10:178–206, 2022.

[225] Neema Kotonya and Francesca Toni. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online, November 2020. Association for Computational Linguistics.

[226] Jiasheng Si, Yingjie Zhu, and Deyu Zhou. Exploring faithful rationale for multi-hop fact verification via salience-aware graph learning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[227] Jiasheng Si, Yingjie Zhu, and Deyu Zhou. Consistent multi-granular rationale extraction for explainable multi-hop fact verification. *CoRR*, abs/2305.09400, 2023.

[228] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[229] Andreas Madsen, Siva Reddy, and Sarath Chandar. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42, 2022.

[230] Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *NeurIPS (Datasets and Benchmarks)*, 2021.

[231] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics, 2018.

[232] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online, November 2020. Association for Computational Linguistics.

[233] Julian Martin Eisenschlos, Bhuwan Dhingra, Jannis Bulian, Benjamin Börschinger, and Jordan Boyd-Graber. Fool me twice: Entailment from wikipedia gamification, 2021.

[234] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, and Dhruv Choudhary et al. The llama 3 herd of models, 2024.

[235] Acemoglu Daron and Restrepo Pascual. "the wrong kind of ai? artificial intelligence and the future of labour demand. *Cambridge Journal of Regions, Economy and Society*, 13(1):25–35, 2020.

[236] Davide La Torre, Francesco Paolo Appio, Hatem Masri, Francesca Lazzeri, and Francesco Schiavone. *Impact of artificial intelligence in business and society: Opportunities and Challenges*. Routledge, 2023.

[237] Jun Liu, Huihong Chang, Jeffrey Yi-Lin Forrest, and Baohua Yang. Influence of artificial intelligence on technological innovation: Evidence from the panel data of china's manufacturing sectors. *Technological Forecasting and Social Change*, 158:120142, 2020.

[238] Salman Bahoo, Marco Cucculelli, and Dawood Qamar. Artificial intelligence and corporate innovation: A review and research agenda. *Technological Forecasting and Social Change*, 188:122264, 2023.

[239] Erik Brynjolfsson, Danielle Li, and Lindsey R Raymond. Generative ai at work. Technical report, National Bureau of Economic Research, 2023.

[240] Erik Brynjolfsson, Daniel Rock, and Chad Syverson. Artificial intelligence and the modern productivity paradox. *The economics of artificial intelligence: An agenda*, 23:23–57, 2019.

[241] Laura Abrardi, Carlo Cambini, and Laura Rondi. Artificial intelligence, firms and consumer behavior: A survey. *Journal of Economic Surveys*, 36(4):969–991, 2022.

[242] Justyna Patalas-Maliszewska, Małgorzata Szmołda, and Hanna Łosyk. Integrating artificial intelligence into the supply chain in order to enhance sustainable production—a systematic literature review. *Sustainability*, 16(16):7110, 2024.

[243] Dan Zhang, LG Pee, and Lili Cui. Artificial intelligence in e-commerce fulfillment: A case study of resource orchestration at alibaba's smart warehouse. *International Journal of Information Management*, 57:102304, 2021.

[244] Erin E Makarius, Debmalya Mukherjee, Joseph D Fox, and Alexa K Fox. Rising with the machines: A sociotechnical framework for bringing artificial intelligence into the organization. *Journal of business research*, 120:262–273, 2020.

[245] Simon Johnson and Daron Acemoglu. *Power and progress: Our thousand-year struggle over technology and prosperity*. Hachette UK, 2023.

[246] Angus Deaton. The great escape: health, wealth, and the origins of inequality. 2024.

[247] A.V. Banerjee and E. Duflo. *Poor Economics: A Radical Rethinking of the Way to Fight Global Poverty*. PublicAffairs, 2012.

[248] Henrik Skaug Sætra. A framework for evaluating and disclosing the esg related impacts of ai with the sdgs. *Sustainability*, 13(15):8503, 2021.

[249] Assunta Di Vaio, Rosa Palladino, Rohail Hassan, and Octavio Escobar. Artificial intelligence and business models in the sustainable development goals perspective: A systematic literature review. *Journal of Business Research*, 121:283–314, 2020.

[250] Renato Camodeca and Alex Almici. Digital transformation and convergence toward the 2030 agenda's sustainability development goals: evidence from italian listed firms. *Sustainability*, 13(21):11831, 2021.

[251] Nenad Tomašev, Julien Cornebise, Frank Hutter, Shakir Mohamed, Angela Picciariello, Bec Connelly, Danielle CM Belgrave, Daphne Ezer, Fanny Cachat van der Haert, Frank Mugisha, et al. Ai for social good: unlocking the opportunity for positive impact. *Nature Communications*, 11(1):2468, 2020.

[252] Sheshadri Chatterjee, Ranjan Chaudhuri, and Demetris Vrontis. Ai and digitalization in relationship management: Impact of adopting ai-embedded crm system. *Journal of Business Research*, 150:437–450, 2022.

[253] Jun Liu, Huihong Chang, Jeffrey Yi-Lin Forrest, and Baohua Yang. Influence of artificial intelligence on technological innovation: Evidence from the panel data of china's manufacturing sectors. *Technological Forecasting and Social Change*, 158:120142, 2020.

[254] Patrick Mikalef, Kieran Conboy, Jenny Eriksson Lundström, and Aleš Popovič. Thinking responsibly about responsible ai and 'the dark side'of ai, 2022.

[255] Emmanouil Papagiannidis, Patrick Mikalef, Kieran Conboy, and Rogier Van de Wetering. Uncovering the dark side of ai-based decision-making: A case study in a b2b context. *Industrial Marketing Management*, 115:253–265, 2023.

[256] J Liedke and KE Matsa. Social media and news fact sheet. Technical report, 2022.

[257] Babur De los Santos, Ali Hortaçsu, and Matthijs R Wildenbeest. Testing models of consumer search using data on web browsing and purchasing behavior. *American economic review*, 102(6):2955–2980, 2012.

[258] Swapan Deep Arora, Guninder Pal Singh, Anirban Chakraborty, and Moutusy Maity. Polarization and social media: A systematic review and research agenda. *Technological Forecasting and Social Change*, 183:121942, 2022.

[259] Brent Kitchens, Steven L Johnson, and Peter Gray. Understanding echo chambers and filter bubbles: The impact of social media on diversification and partisan shifts in news consumption. *MIS Quarterly*, 44(4):1619–1649, 2020.

[260] Anja Lambrecht and Catherine Tucker. Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads. *Management Science*, 65(7):2966–2981, 2019.

[261] Piotr Sapiezynski, Avijit Ghosh, Levi Kaplan, Aaron Rieke, and Alan Mislove. Algorithms that "don't see color"' measuring biases in lookalike and special ad audiences. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 609–616, May 2022.

[262] WHO. A coordinated global research roadmap: 2019 novel coronavirus. Technical report, 2020.

[263] Shivam Gupta, Simone D Langhans, Sami Domisch, Francesco Fuso-Nerini, Anna Felländer, Manuela Battaglini, Max Tegmark, and Ricardo Vinuesa. Assessing whether artificial intelligence is an enabler or an inhibitor of sustainability at indicator level. *Transportation Engineering*, 4:100064, 2021.

[264] Ricardo Vinuesa, Hossein Azizpour, Iolanda Leite, Madeline Balaam, Virginia Dignum, Sami Domisch, Anna Felländer, Simone Daniela Langhans, Max Tegmark, and Francesco Fuso Nerini. The role of artificial intelligence in achieving the sustainable development goals. *Nature communications*, 11(1):1–10, 2020.

[265] Brendan Nyhan. Facts and myths about misperceptions. *Journal of Economic Perspectives*, 34(3):220–236, 2020.

[266] Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. *arXiv preprint arXiv:2103.07769*, 2021.

[267] Bertin Martens, Luis Aguiar, Estrella Gomez-Herrera, and Frank Mueller-Langer. The digital transformation of news media and the rise of disinformation and fake news. Working paper 2018-02, Digital Economy, Joint Research Centre Technical Reports, 2018.

[268] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017.

[269] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[270] Lesley Chiou and Catherine Tucker. Fake news and advertising on social media: A study of the anti-vaccination movement. Working paper, National Bureau of Economic Research, 2018.

[271] Andy Guess, Kevin Aslett, Joshua Tucker, Richard Bonneau, and Jonathan Nagler. Cracking open the news feed: Exploring what us facebook users see and share with large-scale platform data. *Journal of Quantitative Description: Digital Media*, 1, 2021.

[272] Petter Törnberg. Echo chambers and viral misinformation: Modeling fake news as complex contagion. *PLoS one*, 13(9):e0203958, 2018.

[273] Gordon Pennycook, Ziv Epstein, Mohsen Mosleh, Antonio A Arechar, Dean Eckles, and David G Rand. Shifting attention to accuracy can reduce misinformation online. *Nature*, 592(7855):590–595, 2021.

[274] Christy Galletta Horner, Dennis Galletta, Jennifer Crawford, and Abhijeet Shirsat. Emotions: The unexplored fuel of fake news on social media. *Journal of Management Information Systems*, 38(4):1039–1066, 2021.

[275] Manuel Mueller-Frank. As strong as the weakest node: The impact of misinformation in social networks. *Journal of Economic Theory*, 215:105773, 2024.

[276] Kevin Aslett, Zeve Sanderson, William Godel, Nathaniel Persily, Jonathan Nagler, and Joshua A Tucker. Online searches to evaluate misinformation can increase its perceived veracity. *Nature*, pages 1–9, 2023.

[277] Melis Kartal and Jean-Robert Tyran. Fake news, voter overconfidence, and the quality of democratic choice. *American Economic Review*, 112(10):3367–3397, 2022.

[278] Sander Van Der Linden. Misinformation: Susceptibility, spread, and interventions to immunize the public. *Nature Medicine*, 28(3):460–467, 2022.

[279] Sahil Loomba, Alexandre de Figueiredo, Simon J Piatek, Kristen de Graaf, and Heidi J Larson. Measuring the impact of covid-19 vaccine misinformation on vaccination intent in the uk and usa. *Nature Human Behaviour*, 5(3):337–348, 2021.

[280] Mohammed Saeed, Nicolas Traub, Maelle Nicolas, Gianluca Demartini, and Paolo Papotti. Crowdsourced fact-checking at twitter: How does the crowd compare with experts? In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1736–1746, October 2022.

[281] Youri Peskine, Paolo Papotti, and Raphaël Troncy. Detection of covid-19-related conpiracy theories in tweets using transformer-based models and node embedding techniques. In *MediaEval 2022, Multimedia Evaluation Workshop, 12-13 January 2023, Bergen, Norway*, 01 2023.

[282] Julian Just. Natural language processing for innovation search – reviewing an emerging non-human innovation intermediary. *Technovation*, 129:102883, 2024.

[283] Kisik Song, Karp Soo Kim, and Sungjoo Lee. Discovering new technology opportunities based on patents: Text-mining and f-term analysis. *Technovation*, 60-61:1–14, 2017.

[284] Irene Schellner. Japanese file index classification and f-terms. *World Patent Information*, 24(3):197–201, 2002.

[285] Ajay Agrawal, Joshua Gans, and Avi Goldfarb. *Prediction Machines, Updated and Expanded: The Simple Economics of Artificial Intelligence*. Harvard Business Press, 2022.

[286] Ritu Agarwal and Vasant Dhar. Editorial—big data, data science, and analytics: The opportunity and challenge for is research. *Information Systems Research*, 25(3):443–448, 2014.

[287] Eli Pariser. *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. Penguin, 2011.

[288] Eytan Bakshy, Solomon Messing, and Lada A Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.

[289] Ping Liu, Karthik Shivaram, Aron Culotta, Matthew A Shapiro, and Mustafa Bilgic. The interaction between political typology and filter bubbles in news recommendation algorithms. In *Proceedings of the Web Conference 2021*, pages 3791–3801, 04 2021.

[290] Marc Bourreau and Germain Gaudin. Streaming platform and strategic recommendation bias. *Journal of Economics & Management Strategy*, 31(1):25–47, 2022.

[291] Paul DiMaggio, Eszter Hargittai, et al. From the 'digital divide'to 'digital inequality': Studying internet use as penetration increases. *Princeton: Center for Arts and Cultural Policy Studies, Woodrow Wilson School, Princeton University*, 4(1):4–2, 2001.

[292] Avi Goldfarb and Jeff Prince. Internet adoption and usage patterns are different: Implications for the digital divide. *Information Economics and Policy*, 20(1):2–15, 2008.

[293] Chris Forman, Avi Goldfarb, and Shane Greenstein. Digital dispersion: An industrial and geographic census of commerical internet use, 2002.

[294] Anne-Britt Gran, Peter Booth, and Taina Bucher. To be or not to be algorithm aware: a question of a new digital divide? *Information, Communication & Society*, 24(12):1779–1796, 2021.

[295] Sophie Lythreatis, Sanjay Kumar Singh, and Abdul-Nasser El-Kassar. The digital divide: A review and future research agenda. *Technological Forecasting and Social Change*, 175:121359, 2022.

[296] Carolina De Alves, Carolina Salge, Elena Karahanna, and Jason Thatcher. Algorithmic processes of social alertness and social transmission: How bots disseminate information on twitter. *MIS Quarterly*, 46(1):229–259, 2022.

[297] Martin Müller, Marcel Salathé, and Per E Kummervold. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *Frontiers in Artificial Intelligence*, 6:1023281, 2023.

[298] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[299] OpenAI. Interaction with chatgpt, 2024. Accessed on: 2024-02-05. Available from: https://chat.openai.com/chat.

[300] Grazia Cecere, Clara Jean, Vincent Lefrere, and Catherine E Tucker. Trade-offs in automating platform regulatory compliance by algorithm: Evidence from the covid-19 pandemic. *Available at SSRN 3603341*, 2021.

[301] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding, 2022.

[302] Jean-Flavien Bussotti, Enzo Veltri, Donatello Santoro, and Paolo Papotti. Generation of training examples for tabular natural language inference. *Proceedings of the ACM on Management of Data*, 1(4):1–27, 2023.

[303] Youri Peskine, Damir Korencic, Ivan Grubisic, Paolo Papotti, Raphaël Troncy, and Paolo Rosso. Definitions matter: Guiding GPT for multi-label classification, 2023.

[304] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), jun 2020.

[305] Ekaterina Zhuravskaya, Maria Petrova, and Ruben Enikolopov. Political effects of the internet and social media. *Annual Review of Economics*, 12:415–438, 2020.

[306] Daron Acemoglu, Asuman Ozdaglar, and James Siderius. A model of online misinformation. Working paper, National Bureau of Economic Research, 2021.

[307] Venktesh V, Abhijit Anand, Avishek Anand, and Vinay Setty. Quantemp: A real-world open-domain benchmark for fact-checking numerical claims, 2024.

[308] Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification, 2020.

[309] Aman Rangapur, Haoran Wang, Ling Jian, and Kai Shu. Fin-fact: A benchmark dataset for multimodal financial fact checking and explanation generation, 2024.

[310] Fuxiao Liu, Yaser Yacoob, and Abhinav Shrivastava. Covid-vts: Fact extraction and verification on short video platforms, 2023.

[311] Alimohammad Beigi, Bohan Jiang, Dawei Li, Tharindu Kumarage, Zhen Tan, Pouya Shaeri, and Huan Liu. Lrq-fact: Llm-generated relevant questions for multimodal fact-checking, 2024.

[312] Neema Kotonya and Francesca Toni. Explainable automated fact-checking for public health claims. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online, November 2020. Association for Computational Linguistics.