Robust and Efficient Deep Learning
for Misinformation Prevention

by

Tal Schuster

B.Sc., Ben-Gurion University (2008)
M.Sc., Tel Aviv University (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

© Massachusetts Institute of Technology 2021. All rights reserved.

Author

Department of Electrical Engineering and Computer Science
July 12, 2021

Certified by

Regina Barzilay
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students
Robust and Efficient Deep Learning for Misinformation Prevention

by

Tal Schuster

Submitted to the Department of Electrical Engineering and Computer Science on July 12, 2021, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering

Abstract

Deep learning models have recently revolutionized the online environment, opening up many exciting opportunities to improve the user experience. These models, however, also introduce new threats by possibly creating or promoting misinformation, either intentionally or deliberately by malicious users. In this thesis, we present novel methods to fight the proliferation of misinformation online. We focus on the task of automated fact verification where the veracity of a given claim is examined against external reliable sources. We analyze the desired specifications of fact verification systems and describe the need for efficiency when operating against large comprehensive free text information resources, while ensuring robustness to challenging inputs and sensitivity to modifications in the referenced evidence. Our methods are general and, as we demonstrate, improve the robustness, efficiency, and interpretability of many other models beyond fact verification.

In the first part of this thesis, we focus on the robustness, sensitivity, and interpretability of sentence-pair classifiers. We present methodologies for identifying and quantifying idiosyncrasies in large curated datasets that undesirably lead models to rely on non-generalizable statistical cues. We demonstrate how contrastive evidence pairs can alleviate this issue by enforcing models to perform sentence-pair inference. To obtain such examples automatically, we develop a novel rationale-based denoising pipeline for modifying refuting evidence to agree with a given claim. In addition, we present a semi-automated solution for creating contrastive pairs from Wikipedia revisions and share a new large dataset.

In the second part, we turn to improve the inference efficiency of both the evidence retrieval and the claim classification modules, while reliably controlling their accuracy. We introduce new confidence measures and develop novel extensions to the conformal prediction framework. Our methods can dynamically allocate the required computational resources for each input to satisfy an arbitrary user-specified tolerance level. We demonstrate on multiple datasets that our well-calibrated decision rules reliably provide significant efficiency gains.

Thesis Supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science
Acknowledgments

The past several years at the MIT NLP group have been an amazing and indelible experience. Even though due to the pandemic I had to spend the last year of my degree working remotely from home, I felt part of an incredible group that help and support each other. First, I would like to thank my advisor Regina Barzilay who I am immensely grateful for her endless enthusiasm and readiness to guide and share her brilliant insights. I am thankful to her for encouraging me not to settle on partial solutions and for patiently advising me and helping me grow.

I am grateful to my committee members Tommi Jaakkola and Adam Tauman Kalai who have been extremely helpful and provided insightful ideas and valuable feedback to me. I had the privilege of also collaborating with Tommi towards the end of my PhD, and I have learned a lot from him. I was also fortunate to work with Adam during my internship at Microsoft Research and after, and I am thankful to him for patiently teaching me new topics and providing great feedback, as well as for excellent career advice.

I would also like to thank my other academic mentors that I was fortunate to have: Connie Lehman and Kevin Hughes for introducing me to the medical-related research, Amir Globerson for guiding me and supporting me along the way, and Vitaly Shmatikov for helping with the first SKII workshop and providing me valuable career advice. I am also grateful to my Master’s advisor at Tel Aviv University Lior Wolf for patiently teaching me the first steps in academic research and the fundamentals of Deep Learning.

I was fortunate to collaborate with many talented researchers during my PhD. I would like to thank Adam Fisch, Darsh J Shah, Roei Schuster, Ori Ram, and Adam Yala for working closely with me and co-leading several different projects during my PhD. I would also like to thank all my collaborators: Kyle Swanson, Brian Dontchos, Manisha Bahl, Brian Dontchos, Tally Portnoi, Randy Miles, Leslie Lamb, Serene Yeo, Daniel Filizzola, Enrico Santus, Yoav Meri, Beatrice Portelli, Jason Zhao, Giuseppe Serra, Amir M Tahmasebi, Clara Li, Ashwin Kalyan, and Alex Polozov. I have learned a lot from each and every one of my collaborators and I am grateful to them all.

I am also thankful to all of the other group members at MIT for their help and for the
many helpful feedback and valuable conversations: Yujia Bao, Benson Chen, Octavian Ganea, Jiang Guo, Wengong Jin, Bracha Laufer, Jiaming Luo, Peter Mikhael, Juan Ortiz, Yujie Qian, Victor Quach, Tianxiao Shen, Jeremy Wohlwend, Rachel Wu, Justin Xiang, and Santiago Ibañez. Thank you also to Dima Smirnov, Rotem Hemo, Jonathan Rosenfeld, Nili Persits, Adam Yaari, Amit Schechter, Aviv Netanyahu, Erez Kaminski, Shiyu Chang, Guang-He Lee, and all my other friends at MIT that supported me along the way. Special thank you to Arya (Adam’s Samoyed dog) for her playfulness that always brought me joy. I would also like to thank our lab’s administrative assistant Marcia Davidson for all her help.

I also thank all my friends and extended family from home for supporting me along the way and always being there for me when I needed. I would like to especially mention Ohad who left us too soon but his curious and positive spirit is still with me and helped me to go thorough difficult and challenging moments. I am also extremely thankful to my cousin Roei for not only being a great collaborator but also providing infinite support and encouragement when I most needed it.

I am grateful to Roxana for supporting me and providing great advice and care along the way and always being there for me.

Finally, I would like to thank my family for supporting me along my journey. My parents and sister’s careers inspired me to begin my PhD, and their support allowed me to complete it. I am grateful to my parents Ayelet and Gadi for always following my requests and desires and providing unconditional love and support, even when my plans led us to large geographical distances. I am immensely grateful to my sister Meromit and her husband Yaron for always believing in me and patiently helping me every time I needed. I am grateful to Ilai and Daniel for being the best nephews one could ask for and making sure that I don’t forget to play and joke.
Bibliographic Note

Portions of this thesis are based on previously published peer-reviewed papers. Chapter 6 is currently under review. The list of reference publications by chapter is provided below:


- **Chapter 4: Get Your Vitamin C! Robust Fact Verification with Contrastive Evidence.** Tal Schuster, Adam Fisch, Regina Barzilay. *In North American Chapter of the Association for Computational Linguistics (NAACL), 2021.* [242]


* Equal contribution.

All code and data are publicly available at: [https://github.com/TalSchuster](https://github.com/TalSchuster)
# Contents

## 1 Introduction

1.1 Potential Threats and Alternative Defenses .......................... 34
1.2 Fact Verification: Key Challenges ................................... 37
1.3 Robust Claim Classification ............................................ 40
1.4 Efficient Evidence Retrieval and Claim Classification .......... 43
1.5 Contributions .......................................................... 45
1.6 Outline ................................................................. 46

## 2 Limitations of Stylometry-based Detection Methods ............. 47

2.1 Introduction ............................................................ 48
2.2 Background and Related work ........................................ 49
   2.2.1 Stylometry for Human-written Text ............................ 49
   2.2.2 Machine-generated Text Detection ............................. 50
   2.2.3 Fake News Detection Approaches Beyond Stylometry ..... 51
2.3 Adversarial Setting .................................................... 52
2.4 Stylometry Fails to Detect Machine-Generated Misinformation 54
   2.4.1 Results .......................................................... 55
2.5 Stylometry Detects Machine-Human Impersonations .............. 57
   2.5.1 Results ........................................................ 58
2.6 Discussion ............................................................. 58

## 3 Automatic Fact Verification: Identifying and Mitigating Undesired Biases 61

3.1 Introduction ............................................................ 62
List of Figures

1-1 Overview of the automatic fact verification approach. . . . . . . . . . . . . 30

1-2 An hypothetical illustration of a fake Twitter account. The profile picture was generated with StyleGAN2 [131], and the text was generated with GPT-3 [29] by providing four tweets in the respective subject as examples in the prompt. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

1-3 Likelihood of labels in the FEVER dataset by the existence of certain bigrams in the claim. The top phrases are mostly correlated with the “refutes” label and the bottom phrases are highly correlated with “supports”. . . . . . 38

1-4 The 2020 Olympic games were postponed to 2021 but preserved the same name for branding purposes. The respective Wikipedia page was revised accordingly. Fact verification systems should be sensitive to this update. . . . 41

2-1 Examples of the fake class in our experiments. (a) In the news question answering (Section 2.4), a CNN article is presented with two examples of questions (bold) from newsQA [279] and Grover’s generated answer (red). The first answer is verified by a human annotator to be false and the second as true. (b) In article modification ($m = 6$) (Section 2.4), the negations are marked with a cross-line for deletions and underline for addition. (c) In the article extension case (Section 2.5), the bold red text is the generation of GPT-2 Medium to extend the prefix. . . . . . . . . . . . . . . . . . . . . . 53
3-1 An illustration of a REFUTES claim-evidence pair from the FEVER dataset (A) that is used to generate a new pair (B). From the combination of the ORIGINAL and manually GENERATED pairs, we obtain a total of four pairs creating symmetry. 63

3-2 Our fact-guided update pipeline. Given a claim which refutes incorrect information, a masker is applied to remove the contradicting parts from the original text while preserving the rest of the context. Then, the residual neutral text and claim are fused to create an updated text that is consistent with the claim. 81

3-3 Illustrating the flow of the masker module. 83

3-4 A summary of our pipeline. Given a sentence that is inconsistent with a claim, a masker is applied to mask out the contradicting parts from the original text while preserving the rest of the content. Then, the residual neutral text and claim are fused to create an updated text that is consistent with the claim. The Masker and the Two-Encoder Generator are trained separately. 86

4-1 In VITAMIN C, we focus on Wikipedia revisions in which the factual content changes. This example revision now supports an initially refuted claim. 98

4-2 The VITAMIN C dataset uses Wikipedia revisions to motivate four central fact verification tasks. Revision source: https://en.wikipedia.org/wiki/?diff=945689803 105

4-3 Test accuracy of models trained on a dataset of 100K combined SUP and REF examples from VITAMIN C and FEVER. The higher the ratio of VITAMIN C in the training data, the better the performance on adversarial evaluation sets (solid lines). The shaded areas represent standard error across three runs. 112
5-1 A demonstration of our conformalized cascade for $m$-step inference with set-valued outputs, here on an IR for claim verification task. The number of considered articles is reduced at every level—red frames are filtered, while green frames pass on. We only care about retrieving at least one of the admissible articles (starred) for resolving the claim.

5-2 Does calibrating using minimal admissible nonconformity scores (§5.4) improve predictive efficiency? We show predictive efficiency and success rates as a function of $\epsilon$. The efficiency of the min-calibrated CPs (both cascaded and non-cascaded) is significantly better (i.e., lower) than standard CP across all tasks, especially at critical values of $\epsilon$. The min-calibrated CP’s accuracy hugs the diagonal, allowing it to remain valid, but be far less conservative (resulting in better efficiency).

5-3 How effective is cascaded conformal prediction (§5.5) at pruning the candidate space? Incorporating early rejection via the CP cascade reduces the fraction of required p-value computations. Larger tolerance levels ($\epsilon$) allow for more aggressive pruning rates.

6-1 Our CAT model $G$ can save computational resources by exiting early on certain inputs—while guaranteeing predictive consistency with the full model $F$. 

6-2 Confidence levels given by our meta model regarding the consistency of our prediction as computation progresses. Ex.1 from the VitaminC fact verification dataset is “easy”, and is classified consistently by all early classifiers $F_k$ (Supports). The meta confidence captures this, and increases with time. Ex.2 is harder—and the prediction changes (Refutes/NEI) as it propagates though the Transformer layers. Appropriately, the meta confidence is low. The exact exit layer of $G$ is determined as a function of a user-specified tolerance $\epsilon$, see Eq. Equation 6.1.
6-3 Classification results (dev). While both our CP-based methods give valid consis-
tencies (above diagonal), \textit{shared calibration} generally results in earlier exits. This advantage is especially pronounced at smaller tolerance levels (right-hand side), where it significantly outperforms other approaches. Our meta-learned confidence measure $M_k$ improves over using the softmax response as a drop-in replacement, especially for tasks with larger $|\mathcal{Y}|$. Note that we care more about the right-hand side behavior, (i.e., larger $1 - \epsilon$), as it corresponds to higher consistency.

6-4 Distribution of exit layers per tolerance level $\epsilon$ for the IMDB task (dev set) with Shared/ Meta. Larger $\epsilon$ allows the CAT model to shift its predictions earlier by permitting for more inconsistencies with the full model $\mathcal{F}$.  

6-5 Regression results for STS-B (dev).

A.1.1 Distribution of claims in the VITAMINC dataset by the topic of the originated Wikipedia article.

A.1.2 Probability density function of claim-evidence overlap for different labels in the dataset. The overlap is computed as the ratio of mutual bigrams in the two sentences.

A.3.1 The prompt used for GPT-3 few-shot claim extraction.

A.3.2 The prompt used for GPT-3 few-shot fact verification predictions on non-numerical claims (examples 2 and 4 in Table A.3.1. We follow the few-shot setting of [29] for ANLI.

A.3.3 The prompt used for GPT-3 few-shot fact verification predictions on numerical claims (examples 1 and 3 in Table A.3.1).
B.2.1 How does our conformal method compare to non-conformal baselines? We show predictive efficiency and accuracy across various target accuracies (values of $1 - \epsilon$). As expected, with a large enough calibration set, the performance of the threshold heuristic is similar to CP. The top-$k$ threshold is harder to tune, resulting in lower than the desired accuracy for some values of IR, and sometimes in unnecessary large prediction sets. Specific values of $\epsilon$ are compared in Table B.2.2.

B.3.1 Success rate against tolerance thresholds with different methods for MHT correction (validation set). Not applying any correction leads to a invalid classifier (the success rate is below the diagonal). The Bonferroni method is conservative, leading to a valid classifier, but sometimes with a higher accuracy rate than necessary (a result of having excessively large $C_n$). The Simes correction works for the QA and IR tasks and provides a tighter bound for them. The Simes correction does not work for the DR task, likely due to a violation of its MTP2 assumption, but the Bonferroni method provides a relatively tight correction there—especially for small $\epsilon$.

C.2.1 Distribution of exit layers per tolerance level $\epsilon$ (dev sets) with our Shared/Meta Albert-xlarge model. See Figure 6-4 for IMDB.

C.2.2 Development set results with an Albert-base 12-layers model as $F$.

C.2.3 Development set results with an RoBERTa-large 24-layers model as $F$. 
List of Tables

2.1 Results: Section 2.4. We report (macro) F1 score and overall accuracy, as well as precision and recall of the “fake” class. Zero-shot performance (not included) was very low in all cases.

2.2 Detection results on the attacks of Section 2.5 in a zero-shot setting and the adaptive setting (where the discriminator is fine-tuned to a specific attacker). We report (macro) F1 score and overall accuracy. Precision, and recall of the “fake” class are reported as well.

3.1 List of recently released datasets for fact verification. The bottom collections focus on evaluation with adversarial or challenging examples. Our VitaminC (Chapter 4) and Symmetric (Section 3.4) are also included.

3.2 Top 15 LMI-ranked bigrams in the train set of FEVER for REFUTES with its \( p(l|w) \). The corresponding figures for the development set are also provided.

3.3 Top 15 LMI-ranked bigrams in the train set of FEVER for SUPPORT.

3.4 Top 15 LMI-ranked bigrams in the train set of FEVER for NOT ENOUGH INFO. * denotes computationally infeasible, as there are no occurrences.

3.5 Examples of pairs from the Symmetric Dataset and, above each line, the related pair from the FEVER dataset. Each claim-evidence pair holds the relation described in the right column. Crossing the Symmetric sentences with the FEVER ones creates two additional claim-evidence pairs with an opposite label (see Figure 3-1).
3.6 Classifiers’ accuracy on the Supports and Refutes cases from the FEVER DEV set and on the Generated pairs for the Symmetric Test Set in the setting of without (BASE) and with (R.W) re-weight.

3.7 Re-weighted LMI and conditional probabilities ($l = \text{Refutes}$) for the bigrams from Table 3.2. The weights were obtained following the optimization of Equation 3.3 on the training set which contains three labels.

3.8 Human evaluation results for our model’s outputs for the fact update task (top) and for the data augmentation task (bottom). The left part of the table shows the geometric SARI score with the three F1 scores that construct it. The right part shows the human’s scores in a 1-5 Likert scale on grammatically of the output sentence and on agreement with the given claim.

3.9 Classifiers’ accuracy on the symmetric DEV and TEST splits. The right column (+TURK) shows the accuracy on the TEST set extended to include the 500 responses of turkers for the simulated process and the refuted pairs that they originated from. The BERT classifiers were trained on the FEVER training dataset augmented by outputs of the different methods.

3.10 Results of different values of $\lambda$ for the masker with and without syntactic guidance. The left three columns describe the accuracy and average mask size (% of the sentence) over the FEVER development set with the masked evidence and a neutral target label. The right three columns contain the precision, recall and F1 of the masks that we have human annotations for.

3.11 We compare our model outputs against different models. Each example is showing the two input sentences following the output of each model. The Concat model setting is similar to the M. Concat one but the original text is left unmasked. For the Claim Ext. model, only the claim sentence is given as input.
4.1 Examples of *non-factual* revisions vs. *factual* revisions, and the claims associated with the later. Factual updates change the outcome (i.e., true or false) of a claim that might be in question. Accordingly, the verdict of a classifier should change based on the version presented. Modified words are underlined and colored.

4.2 Number of claim-evidence pairs in VITAMINC. Breakdowns of real vs. synthetic revisions are presented on the left and right of each cell, respectively.

4.3 Factual revision flagging scores for models aware of the full sentence-pair (full) and aware only of the modified words (diff). We use ALBERT-base.

4.4 Test accuracy of fact verification and NLI models. VITAMINC-trained models are more robust to adversarial examples and more sensitive to contrastive contexts. The rightmost column shows the percent of FEVER claims in which the prediction flipped when presented with contrastive contexts.

4.5 The distant token-level supervision of VITAMINC improves the edit prediction, and as result identifies the anchoring words (rationales) more accurately.

4.6 Factually consistent generation results. Higher is better for all scores and the max value is 100 (except for BLEURT). $f_{\text{verdict}}$ is the score of our VITAMINC-trained ALBERT-base model on the outputs. For manual evaluation, outputs were rated by their grammaticality and by how much the evidence supports the claim (SUP). For reference, human-written pairs received 75.75 and 76.0 average scores for Grammar and SUP, respectively.

4.7 Example outputs for expressing claims that reflect the factual changes in a single Wikipedia revision. The BART-base model is trained on VITAMINC data, while GPT-3 is applied in a 2-shot setting with a temperature of 0 or 0.7 (see Appendix A.3). The revision $(s_{t-1}, s_t)$ is given to the model as a single sentence, where the edits are between curly brackets. The human-written claim is provided for reference. The rightmost column contains the prediction of our ALBERT-xlarge $f_{\text{verdict}}(c, s_t)$ model (trained on VITAMINC) when using the generated claim.
5.1 CP results for specific tolerance levels $\epsilon$. Each line shows the empirical accuracy (Acc.) and the (raw) size of the prediction set for our two methods as compared to regular CP, per target accuracy rate ($1 - \epsilon$). The amortized computation cost of the cascade, a consequence of pruning, is also given.

6.1 Additional meta features used as input to the meta early exit classifier, $\mathcal{M}_k$. Where specified, the probability $p_k$ is taken from the model’s early softmax. $p_k^{\text{max}}$ and $p_k^{\text{diff}}$ are only used for classification tasks.

6.2 Task dataset and label space sizes. The rightmost column reports either test accuracy (classification) or Pearson-correlation (regression). *We downsample the 63K public development set to expedite validation.

6.3 Classification results (test) for specific tolerance levels. We report the accuracy lower bound guaranteed by our CP methods in parentheses. Shared/Meta is reliably the most efficient method (and is $\epsilon$-consistent). Greyed rows reflect approaches without guarantees; our CAT approaches with guarantees are presented below them.

6.4 Test results for the STS-B regression task.

A.1.1 Estimated distribution of claims in the VITAMINC, based on manual annotations of 100 randomly sampled claims from the development split of the real and synthetic subsets. An example claim from each category is provided for reference.
See Table A.3.1 for lines 4-5. GPT-3 fact verification predictions on examples from the VITAMIN test dataset (examples 1.5-1.7 and 5.3-5.4 were manually modified to examine the model’s behavior). We follow the few-shot setting of [29] for ANLI (see Figures A.3.2 and A.3.3). The bold spans are for presentation and are not part of the input. Our VITAMIN-trained ALBERT classifiers predicted correctly on all these examples (though they weren’t picked this way). The GPT-3 few-shot succeeds on some examples and even expresses sensitivity to evidence in lines 2.1-2.2. In several cases, however, GPT-3 abstains from a True/False verdict, even when provided with strong evidence (see “Neither” predictions). Line 1.4 shows an example where GPT-3’s verdict is opposite of the provided evidence. Only when rephrasing the claim to exactly overlap with the evidence, it predicts an agreement.

Lines 4-5 of Table A.3.1

Fact verification Complementary results for Table 4.4 with a BERT-base model.

Example masks produced by the word-level rationale model for identifying anchoring words in the evidence that are responsible for the classifiers verdict regarding the claim (shown with green background). Masking these words leads the classifier to predict NEI instead of what would have been SUP or REF.

Example outputs of the BART-base used for generating factually consistent revisions given the old version $s_{t-1}$ and the updated claim we wish to support. The “ground-truth” $s_t$ is provided for reference.
A.5.3 Additional examples for Table 4.7. Example outputs for extracting claims that express the factual change in a Wikipedia revision. The BART-base model is trained on VITAMIN-C data and GPT-3 is applied in a 2-shot setting with a temperature of 0 or 0.7. The revision \((s_{t-1}, s_t)\) is given to the model as a single sentence visualization where the edits are between curly brackets, preceded by the article’s title. The human-written claim is provided for reference. The prediction of our ALBERT-xlarge VITAMIN-C-trained model \(f_{\text{verdict}}(c, s_t)\) on the generated claim against \(s_t\) is also reported in the rightmost column.

B.1.1 Nonconformity measures used in our experiments.

B.2.1 CP results on the SQUAD 2.0 dataset.

B.2.2 Non-CP baseline results on the test set for different target accuracy values. We compare the predictive efficiencies of the CP methods \(|C_n|\) and \(|\hat{C}_{\min n}|\) to those of the baseline methods \(|\mathcal{B}|\).

C.1 Results (dev) using the naive development set calibration method (see §6.3.3). This method tunes the early exit thresholds to get efficient \(\epsilon\)-consistent predictions on a development set, but does not guarantee that prediction will be \(\epsilon\)-consistent on new data. “Consist.” measures the empirical consistency on a test set, from which we compute a guaranteed lower bound (“Bound”) to 99% confidence. The bound is significantly lower than our target \(1 - \epsilon\), and the measured consistency in our experiments also falls slightly below \(1 - \epsilon\) in some cases.
C.2 Results (dev) of our Shared model on the classification tasks using different nonconformity measures. \( p_k^{\text{diff}} \) and \( p_k^{\text{max}} \) are defined in Table 6.1. \( D_{\text{KL}}(p_{k-1}||p_k) \) is the Kullback-Leibler Divergence between the previous layer’s softmax outputs and the current layer, and \( \mathcal{H}(p_k) \) is the entropy of the softmax outputs. Our CP-based Shared method provides the guaranteed consistency with any measure, even random. The benefit, however, of using a better measure is in confidently exiting earlier. Our Meta measure allows the use of least Transformer layers meeting the consistency requirement with enough confidence.

C.2.2 Results (dev) with the Shared/ Meta model comparing the non-conditional (unlabeled) with the conditional calibration. Given the availability of labeled training data, conditional calibration allows a refined guarantee of consistency only over the inputs that are predicted correctly by \( \mathcal{F} \). In return, this leads to improved efficiency as the meta classifier performs better on these inputs. We report both conditional and non-conditional consistency results here and observe the validity of both calibration methods. In chapter 6 we use non-conditional calibration for all classification tasks as we assume an unsupervised setting.

C.3.1 Number of Transformer layers used for example inputs from the task’s test sets with our Shared/Meta CAT with a tolerance level of \( \epsilon = 0.1 \) .

C.3.2 Number of Transformer layers used for example inputs from the task’s test sets with our Shared/Meta CAT with a tolerance level of \( \epsilon = 0.1 \).
Chapter 1

Introduction

Since the invention of the personal computer and the internet, humans gradually shifted from reading printed newspapers, books and encyclopedias, to consuming most content online [183]. The online environment allows users to receive constant updates from a large range of sources all over the globe without delay. While there have been many advantageous to this technologies, such as helping people to stay informed on news and events all over the world, it also created a new domain for adversaries to operate in [149]. Even though the threat of misinformation and the difficulty of verifying information have been long-standing concerns from the early history of humanity, these recently developed technologies opened up a new arena that requires humans to modify their literacy skills [28, 108, 156] and develop a different kind of alertness [300].

At the same time, the advancements in computational infrastructure and the vast availability of data have also contributed to the rapid improvements in deep learning methods. Most of the current best-performing systems in the field of computer vision (CV) and natural language processing (NLP) are based on deep learning models. These artificial intelligent (AI) solutions have become an integral part of the online environment, ranging from translators [10] to dialog assistant tools [264] and recommendation systems [47] to improve search results [56] and advertisements. Lately, the remarkable performance of deep language models [e.g., 29, 135] has even demonstrated the possibility of automatically generating content by either summarizing articles, suggesting edits, or producing complete paragraphs from scratch [317]. In general, the main objective of these AI tools is to improve
the user experience and increase engagement. By default, however, they do not provide any guarantees on the accuracy of the content that they provide. In fact, since users are more likely to share misinformation [292], without intervention these tools might even prioritize inaccurate facts when suggesting or creating content.

The potential danger and harmfulness of fake news and misinformation [149, 300] have brought up the need for developing techniques to help users assess the veracity of content online. Some success in reducing the spread of misinformation was obtained by increasing users’ awareness of the subject [204], and publishing fact-checking articles on topics of high public interest (e.g., see PolitiFACT.com, Snopes.com, toolbox.google.com/factcheck). Yet, manually verifying claims by experts at scale is infeasible, and with crowd-sourcing, it could be challenging to promise high accuracy [227, 228]. Therefore, developing accurate, robust, and fast automatic tools for fact verification is of high interest. Such tools can be used to identify false or suspicious claims and present effective warnings to the reader [127, 253].

Fact verification involves evaluating the veracity of an arbitrary claim against external evidence from reliable sources [270, 272]. Due to the complexity of the task, it is usually divided into three steps, performed either by separate models or by a joint one [e.g., 98, 191, 272, 316]. First, the scope of search is narrowed into a small set of documents that are likely to provide supporting or refuting evidence to the given claim. Second, a more fine-grained selection is made to identify the sentences that contain the most relevant information. Finally, the retrieved evidence is used to classify the veracity of the claim. Specifically, the claim is classified as true if it is supported by the evidence, false if it is refuted, or “neither” if...
the evidence does not provide enough information. These three steps are illustrated in Figure 1-1.

As we discuss in Chapter 2, other approaches could also be used for combating misinformation. However, they either rely on additional features that are not always available (e.g., the provenance of the claim or propagation patterns) or provide limited efficacy [245, 286]. A key benefit of the automatic fact verification approach is that it resembles the strategy of human experts. Specifically, experts are typically required to explain their verdict by pairing it with compelling evidence. Therefore, the evidence-based design of fact verification models provides an advantage over other alternatives. More generally, we would like a fact verification system to satisfy the following requirements:

- **Comprehensiveness**: covering a wide range of topics of interest against large free text information resources.

- **Robustness**: providing consistently high accuracy, even when faced with text manipulations or challenging inputs.

- **Interpretability**: explaining the prediction by matching it with respective evidence that justifies the verdict.

- **Sensitivity**: modifying the verdict in accordance to changes and developments in the real world, as reflected by revisions to information resources.

- **Efficiency**: providing fast predictions at scale, enabling real-time applications.

As we shall see, these specifications are not trivial to attain. For instance, comprehensiveness can come at odds with efficiency, as the complexity of an iterative retriever grows linearly with the size of the corpus.

**Aligning the verdict with the evidence.** Interestingly, even though fact verification models always couple their prediction with evidence, interpretability is not guaranteed. In practice, there could be discrepancies between the prediction and the evidence that is supposed to justify it. The reason for this lies in the claim-evidence classifier (step 3 in Figure 1-1). Correctly predicting the relation between two sentences requires advanced
reasoning capabilities, which are known to be challenging for many NLP systems [17, 21]. Without sufficient control, deep learning models are likely to resort to easier cues in the training dataset. As we discuss in Chapter 3, idiosyncrasies in fact verification datasets can lead models to rely on certain phrases in the claims, potentially ignoring the evidence entirely.

Models that follow cues in the claims instead of performing claim-evidence inference, break the aforementioned specifications. First, they rely on dataset-specific features that are unlikely to represent the real data distribution, affecting their robustness. Second, their evidence-based explanations are not necessarily aligned with the actual prediction, hurting their interpretability. Lastly, they are unlikely to be sensitive to the presence of new, updated evidence since the prediction is independent of the evidence. In this work, we present methods to enforce evidence-dependant inference, thereby significantly increasing the robustness, interpretability, and sensitivity of models.

Supporting large and dynamic information resources. Throughout this thesis, we use the English Wikipedia as our external information resource. Even though most of its articles are open to public editing, Wikipedia is carefully maintained by a large and active community that quickly corrects wrong information and openly discusses controversial topics [143]. Indeed, Wikipedia has established itself as a comprehensive and objective resource that users commonly refer to [19]. For these reasons and thanks to its open structure, we develop and evaluate our models against Wikipedia. Our methods, however, are general and can be adapted to any other domain.

Operating over large resources, such as whole Wikipedia, requires efficient inference. Some systems expedite retrieval by pre-computing and efficiently indexing entries [124, 129, 208, 306]. However, such solutions are less trivial when wishing to follow the most recent evidence—especially in a dynamic environment with frequent revisions. Other approaches, such as reducing the complexity of the models, come with the cost of an unpredictable decrease in accuracy. Instead, we propose methods that provide better efficiency while reliably controlling the required performance with a user-specified tolerance level. We develop such methods both for the evidence retrieval step, and for the classification task,
thus facilitating operating with larger information resources and over many claims.

In the reminding of this chapter, we discuss specific examples of misinformation threats and matching defenses (§1.1), following by a brief summary of our methods for improving the robustness and efficiency of fact verification models (§1.2-§1.4). Then, we list the main contributions of this work (§1.5) and outline the rest of the thesis (§1.6).
1.1 Potential Threats and Alternative Defenses

Media outlets have long focused on information integrity; however, recent advances in NLP, combined with the digital environment, have imposed new threats that demand the use of new defenses. In particular, malicious actors can take advantage of NLP tools to expedite the propagation of false information, or even to automate the process of generating the content itself. A ban on such technologies is not a panacea since most users are making good use of them. Therefore, advanced intention-based or outcome-based filtering is necessary. Notably, an entirely manual human-based solution is insufficient, as dealing with a large scale of misinformation is part of the new challenge. As a result, the development of AI-based defenses is becoming a Red Queen’s race against attackers.

**Machine-human impersonations.** Many online interactive platforms, such as social media and forums, rank postings by their popularity, determined by the interactions of other users. Therefore, the value of possessing and controlling many accounts is clear, and, unsurprisingly, we observe an increasing amount of fake accounts over time [94, 174]. The improvements in deep learning generative algorithms ease the process of creating fictional identities at scale. Figure 1-2 demonstrates a fake Twitter user with a GAN-generated [86, 131] profile picture and GPT-3-generated [29] tweets.

The fight against machine-generated fictional profile pictures and fake videos (often called DeepFake) has become a major challenge, leading to a public competition in 2020 with one million dollar prizes [60]. Typically, similar to the generation model, the detection solution is also based on a deep learning model, this time with a binary classification head. These models aim to identify subtle distributional features that are specific to the images that machines generated. Usually, they are trained in a supervised fashion with a large collection of real and fake images. Alternatively, the classifier (often called discriminator) could be trained jointly with a generative model in an alternating two-player optimization process [86].

In the textual domain, as outputs of large language models create more fluent and realistic looking outputs [29, 317], it is becoming increasingly challenging for human
Readers to identify such texts \cite{113,241}. As a result, malicious actors can use such models to automatically generate content for fake accounts to share, potentially increasing their visibility and likeability, without exposing their fakeness. Similar to the DeepFake defenses, several distributional-based solutions were recently proposed \cite{218,292,317}. Such solutions aim to identify subtle differences between the language used by humans and the generated texts of deep learning models.

This approach, named stylometry, has previously shown remarkable success in identifying the provenance of a texts \cite{1,3,32,87,182,187,216,282}. However, the increasing performance of large language models reduces its effectiveness \cite{20,286}. Moreover, unlike fictional profile pictures that are undoubtedly maleficent, generated text by itself is not necessarily a risk. For example, people benefit greatly from writing assistants and speech-to-text software. Therefore, it is crucial to ensure that the delivered content is factually accurate, rather than simply identifying its origin.

**Identifying false claims.** Focusing on the factuality of texts is of significant importance as it is independent of the text’s provenance. Verifying the veracity of claims is useful both for reviewing human-written articles and for avoiding factual mistakes introduced by language models. As discussed in the next section, however, automatically inferring the truthfulness of claims is very challenging. In this work, we present methods to improve the performance of fact verification models by increasing their robustness to adversarial examples and their sensitivity to changes in the world. We also present techniques for improving the models’ interpretability and increasing their efficiency.
**Non-textual defenses.** While fact verification focuses on the content of the text, other information, when available, could help in identifying the intent of the writer. This includes propagation patterns over social media and other metadata about the origin of the claim [35, 93, 322, 140, 166]. Also, some platforms can leverage users’ feedback to identify suspicious content [280]. When possible, such information can be combined with fact verification models to improve overall accuracy.
1.2 Fact Verification: Key Challenges

Over the past several years, multiple datasets have been proposed for training and developing fact verification models. Many of these datasets are based on fact-checking websites [9, 12, 97, 213, 289, 299], and therefore provide relatively small sets of hundreds to a few thousands of claims. The FEVER dataset [272] collected 185K claims, extracted by crowd-workers from Wikipedia articles, initiating two shared tasks that led to the development of multiple deep learning models [275, 276]. Following FEVER, other Wikipedia-based fact verification datasets were created, focusing on tabular data [44], multi-hop reasoning [122], and article references [240]. In Chapter 4, we present a dataset with over 400K claim-evidence pairs that we collected with the intention to address some of the challenges discussed below.

The main advantage of the fact verification approach over other claim classification solutions is its inherent use of rationales in the form of an evidence sentence (See Figure 1-1). Rationales provide users a valuable tool for interpreting the predictions of black-box deep learning models [155, 223]. For example, a rationale-based sentiment analysis model for movie reviews is expected to highlight the most praising spans in a positive review, justifying its prediction. Similarly, in fact verification, the paired evidence should explain the verdict of the claim. In practice, however, this is not necessarily the case.

Discrepancies between the retrieved evidence and the verdict. We expect a fact verification model to estimate $\mathbb{P}(Y = y | X_{\text{claim}}, X_{\text{evidence}})$, where $y \in \{\text{SUP}, \text{REF}, \text{NEI}\}$ and $X_{\text{claim}}, X_{\text{evidence}}$ are the claim and evidence sentences, respectively. This modeling, however, does not ensure us that the evidence is indeed taken into consideration when predicting $y$. In order for the evidence to provide a faithful explanation, we must also require that $\mathbb{P}(Y = y | X_{\text{claim}}, X_{\text{evidence}}) \neq \mathbb{P}(Y = y | X_{\text{claim}})$. More specifically, the claim alone should not provide any clues regarding the relation with the evidence without observing it, meaning $\mathbb{P}(Y = y | X_{\text{claim}}) = 1/3$. Unfortunately, fact verification models trained on popular datasets are unlikely to satisfy this requirement.

Datasets for sentence-pair inference are prone to include annotation artifacts [95, 179, 211, 281]. Examining the popular FEVER dataset, we find that a claim-only classifier (i.e.,
Figure 1-3: Likelihood of labels in the FEVER dataset by the existence of certain bigrams in the claim. The top phrases are mostly correlated with the “refutes” label and the bottom phrases are highly correlated with “supports”.

modeling \( P(Y = y|X_{\text{claim}}) \) achieves 61.7% accuracy, far above the desired 33.3%. In comparison, the best performing model in the public shared task \[275\] that uses evidence achieved only 8% absolute higher accuracy \[191\]. The reason for the high performance of the claim-only classifier is that the claims in the dataset, created by crowd-workers, contain statistical cues to their label. As demonstrated in Figure 1-3, negations are likely to appear in false claims (i.e., that are paired with refuting evidence). These idiosyncrasies are widespread in the data, going beyond simple negations and even including certain entities. Relying on these statistical cues breaks our requirement of conditioning the prediction on the relation of the evidence with the claim, introducing discrepancies between the presented evidence and the true rationale.

In addition to affecting interpretability, relying on statistical cues also reduces models’ responsiveness to evidence updates and hurts their robustness in the face of unfamiliar examples. In §1.3 we discuss our proposed methods for addressing this issue.

Scaling up the retrieval and classification modules. Much of the recent success of deep learning models for NLP comes with large models that include millions of parameters and are slow to use. A naive application of such models would probably not be suitable for retrieving evidence and classifying claims. Thousands of posts that require verification are being shared every minute, and the English Wikipedia alone contains over 6 million articles
at the moment. In order to process many claims against comprehensive free text knowledge bases, we need to improve the efficiency of models while maintaining high accuracy. In §1.4 we expand on this matter and present reliable and efficient models for evidence retrieving, as well as for classification.
1.3 Robust Claim Classification

Automatically determining the relation between a given evidence sentence and a claim is considered a challenging task, even for advanced deep learning models. The related task of natural language inference (NLI), resolving whether a hypothesis is supported by a given premise, has been widely studied. Several recent studies have shown that NLI models are not following the same logical patterns as humans [85, 95, 179] and fail under adversarial and challenging cases [51, 101, 132, 192]. As we previously discuss (§1.2), the same is true for claim-evidence relation classifiers which can resort to relying on statistical cues that do not generalize.

A crucial step towards increasing the robustness of sentence-pair classifiers is creating challenging and realistic evaluation settings. Previous studies either use rules for generating or altering examples for testing sensitivity and robustness [179], or ask users to compose adversarial examples [132, 192, 273]. Following such approaches could lead to unrealistic inputs, or focus only on the specific models developed against. Instead, in this work, we pair claims with a set of at least two contrastive evidence sentences. Rather than being adversarial towards a specific model, this setting is challenging in nature and requires the classifier to be sensitive to evidence modifications. As a result, successful models have to attend to the evidence rather than following statistical cues in the claim.

In our work, we also focus on extracting word-level rationales that both improve models’ interpretability [155], and their alignment with human selections can indicate proper reasoning by the model.

In order to train sensitive classifiers, we need a large collection of contrastive evidence pairs. First, we create synthetic examples automatically with a novel fact-guided sentence modification pipeline. Then, we leverage real revisions to Wikipedia to create a large corpus with contrastive evidence.

**Automatic augmentation with contrastive evidence.** In order to automatically create contrastive pairs, we augment the FEVER dataset [272] by generating supporting evidence to false claims. As we lack supervision in the form of disagreeing claim-evidence pairs with
Sensitivity to dynamic changes in knowledge sources. As demonstrated in Figure 1-1, fact verification systems rely on external free text information resources such as Wikipedia. If these external sources are being actively maintained and updated with recent events, this can allow models to dynamically adjust their verdict by the most up-to-date information. As previously discussed, however, this is also dependent on the model’s sensitivity. In this work, we take advantage of the open-source logs of Wikipedia revisions to collect examples of factual updates in information resources. Figure 1-4 shows an example of such an update where due to an unexpected pandemic the 2020 Olympic games were postponed, but kept their original name. Shortly after the announcement, the respective Wikipedia page was revised to reflect this update. As detailed in Chapter 4, we leverage these revisions to create claims for which the two timestamps of Wikipedia provide contrastive evidence. Training classifiers with these examples significantly improves both their sensitivity and their robustness to adversarial inputs. For example, we find our model to succeed on the
example in Figure 1-4, compared to a FEVER-trained classifier and GPT-3 few-shot model that predict the claim as supported by the post-revision evidence.

**Moderating and updating free text information resources.** When expecting fact verification models to be consistent with external sources like Wikipedia, we assume that these resources are correct and up-to-date. Currently, the editing and moderating process is performed mostly manually, with rule-based bots automating simple tasks. Further automating some steps of the process can help in scaling up the reliability of articles, even on topics that are out of the main focus of most users.

For example, at the moment there are over eight thousand edits per hour in the English Wikipedia. Most of those edits include only stylistic changes. Automatically flagging revisions that modify an underlying fact can help topic moderators in tracking content changes and avoid vandalism. Building on our revision-based large resource (Chapter 4), we develop classifiers that reach high accuracy on this task. We also present text generation models to automatically distill the factual change in revisions, and to output edit suggestions to outdated content.

1.4 Efficient Evidence Retrieval and Claim Classification

In order to be used at scale, and against large information resources, fact verification systems should be efficient. Improving the efficiency of deep learning models, however, usually comes with an uncontrolled reduction in performance [65, 89, 112, 134, 236, 248, 266, inter alia]. In this work, we aim to reliably control this trade-off. We let the user specify an arbitrary tolerance level, and we improve the inference efficiency while marginally guaranteeing performance within the specified tolerance. To this end, we build on the conformal prediction framework (CP) [294] that provides performance guarantees, and we develop extensions that improve efficiency. In CP, we use an exchangeable set of examples \((X_1, Y_1), \ldots, (X_n, Y_n)\) to construct a valid prediction set \(C_{\epsilon,n}(X_{n+1})\) for a new example. By valid, we mean that \(P(Y_{n+1} \in C_{\epsilon,n}(X_{n+1})) \geq 1 - \epsilon\) where \(Y_{n+1}\) is the correct label for \(X_{n+1}\) and \(\epsilon\) is the user-specified tolerance level.

In this work, we focus on improving both statistical efficiency, the size of the prediction set; and computational efficiency, the required computational effort for obtaining this set. As we shall see, the two types of efficiencies are connected. On the one hand, utilizing computationally expensive models can lead to better label prediction scores than result in smaller prediction sets. On the other hand, in our multi-step systems, smaller prediction sets in an early step can reduce computational effort in later steps. We develop our models to preserve the CP validity guarantee while improving both types of efficiencies.

**Efficient evidence retrieval.** When retrieving evidence for a given claim, we consider a very large label space \(\mathcal{Y}\) of documents and sentences that can potentially provide sufficient evidence for the verdict. Therefore, constructing a small prediction set \(|C_{\epsilon,n}(X_{n+1})| \ll \mathcal{Y}\) is of crucial importance. In Chapter 5 we describe our cascaded approach for obtaining this set with reduced computational effort. We use a sequence of increasingly more powerful predictors, allowing us to early filter out many of the improbable candidates with simple and fast predictors.
**Efficient claim classification.** Following their remarkable performance in many domains, large Transformers \[287\] have become the most common models for fact verification. These multi-layered models recursively encode the input across many layers until a classifier is used on top of the deep representation. While successfully succeeding on many challenging examples, some claim-evidence pairs are trivial to classify and do not require the massive computational power of Transformer layers. Reliably identifying the exact effort required for every input, however, is challenging. To this end, we develop new confident early-exit classifiers that use meta-features (e.g., past predictions from lower layers) to predict whether an early classifier from an intermediate layer is consistent with the top layer. Then, we calibrate a decision rule that marginally guarantees consistency with the top layer. As we discuss in Chapter 6, to achieve this guarantee we use conformal prediction to obtain a superset of the inconsistent layers, and exit on the first layer that is not in that set. By doing so, we can substantially reduce the overall computational effort while providing the user with and tunable and reliable tolerance level $\epsilon$ for prediction consistency.
1.5 Contributions

The primary contributions of this work are:

- **Learning robust evidence-sensitive fact verification models:** We introduce new evaluation metrics and strategies for analyzing undesired model behaviors, and propose training techniques for addressing those. Our main methodological novelty is in creating contrastive evidence pairs for training, enforcing sensitivity of the model to evidence changes. We obtain these examples either with our introduced rationale-guided denoising task for factually modifying sentences, or by carefully curating examples from revisions to Wikipedia.

- **Improving the computational and statistical efficiency of retrieval systems:** We present extensions to the conformal prediction framework that enable smaller prediction sets with less computational effort, while providing accuracy marginal guarantees. Specifically, we develop a cascaded model with increasingly expensive models that can early prune a large amount of the candidates while reliably passing the least confident ones forward. We demonstrate the efficacy of our method on evidence retrieval, among other tasks.

- **Reducing the computational effort of multi-layered classifiers:** We propose an extension to multi-layered models such as Transformers, that allows confident adaptive computational utilization while preserving arbitrarily high consistency. Our method can use meta-features for learning confidence scores and calibrating a decision threshold for early-exiting the model. Importantly, by building on conformal calibration, we provide marginal guarantees on the consistency of the amortized model with the full model. Our experiments on claim-evidence classification and other tasks empirically validate our method.

Combined, we believe that these pieces will assist in advancing systems for misinformation prevention by automatically, efficiently, and reliably identifying suspicious claims and providing explanations.
1.6 Outline

The rest of this thesis is organized as follows:

- **Chapter 2** presents veracity-based misinformation detection benchmarks and analysis of stylometry-based defenses. It shows that such defenses can be useful against machine-human impersonations, but are limited in their ability to identify machine-generated false content.

- **Chapter 3** explores idiosyncrasies in fact verification datasets and how they cause undesired behaviors of models trained on them. It also introduces automatic debiasing techniques to alleviate the effect of these statistical cues on the model.

- **Chapter 4** introduces our new large-scale dataset based on Wikipedia revisions that significantly improves the robustness of classifiers. This dataset also supports several other tasks in the fact verification ecosystem that are formalized and discussed.

- **Chapter 5** presents our conformal prediction extensions for enabling efficient evidence retrieval with performance guarantees.

- **Chapter 6** presents our confident adaptive Transformer models for accelerating the inference of classifiers.

- **Chapter 7** summarizes the thesis with conclusions and future work.
Chapter 2

Limitations of Stylometry-based Detection Methods

Recent developments in neural language models (LMs) have raised concerns about their potential misuse for automatically spreading misinformation. In light of these concerns, several studies have proposed to detect machine-generated fake news by capturing their stylistic differences from human-written text. These approaches, broadly termed stylometry, have found success in source attribution and misinformation detection in human-written texts. However, in this chapter, we show that stylometry is limited against machine-generated misinformation. While humans speak differently when trying to deceive, LMs generate stylistically consistent text, regardless of underlying motive. Thus, though stylometry can successfully prevent impersonation by identifying text provenance, it fails to distinguish legitimate LM applications from those that introduce false information. We create two benchmarks demonstrating the stylistic similarity between malicious and legitimate uses of LMs, employed in auto-completion and editing-assistance settings. Our findings highlight the need for non-stylometry approaches in detecting machine-generated misinformation, and open up the discussion on the desired evaluation benchmarks.
2.1 Introduction

Many previous studies on stylometry—the extraction of stylistic features from written text—showed promising results on text classification. Two of stylometry’s common applications are: (1) Detecting the provenance of text (i.e. identifying the author) in order to prevent impersonations [282, 27, 3, 32, 187, 238]; (2) Detecting misinformation in text due to deception [63, 181, 195, 68, 2], fake news [221, 206], or other false or illegal content [49]. In the former, the classifier identifies language features that correlate with a specific person or group. The latter, misinformation detection, relies on idiosyncrasies of lies, i.e. style and language characteristics that are unique to text that is false or misleading.

Stylometry has recently gained attention as a potential answer to concerns that LMs could be used to mass-produce malicious text [292, 218], that (1) impersonates a human author’s text and/or (2) is fallacious and misleading. Indeed, stylometry-based approaches have shown promising results for defending against human-impersonating LMs [11, 317]. However, as applications of text generation such as text auto-completion [305, 110] and automatic question answering become widely used, labeling text as generated by a LM might not indicate anything at all about its trustworthiness. This motivates our core inquiry subject: can stylometry be used to distinguish malicious uses of LMs from legitimate ones?

We build the first benchmark for detection of LM-produced fake news that labels text as “real” or “fake” according to its veracity. Inspired by studies on deceitful behaviors in humans, showing that people try to diverge as little as possible from the truth when lying [178], we focus on automatic false modifications or additions to otherwise truthful news stories.

Our datasets contain articles produced by both malicious and responsible uses of language models, and the detector’s task is to identify the malicious ones. In one dataset, we produce text by prompting a LM to extend news articles with relevant claims. We simulate malicious user, who only accepts the LM’s suggestion if the claim is factually false, and a responsible user, who only accepts correct claims. The produced sentences

are short and concise statements, similarly to fake news and false claims as represented in human-generated datasets [299, 9]. In another dataset, we modify existing news articles to include false information by inverting article statements. In this case, the LM is used to automatically identify the most plausible edit locations. This is similar to (mis-)using an autocorrect tool that suggests local modifications.

We find that with the state-of-the-art stylometry-based classifier, even a single auto-generated sentence within a wall of human-written text is detectable with high accuracy, yet the truthfulness of a single sentence remains largely undecidable. Moreover, even a relatively weak LM can be used to produce statement inversions that the state-of-the-art stylometry-based model cannot detect. Thus, stylometry fails to distinguish malicious from responsible behaviors. This indicates that, unlike humans who expose stylistic cues when writing false content [195, 75, 175], LMs maintain consistency for both true and false content. Worse yet, while a provenance classifier can effectively detect a potentially malicious author or publication venue (given enough examples), it might not distinguish malicious from legitimate authors if they are both using the same LM to generate their text. In this regard, malicious text generated by a LM might actually be harder to detect than hand-crafted malicious text.

Our human evaluation tests show that humans are also fooled by machine-generated misinformation, but that access to external information sources can help. Therefore, we recommend future research on machine-generated misinformation to focus on non-stylometry strategies. Finally, we discuss what benchmarks are required for evaluating the performance of such detectors.

2.2 Background and Related work

2.2.1 Stylometry for Human-written Text.

The use of statistical methods for analyzing human-written documents has been studied extensively since the early days of the field. One common application is provenance detection. For example, [182] used word counts to predict the authorship of historical
documents, [282] extracted other stylistic features and applied a neural network to the same task. While these classifiers could be fooled by an aware writer that intentionally imitates other’s style [27], this approach was found useful for de-anonymizing cybercriminals in forums [3], identifying programmers [32] and more [187].

In a related line of study, stylometry was applied to, rather than detecting the specific person, identifying characteristics of the author, such as age and gender [87], political views [216], or IQ [1]. Recently, as detailed later below, stylometry was used to distinguish machine- from human-writers.

Another common application of stylometry is detecting human-written misinformation. [181] found specific words that are highly correlated with true and false statements. [195, 68] used a richer set of features such as POS tags frequencies and constituency structure to deceptive writing.

Following these observations and the increasing interest in fake news, recent studies applied stylometry on entire news articles [210], short news reports [206], fact and political statements [184, 221] and posts in social media [290]. The success of these studies is mostly attributed to stylistic changes in human language when lying or deceiving [23, 75]. In this work, we evaluate the viability of this approach on machine-generated text, where stylistic differences between truth and lie might be more subtle.

### 2.2.2 Machine-generated Text Detection.

The recent improvement in the quality of text generated by language models (LM) motivated several studies to examine their differences from human-written text. Detecting text’s provenance is similar to authorship attribution and, therefore, stylometry can be effective. [99] observe that a generative model’s likelihood scores can be used to distinguish machine-generated from human-generated text: generally, human-written texts exhibit lower likelihoods.

Indeed, the GLTR tool [79] shows the existence of distributional differences between human-written texts and machine-generated ones by visualizing the per-token probability according to a LM. [11] learn a dedicated provenance neural classifier. While their classifier achieves high in-domain accuracy, they find that it overfits the generated text distribution.
rather than detecting outliers from human texts, resulting in increased “human-ness” scores for random perturbations.

Nevertheless, an advantage of such neural approaches over more traditional stylometry methods is that, given enough data, the model learns hidden stylistic representations without the need to manually define any features.

Building on the above observation, [317] focus on fake news and create a Transformer-based LM [287] dubbed Grover and train it on a large news corpus. Grover also includes a “neural fake news detector”, a linear classifier on top of the hidden state of the last token of the examined article, fine-tuned to classify if the news text was machine-generated or not. The experiments in this work are based on the Grover-Mega classifier, fine-tuned for the target task (see section 2.3).

2.2.3 Fake News Detection Approaches Beyond Stylometry.

The other most extensively studied NLP-based approach for fake news detection is based on fact-checking. This approach has recently gained increasing attention thanks to several synthetic [272] and real-world datasets [96, 299, 214, 9]. In the rest of this thesis, we discuss multiple solutions for improving the performance of fact verification models, as well as the new datasets that we created. Below, we describe several other strategies for reducing misinformation were also recently studied. Each approach assumes access to a different set of features, such as propagation data, user’s history, feedback from other users and others. When available, this information can help to further improve the performance of fact verification

**Rumor Propagation.** Assessing the reliability of information propagated in social networks has recently been of high interest. To this end, methods that leverage features of the content’s propagation pattern [35, 93, 165, 188, 171, 170, 140, 166, 322], patterns of reaction to content [93, 166, 188, 171, 322, 140, 35], characteristics of the propagating users [93, 188, 166, 35, 212], and, finally, features of the textual content itself [93, 212] were applied. This line of work focuses on detecting fake news or rumors focused on settings where ample auxiliary information exists, beyond the potentially disputable text itself.
Non-textual Information. Another line of work for fake news detection utilizes non-textual information such as how content is propagated, by which users, its originating URL, and other metadata [35, 93, 322, 140, 166], as well as incorporating users’ explicit feedback, such as abuse reporting [280]. Social network platforms, ISPs, and even individual users can employ such methods to moderate content exposure. These approaches are beset by the challenges [256] of noisy and incomplete data, especially given the need to detect fake news early [166] (before propagation and user engagement patterns are fully formed).

2.3 Adversarial Setting

“Fake News”: Our Working Definition Our attackers focus on automatically introducing false information into otherwise trustworthy content. We call the resultant manipulated articles “fake news”. This definition matches the one of [327] and is in line with the disinformation focused view of [300]. Also, this is in line with how false claims are represented in many human-generated fake news datasets [299, 9]. Conversely, [317] focus on entirely fabricated articles, a different type of fake news where the goal is mostly to create “viral and persuasive” content.

Our choice of creating articles with only a limited number of false statements is aligned with the way humans tend to deceive or lie. Psychological studies [178] support the age-old notion that, when lying, “the best policy for the criminal is to tell the truth as nearly as possible”\(^2\). This helps preserve an honest self-image and, perhaps more importantly, reduces the chances of the lie being detected. For example, a study on the longest-surviving known fake Wikipedia article \(^3\) revealed that many of the presented facts were only slightly altered from other, true facts.

Attack and Defense Capabilities We adopt an adversarial setting similar to that of [317]. Our attacker wishes to generate fake text, that contains unverified or false claims, en masse, using a language model to automate the process. The attacker’s goal is to produce fake text that the verifier classifies as real. Our verifier is adaptive: it receives a limited set of

\(^2\)Raskolnikov, “Crime and Punishment” [61].
Fernandez defends Argentine grain export tax: $2 billion

President Cristina Fernandez on Tuesday defended an increase in export taxes on grains that has riled many farmers, and she called on them to respect the law in protecting her policies. <...> In a concession to her critics, Fernandez said the increase in taxes on exports of grains that she instituted in March by decree will be debated by Congress. But there is little likelihood that the Congress will order major changes, since her party controls both houses. But Hilda Duhalde, an opponent of Fernandez, was not persuaded. "It's true that they have a majority in both houses, but we have to put white on black and watch out for the small- and medium-sized producers, who are the ones suffering," she said. Argentina raised export taxes in March by more than 10 percent. Fernandez has said growers have benefited from rising world prices and the profits should be spread to help the poor.

Fernandez said she was open to dialogue, but a dialogue that does not countenance the blocking of roads or other disruptions to the lives of Argentina. "Democracy for the people, not the corporations," she said.

We attempt to answer: Who appealed for dialogue and respect? Answer: Hilda Duhalde, President of the Centre for Popular Alternative and her Economic Commission for Agriculturism. (Fake; President Cristina Fernandez)

We attempt to answer: What do farmers say higher taxes do? Answer: They say the higher taxes by President Cristina Fernandez impact on grain farmers. (Real)

---

Until Tuesday, the fight over Betsy DeVos’s nomination to be secretary of education revolved mostly around her support of contentious school choice programs. But her confirmation hearing that night opened her up to new criticism: <...> Ms. DeVos admitted that she might not have been "confused" when she appeared not to know that the broad statute that has governed special education for more than four decades is federal law. <...> She appeared blank on basic education terms. Asked how school performance should be assessed, she did not know the difference between growth, which measures how much students have learned over a given period, and proficiency, which measures how many students reach a targeted score. Ms. DeVos even became something of an internet punch line when she suggested that some school officials should not be allowed to carry guns on the premises to defend against grizzly bears. <...> But her statements on special education could make her vulnerable families of children with special needs are a vocal lobby, one that Republicans do not want to alienate. <...> Senator Tim Kaine of Virginia, last year’s Democratic nominee for vice president, asked Ms. DeVos whether schools that receive tax dollars should be required to meet the requirements of IDEA. "I think that is a matter that’s best left to the states," Ms. DeVos replied. Mr. Kaine came back: "So some states might be good to kids with disabilities, and other states might not be so good, and then what? People can just move around the country if they don’t like how their kids are being treated?" Ms. DeVos repeated, "I think that is an issue that’s best left to the states.” "It’s not federal law," an exasperated Mr. Kaine replied. "Do you think families should have recourse in the courts if schools don’t meet their needs?” she asked. "Senator, assure you that if confirmed I will be very sensitive to the needs of special needs students,” Ms. DeVos said. “It’s not about sensitivity, although that helps,” Ms. Hassan countered. <...>

---

SEOUL, South Korea — North Korea’s leader, Kim said on Sunday that his country was making final preparations to conduct its first test of an intercontinental ballistic missile — a bold statement less than a month before the inauguration of Donald J. Trump. Although North Korea has conducted five nuclear tests in the last decade and more than 20 ballistic missile tests in 2016 alone, and although it habitually threatens to attack the United States with nuclear weapons, the country has never fired an intercontinental ballistic missile, or ICBM. <...> In his speech, Mr. Kim did not comment on Mr. Trump’s election. Doubt still runs deep that North Korea has mastered all the technology needed to build a viable ICBM. But analysts in the region said the North’s launchings of rockets to put satellites into orbit in recent years showed that the country had cleared some key technological hurdles. After the North’s satellite launch in February, South Korean defense officials said the Unha rocket used in the launch, if successfully reconfigured as a missile, could fly more than 7, 400 miles with a warhead of 1, 10 to 3, 800 pounds. — far enough to reach most of the United States. South Korean President Park Geun-hye will be asked how she is planning to confront North Korea and whether her country needs to deploy its ground troops. It is also unlikely that she will deploy U.S. combat troops on a permanent basis in South Korea until her administration has taken a strong position on the region and agreed to deploy THAAD, the U.S. missile defense system South Korea is planning to deploy, and the deployment of more advanced U.S. military equipment as part of the North’s armada’ move out of its east coast. Mr. Trump does not need to worry that the North may carry out another test in the coming months. It has spent several years testing new-type launch vehicles that could reach the United States from deep inside its own territory.
Grover-Mega discriminator by querying its Web interface. We report human performance on some of the attacks.

2.4 Stylometry Fails to Detect Machine-Generated Misinformation

We create two datasets, simulating two different uses of LMs to automatically produce fake news. In the first, the extension scenario, an auto-completion text generator extends a news article. A responsible user of this generator verifies the correctness of the output (producing real text), whereas an attacker verifies incorrectness (producing fake text). In the second, the modification scenario, the attacker uses a human-written news article and performs subtle modifications to semantically modify statements. Specifically, we add and remove negations. This follows the intuition that, if we take care to add negations in a syntactically-correct fashion, the new sentence is a negative inversion of the original [232]. Yet, such changes are subtle enough to retain the original article’s style and distributional features. See below for the full details on the creation of the datasets.

Additionally, we used about 100 examples from each dataset to test human performance in detecting this form of misinformation. For the extension scenario, we assigned two subject volunteers with two different tasks. The first had the same task as Grover’s detector, and the second was allowed to use external sources to verify facts (which Grover cannot do). For the modification scenario, the participants had the same task as Grover but we highlighted negations to help them focus on the relevant parts.

(1) Creating the Extension Dataset We use the newsQA dataset [279] that contains Cable News Network (CNN) articles with corresponding questions and answers (segments copied verbatim from the text). Following [218] and [291], we embed the question in a template that is appended to the end of the input text:

We attempt to answer: <question>

Answer:
Then, the first sentence produced by Grover’s generator is used as a potential answer. We choose this template by examining common formats of questions in news articles and finding the one that empirically produces mostly reasonable answers by Grover’s generator. For example, given text about a suspect caught in Washington D.C., and the question “Where was the suspect stopped?”, Grover answered “2 blocks from the U.S. Capitol.”

We manually labeled the generated answers. We filtered out nonsensical ones (29%) and labeled the rest as real or fake texts by correctness. Answers containing at least one false statement (judging by the article and/or external knowledge) were labeled fake. We measured inter-annotator agreement on a subset of 100 examples labeled by two different annotators, and found it to be substantial (Cohen’s kappa score of $k = 0.78$).

Finally, we remove the sentence containing the answer from each article by removing the sentence with the highest TF-IDF-weighted word-count similarity with the question and answer (concatenated).

(2) Creating the Modifications Dataset We use articles from The New York Times [267]. We perform the modifications by removing and adding negations from statements: first, we randomly delete $m/2$ “not” or “no” occurrences from the text. Then, we automatically find statements to add $m/2$ negations to, using the probabilities of a GPT-2 Medium language model. We randomly sample 100 locations in the article and choose the ones with the maximal score, defined as the probability for either “not” or “no”, multiplied by the probability for the word following the negation. By both adding and removing, we keep the total number of negations unchanged, avoiding anomalous negations counts. Original NYT articles are labeled as real and modified ones as fake.

2.4.1 Results

The detector’s performance figures are given in Table [2.1]. Against QA extension, the detector reached 71% accuracy in distinguishing true vs. false extensions generated by Grover. While this is above a trivial majority baseline (51%), some of the success could be a side effect of the way that the dataset was constructed. Labeling instances with at least one false statement as fake makes these sentences generally longer, allowing a simple linear
Table 2.1: Results: Section 2.4. We report (macro) F1 score and overall accuracy, as well as precision and recall of the “fake” class. Zero-shot performance (not included) was very low in all cases.

classifier that only uses the extension’s length to reach 56%. When evaluated only on the short false answers (up to 10 words), the accuracy drops from 71% to only 62%. Ultimately, however, despite performing better than simple baselines, the detection in this setting is much worse than in distinguishing human from machine text (Section 2.5).

Comparing this with the results of the human evaluation is particularly instructive: the first subject, given the same task as the detector, attained an F1 score of 0.68, only marginally below the detector. The second participant, who was allowed to use external sources, attained an F1 of 0.84. We conclude that the stylometry-based classifier doesn’t perform much better than humans in detecting potential misinformation, and that verifying against other resources can drastically improve results.

For the automatic article modification attack, the classifier fails completely to detect two modifications \((m = 2)\), and is not much better even when \(m = 10\) statements are inverted. Even though the change includes negations that might introduce surprising factual claims, the language style remains too similar to detect it. Humans, evaluated with \(m = 2\), had an F1 of 0.74, with 0.91 and 0.59 recall for real and modified articles, respectively. Thus, while humans perform better than the model, many of the modified statements were interpreted as real. Manually examining these, we found that the added negation actually changed the meaning in 60% of the instances that were misclassified.

Overall, the results show that both attacks can produce articles containing misinformation that evade the stylometry detector and mislead humans.
<table>
<thead>
<tr>
<th>Provenance detection</th>
<th>precision</th>
<th>recall</th>
<th>F1</th>
<th>accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero-shot</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full article</td>
<td>0.84</td>
<td>0.98</td>
<td>0.90</td>
<td>90%</td>
</tr>
<tr>
<td>vanilla extension (g = 20%)</td>
<td>0.52</td>
<td>0.20</td>
<td>0.45</td>
<td>51%</td>
</tr>
<tr>
<td>vanilla extension (g = 1%)</td>
<td>0.07</td>
<td>0.01</td>
<td>0.28</td>
<td>37%</td>
</tr>
<tr>
<td>adaptive</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full article</td>
<td>0.93</td>
<td>0.94</td>
<td>0.94</td>
<td>94%</td>
</tr>
<tr>
<td>vanilla extension (g = 20%)</td>
<td>0.90</td>
<td>0.97</td>
<td>0.95</td>
<td>95%</td>
</tr>
<tr>
<td>vanilla extension (g = 1%)</td>
<td>0.91</td>
<td>0.95</td>
<td>0.94</td>
<td>95%</td>
</tr>
<tr>
<td>QA extension (machine vs. human)</td>
<td>0.82</td>
<td>0.86</td>
<td>0.83</td>
<td>83%</td>
</tr>
</tbody>
</table>

Table 2.2: Detection results on the attacks of Section 2.5 in a zero-shot setting and the adaptive setting (where the discriminator is fine-tuned to a specific attacker). We report (macro) F1 score and overall accuracy. Precision, and recall of the “fake” class are reported as well.

2.5 Stylometry Detects Machine-Human Impersonations

One might suspect that the low performance in Section 2.4 is due to limited capacity of our detector or the small fraction of generated text. We now show that the detector performs well on the provenance task and is sensitive to differences in small text portions.

(1) Fully-generated Articles We perform a first evaluation of Grover against texts generated by a different model of similar size, namely GPT-2 XL [218]. We include examples from the WebText test set [218], labeled real text, and the released unconditioned generations of the GPT-2 XL model, labeled fake text.

(2) Automatic Article Extension (Vanilla) In order to experiment with mixed human-machine articles, we build an attacker that automatically extends unfinished human-written articles. We simulate this attacker by querying a GPT-2 Medium model, conditioned on the first 500 words from an article. We include New York Times articles, truncated to match the length of the fake class, labeled as real text, and articles that were automatically extended, labeled fake text. We use different percentages (g) of machine-generated text, by stopping the generation after an appropriate number of sentences.
(3) Automatic QA-based Extension  In this experiment, we test whether the QA defense from Section 2.4 can perform better if we can assume that true texts are always human-written. To this end, we repeat the same setting but use the gold answers from the newsQA dataset for the “real text” class.

2.5.1 Results

Zero-shot Setting  As Table 2.2 shows, the zero-shot classifier is effective in detecting the fully generated articles of a different model, with a 0.9 F-1 score. The extended articles, containing a substantial amount of human-written text, are mostly classified as human-written in this setting. This is unsurprising considering the dataset of full articles the detector was originally trained on.

Adaptive Setting  After fine-tuning, Grover improves on the full articles and performs well against article extension generations, reaching 0.94 F1 score for articles with a single generated sentence ($g = 1\%$). This stylometry detector is thus highly effective in distinguishing human from machine.

In the QA setting, also containing a single machine-generated sentence, the detector performs worse by 0.11 F1 points. This might be due to using the (stronger) Grover-Mega generator for the QA setting, and because the appended template for QA was selected to empirically maximize human “reasonableness” scores. Still, this setting allows the stylometry detector to reach much higher performance than in the veracity-based setting (Section 2.4). This indicates that a restricted benchmark, that assumes no legitimate use of LMs, might not reflect the misinformation detection performance if the model is deployed in a world where LMs are used for both legitimate and malicious purposes.

2.6 Discussion

Advancements in LM technology and their various applications have introduced a new challenge: distinguishing truthful text from misinformation, when the text is generated or edited by a LM. Our experiments indicate that LM-generated falsified texts are very
similar in style to LM-generated texts containing true content. As a result, stylometry-based classifiers cannot identify auto-generated intentionally misleading content.

We conclude with the following recommendations:

(1) **Extending Veracity-based Benchmarks.** In order to better evaluate detectors against LM-generated misinformation, we recommend extending our benchmarks by creating other veracity-oriented datasets, that represent a wide range of LM applications, from whole-article generation to forms of hybrid writing and editing.

(2) **Improving Non-stylometry Methods.** Other detection approaches, as surveyed at the end of Section 2.2.3 are less affected by the use of LMs. Therefore, advancements in such methods can improve the detection of both human- and machine-generated misinformation. Notably, the fact-checking setting makes fewer assumptions on the available auxiliary information and can be applied even if the text was sent to the verifier through a private channel such as E-mail. However, since fact-checking requires advanced inference capabilities, incorporating non-textual information, when available, can yield better results.

**Conclusion** The potential use of LMs in creating fake news calls for a re-evaluation of current defense strategies. We examine the state-of-the-art stylometry model, and find it effective in preventing impersonation, but limited in detecting LM-generated misinformation. This new kind of misinformation could be created by the same model that is used by legitimate writers as a writing-assistance tool, hiding stylistic differences between falsified and truthful content. This motivates (1) constructing more instructive benchmarks for NLP-based approaches and improving non-stylistic methods, and (2) addressing a set of challenges that spans many disciplines beyond NLP, including social networks, information security, human-computer interaction, and others.
Chapter 3

Automatic Fact Verification: Identifying and Mitigating Undesired Biases

Fact verification requires validating a claim in the context of evidence. In this chapter we show, however, that in the popular FEVER dataset this might not necessarily be the case. Claim-only classifiers perform competitively with top evidence-aware models. We investigate the cause of this phenomenon, identifying strong cues for predicting labels solely based on the claim, without considering any evidence. We create an evaluation set that avoids those idiosyncrasies. The performance of FEVER-trained models significantly drops when evaluated on this test set. Therefore, we introduce a regularization method which alleviates the effect of bias in the training data, obtaining improvements on the newly created test set.

In the second part of this chapter, we present another approach for alleviating the claim-only bias by automatically modifying a refuting evidence to agree with a given claim. This is a challenging constrained generation task, as the output must be consistent with the new information and fit into the rest of the existing document. To this end, we propose a two-step solution: (1) We identify and remove the contradicting components in a target text for a given claim, using a neutralizing stance model; (2) We expand the remaining text to be consistent with the given claim, using a novel two-encoder sequence-to-sequence model with copy attention. Applied to a Wikipedia fact update dataset, our method successfully generates updated sentences for new claims, achieving the highest SARI score. Furthermore, we
demonstrate that generating synthetic data through such rewritten sentences can successfully augment the FEVER fact-checking training dataset, leading to a relative error reduction of 13%.

3.1 Introduction

Creating quality datasets is essential for expanding NLP functionalities to new tasks. Today, such datasets are often constructed using crowdsourcing mechanisms. Prior research has demonstrated that artifacts of this data collection method often introduce idiosyncratic biases that impact performance in unexpected ways [211, 95]. In this chapter, we explore this issue using the FEVER dataset, designed for fact verification [272].

The task of fact verification involves assessing claim validity in the context of evidence, which can either support, refute or contain not enough information. Figure 3-1(A) shows an example of a FEVER claim and evidence. While validity of some claims may be asserted in isolation (e.g. through common sense knowledge), contextual verification is key for a fact-checking task [4]. Datasets should ideally evaluate this ability. To assess whether this is the case for FEVER, we train a claim-only BERT [56] model that classifies each claim on its own, without associated evidence. The resulting system achieves 61.7%, far above the majority baseline (33.3%).

Our analysis of the data demonstrates that this unexpectedly high performance is due to idiosyncrasies of the dataset construction. For instance, in section 3.3 we show that the presence of negation phrasing highly correlates with the REFUTES label, independently of provided evidence.

To address this concern, we propose a mechanism for avoiding bias in the test set construction. We create a SYMMETRIC TEST SET where, for each claim-evidence pair, we manually generate a synthetic pair that holds the same relation (e.g. SUPPORTS or REFUTES) but expressing a different, contrary, fact. In addition, we ensure that in the new pair, each sentence satisfies the inverse relation with the original pair’s sentence. This process is illustrated in Figure 3-1, where an original REFUTES pair is extended with a synthetic REFUTES pair. The new evidence is constrained to support the original claim, and the new
(A) **ORIGINAL** pair from the FEVER dataset

**Claim:** Stanley Williams stayed in Cuba his whole life.

**Evidence:** Stanley [...] was part of the West Side Crips, a street gang which has its roots in South Central Los Angeles.

---

(B) **Manually GENERATED** pair

**Claim:** Stanley Williams moved from Cuba to California when he was 15 years old.

**Evidence:** Stanley [...] was born in Havana and didn’t leave the country until he died.

---

Figure 3-1: An illustration of a **REFUTES** claim-evidence pair from the FEVER dataset (A) that is used to generate a new pair (B). From the combination of the **ORIGINAL** and manually **GENERATED** pairs, we obtain a total of four pairs creating symmetry.

The original claim is supported by the original evidence. In this way, we arrive at three new pairs that complete the symmetry.

Determining veracity with the claim alone in this setting would be equivalent to a random guess. Unsurprisingly, the performance of FEVER-trained models drop significantly on this test set, despite having complete vocabulary overlap with the original dataset. For instance, the leading evidence-aware system in the FEVER Shared Task, the NSMN classifier by [191](https://github.com/easonnie/combine-FEVER-NSMN) achieves only 58.7% accuracy on the symmetric test set compared to 81.8% on the original dataset.

While this new test set highlights the aforementioned problem, other studies have shown that FEVER is not the only biased dataset [211][95]. A potential solution which may be applied also in other tasks is therefore to develop an algorithm that alleviates such bias in the training data. We introduce a new regularization procedure to downweigh the give-away phrases that cause the bias.

[1](https://github.com/easonnie/combine-FEVER-NSMN)
The contributions of this chapter are:

- We show that inherent bias in FEVER dataset interferes with context-based fact-checking.
- We introduce a method for constructing an evaluation set that explicitly tests a model’s ability to validate claims in context.
- We propose a new regularization mechanism that improves generalization in the presence of the aforementioned bias.
- We introduce a novel sentence modification pipeline with a rationale-based denoising task that automatically creates contrastive evidence pairs for data augmentation.

### 3.2 Related work

As we discussed in Chapter 1, the recent proliferation of misinformation online led to an increased interest in researching and developing mitigation tools. In Section 2.2.3 we covered a range of computer science approaches beyond NLP that could potentially be combined with NLP solutions if the relevant information is available. Chapter 2 also discussed stylometry-based solutions and their effectiveness against machine-human impersonations. Starting from this chapter, we focus on automatic fact verification for assessing the veracity of claims.

In this section, we provide an overview of previous related work in the field of fact verification, covering main datasets and methods. We also cover related work on annotation bias, text rewriting, and data augmentation since they are related to the topics discussed in this chapter. In the following chapters, we include a related work section that provides a more focused comparison with previous research directly related to the topic of each chapter.

#### 3.2.1 Fact Verification

Perhaps the first to introduce the task in the setting most similar to the one used today were Vlachos and Riedel in 2014 [289]. They motivate the need for automatic fact verification
due as it can reduce the overload for journalists that have to perform this time-consuming process manually. They discuss several strategies for performing fact-checking tasks and conclude that emulating the manual process is the most promising option. Other alternatives include learning from existing statements or identifying previously fact-checked statements, however, those options are limited in their potential coverage. The main advantage of imitating the manual process is the ability to break down the pipeline to well-defined NLP tasks: (1) extracting check-worthy claims, (2) constructing questions, (3) obtaining answers for the questions, and (4) reaching a verdict. In later work, the two intermediate steps are usually replaced with a single evidence retrieval step.

In 2018, Thorne and Vlachos [270] published a survey on automated fact-checking, discussing the different formulations settings and datasets available at that time. At the time, available datasets were relatively small. Also, the process of retrieving evidence was not well formalized and datasets either provide the relevant evidence or ignore this step. A common issue with these datasets is that claims mostly include statements that were examined and covered in multiple fact-checking and news websites. Therefore, it could be significantly easier for a system to retrieve and verify these claims, compared to the claims that we are interested to automatically verify as they were yet to be manually examined. An alternative use of previously fact-checked claims is to identify new appearances of these claims online and refer the reader to the respective articles [100]. For new claims, however, this solution is insufficient.

The analysis of Thorne and Vlachos [270] also motivated them to create a large dataset, named FEVER, with claims to be verified against context from English Wikipedia (see more details below) [272]. Building on Wikipedia, this allows a well-defined test-bed for models to look for relevant information for new claims. Compared to other approaches that evaluate claims against knowledge graphs or solely based on their writing style and origin, this strategy simulates the manual fact-checking process of searching for evidence online and reaching a verdict. As we discuss in this chapter, however, the claim creation process of FEVER led to idiosyncrasies that can be exploited by models, questioning whether FEVER-trained deep learning models indeed rely on external evidence. As a result, systems might not be sensitive to changes and developments in the real world, reflected by revisions.
to Wikipedia. In Chapter 4 we present our VitaminC dataset that is designed to alleviate this issue.

The FEVER dataset was used for two shared tasks [275, 276] and fueled the development of several deep learning models. In this benchmark, systems are evaluated on their ability to both retrieve relevant information for each claim, and to reach the correct final verdict on its veracity. Here, we focus on the recent advancements in the field that followed the release of FEVER and other large datasets.

Next, we discuss several of the recently shared datasets for fact verification. Then, we analyze proposed modeling solutions for this task. We also refer the reader to [16, 141] for other recent surveys on automatic fact verification.

### 3.2.2 Fact Verification Datasets

**Vlachos and Riedel [289]** collected 106 claims from the Channel 4 blog and the PolitiFact website. Following the website annotations, they used five labels: true, mostly true, half true, mostly false, and false. For some of the claims, they provide links to external sources where sufficient evidence could be found.

**Emergent [69]** is a collection of 300 claims based on rumors collected online from the Snopes website and Twitter hoax accounts. They also provide a set of 2,595 news articles to allow systems to evaluate the claims against. The claims were labeled by journalists by their veracity using three labels (true, false, and unverified).

**LIAR [299]** collected a larger set of 12,836 claims from the PolitiFact website. LIAR-PLUS [4] extend this collection by including the fact-checking article associated with each claim.

**RumorEval [55]** contains 325 Twitter threads about 8 events that lead to breaking news. They also provide Wikipedia articles about the related topics. To allow a realistic setting, they save the last article version before the time the respective tweet was posted.
Baly et al. [12] created an Arabic fact-checking corpus with 219 false claims and 203 true claims. Their false claims are collected from the VERIFY website that covers false claims about the war in Syria and the Middle East. For true claims, they searched the REUTERS website for news on the same topics as the false claims. This domain separation between the two classes could potentially lead models to rely on domain-specific features (for example, predicting any future REUTERS-originated claim as true). They use Google search to look for relevant evidence.

Perspectrum [12] used online debate websites to collect 907 controversial claims and used search results to find a collection of related documents that express different opinions on the examined claim. Acknowledging multiple evidence pieces of evidence with different relations towards a claim is similar to our approach with the contrastive evidence pairs in the symmetric and VitaminC datasets. However, our VitaminC dataset is significantly larger and relies on Wikipedia revisions that express both edit battles over controversial claims (e.g., whether spayed animals gain weight), as well as new information that updates with time.

Hanselowski et al. [97] collected 6,422 claims from the Snopes fact-checking website and asked annotators to mark specific sentences in the fact-checking articles that express a specific stance regarding the examined claim.

MultiFC [9] collected 36,534 claims from multiple fact-checking websites and opinion expressing platforms. They organized and standardized the meta-information about each claim such as the date, source, and mentioned entities. Each of the used sources has a different set of labels, sometimes due to different granularity levels (e.g., false vs. mostly-false) or due to a different type of annotation (e.g., “unclear” label). The benchmark unifies all these different kinds of claims with their original labels and uses a macro average across sources to obtain a single success score. For evidence pieces, they collected snippets of Google search results. However, they did not constrain the results by their published date. Therefore, some snippets include related fact-checking articles which are unrealistic to expect when verifying novel claims.
FEVER \cite{272} created a large set of 185,445 claims extracted by annotators from Wikipedia articles about popular topics. The resulting dataset pairs each false or true claim with evidence sentences from Wikipedia that either supports or refutes it. For some of the claims, no evidence was found, resulting in a “not enough information” label with any sentence. They also released a pre-processed dump of a subset of English Wikipedia to standardize the evidence retrieval training and evaluation. This dataset was used in shared tasks \cite{273, 276} and fueled the development of many deep learning models for evidence retrieval and fact verification. However, as we detail in this chapter, we find claims in FEVER to include statistical cues that result from its claim creation process. Our Symmetric and VitaminC datasets address this bias in the claim classification training and evaluation.

FEVER2.0 (adversarial) \cite{272} was created in the second shared task, where teams were asked to compose realistic-looking adversarial claims that fool the best performing FEVER models. This resulted in a collection of 1,174 claims.

Triggers \cite{8} used the universal adversarial triggers approach \cite{297} to generate 186 adversarial claims for FEVER models. Some of the claims for the NEI class are long, with many word repetitions, making them unnatural.

SciFact \cite{296} is a collection of 1,409 claims to be verified against a collection of 5,183 abstracts of scientific articles. The format is mostly similar to the FEVER dataset.

Climate-FEVER \cite{59} also used the FEVER framework and focused specifically on claims related to climate change.

HoVer \cite{122} focus on claims that require multi-hop reasoning and retrieving a sequence of evidence sentences from Wikipedia. The FEVER dataset already contained a few of such claims but most of the claims in FEVER can be verified against a single Wikipedia sentence. In HoVer, the authors relied on the multi-hop QA dataset HotpotQA \cite{314} and rewrote the questions to express claims that include facts from multiple documents. In total, this dataset contains 26,171 claims.
TabFact [44] focus on tables and create 118,275 claims to be verified against tables from Wikipedia. Many of the claims also require numerical reasoning.

SEM-TAB-FACTS [298] also use tables as evidence resources but collect them from scientific documents rather than Wikipedia. They released 4,506 manually written claims based on 981 tables and provided a larger set of auto-generated claims.

Thorne et al. [268] wanted to replace the synthetic claims of FEVER that sometimes non-realistic statements (e.g., “Lake Powell is a dog”) with more natural and common statements. They relied on the yes/no questions from the BoolQ dataset [50] to rewrite the questions as claims and paired them with the matching evidence from Wikipedia.

WikiFactCheck-English [240] took a different approach than most Wikipedia-based datasets. Instead of relying on evidence from Wikipedia, they collected the claims from statements in Wikipedia articles and used citations to external documents as evidence.

FEVEROUS [5] collected 87,026 claims written by annotators based on both sentences and tables from Wikipedia, requiring systems to support both unstructured and structured evidence sources.

VitaminC [242] and Symmetric [245] are described in this thesis in Chapter 4 and Section 3.4, respectively. Unlike other previous datasets, VitaminC contains evidence sentences from multiple time versions of Wikipedia and contains many claims that their truthfulness changes over time. Therefore, it enforces systems to be sensitive to changes in evidence resources. VitaminC also covers a wide range of domains including popular Wikipedia articles and COVID-19 related claims. Many of the claims require numerical reasoning.

3.2.3 Fact Verification Models

With the availability of large datasets, most recent fact verification systems use deep learning models. As deep learning models keep improving—thanks to advancements in model design,
Table 3.1: List of recently released datasets for fact verification. The bottom collections focus on evaluation with adversarial or challenging examples. Our VitaminC (Chapter 4) and Symmetric (Section 3.4) are also included.

The success of the BERT Transformer model [56] in many NLP tasks lead to its adoption by most participants of the second FEVER shared task [276]. Indeed, models using BERT instead of ESIM generally perform better on this task [see, e.g., 104, 261]. Later systems also use the RoBERTa model [167] that was pretrained longer than BERT. We find the Albert model [146] to provide improved robustness compared to most other Transformers, perhaps due to its shared parameters design.

In order to aggregate different evidence pieces and reach a single verdict for a claim, many systems use simple solutions such as averaging, majority vote, maximum confidence, or adding attention aggregation layers [169, 214]. Other systems concatenate all retrieved
evidence sentences before passing the whole text to the neural network. Recent work proposes to include graph components in the deep model to more directly model the inter-evidence semantic reasoning. Transformer-XH [321] add an extra-hop layer to the Transformer, allowing better performance compared to models that use a BERT encoder followed by a Graph model [168, 257, 323, 325]. However, recent studies reveal that stronger feedforward Transformers (without any graph components), paired with maximum confidence pooling match or outperforms the graph-based models [260, 283].

Several studies focus on explicitly modeling algebraic reasoning capabilities [44, 269] to fact-check numerical claims. However, in our experiments with VitaminC we observe that the large number of numerical examples in its training set allow Transformer models to learn algebraic reasoning. Specifically, we find VitaminC-trained models to succeed in classifying relations such as more/less, and before/after. This finding aligns with other work that reports Transformers to perform well on numerical reasoning tasks when trained with relevant data [82].

Finally, Lee et al. [152] suggest use language models directly on claims instead of retrieving external evidence. This approach is motivated by some evidence that Transformers capture some capacity of world knowledge in their pretraining process [209]. However, such models are lack interpretability by construction as no evidence is provided with the prediction. Also, in order to make the model sensitive to changes in the world, the whole language model needs to be frequently re-trained. Finally, as we discuss in Section 3.3, the success of such models could be attributed to bias in the evaluation dataset rather than truthfully capturing world knowledge.

### 3.2.4 Annotation Bias

Large scale datasets are fraught with give-away phrases [179, 193]. Crowd workers tend to adopt heuristics when creating examples, introducing bias in the dataset. In SNLI (Stanford Natural Language Inference) [24], entailment based solely on the hypothesis forms a very strong baseline [211, 95].

Similarly, as shown by [133], reading comprehension models that rely only on the question (or only on the passage referred to by the question) perform exceedingly well on several
popular datasets [302, 194, 106]. To address deficiencies in the SQuAD dataset [118], researchers have proposed approaches for augmenting the existing dataset [220]. In most cases, these augmentations are done manually, and involve constructing challenging examples for existing systems.

3.2.5 Text Rewriting

There have been several recent advancements in the field of text rewriting, including style transfer [255, 320, 43] and sentence fusion [14, 185, 83]. Unlike previous approaches, our sentence modification task addresses potential contradictions between two sources of information.

Our work is fairly related to the approach of [160], which separates the task of sentiment transfer into deleting strong markers of sentiment in a sentence and retrieving markers of the target label to generate a sentence with the opposite sentiment. In contrast to such work, where the requisite modification is along a fixed aspect (e.g. sentiment), in our setting, an arbitrary input sentence (the claim) dictates the space of desired modifications. Therefore, in order to succeed at our task, a system should understand the varying degree of polarization in the spans of the outdated sentence against the claim before modifying the sentence to be consistent with the claim.

3.2.6 Wikipedia Edits

Wikipedia edit history has been analyzed for insights into the kinds of modifications made [53, 312, 67]. The edit history has also been used for text generation tasks such as sentence compression and simplification [315], paraphrasing [176] and writing assistance [31]. In this work, we are interested in the novel task of automating the editing process with the guidance of a textual claim.

3.2.7 Data Augmentation

Methods for data augmentation are commonly used in computer vision [205]. There have been recent successes in NLP where augmentation techniques such as paraphrasing and
Table 3.2: Top 15 LMI-ranked bigrams in the train set of FEVER for REFUTES with its \( p(l|w) \). The corresponding figures for the development set are also provided.

| Bigram                | Train LMI-10^{-6} | \( p(l|w) \) | Development LMI-10^{-6} | \( p(l|w) \) |
|-----------------------|-------------------|--------------|--------------------------|--------------|
| did not               | 1478              | 0.83         | 1038                     | 0.90         |
| yet to                | 721               | 0.90         | 743                      | 0.96         |
| does not              | 680               | 0.78         | 243                      | 0.68         |
| refused to            | 638               | 0.87         | 679                      | 0.97         |
| failed to             | 613               | 0.88         | 220                      | 0.96         |
| only ever             | 526               | 0.86         | 350                      | 0.82         |
| incapable being       | 511               | 0.89         | 732                      | 0.96         |
| to be                 | 438               | 0.50         | 454                      | 0.65         |
| unable to             | 369               | 0.88         | 346                      | 0.95         |
| not have              | 352               | 0.78         | 211                      | 0.92         |
| to ever               | 312               | 0.84         | 349                      | 0.90         |
| refuses to            | 246               | 0.91         | 223                      | 0.89         |
| anything except       | 235               | 0.94         | 52                       | 1.00         |
| only one              | 232               | 0.88         | 138                      | 0.88         |
| ever been             | 209               | 0.83         | 171                      | 0.86         |

word replacement were applied to text classification \[139, 307\]. Adversarial examples in NLI with syntactic modifications can also be considered as methods of data augmentation \[114, 319\]. In this work, we create constrained modifications, based on a reference claim, to augment data for our task at hand. Our additions are specifically aimed towards reducing the bias in the training data, by having a false claim appear in both “Agrees” and “Disagrees” classes.

### 3.3 Bias in Fact Verification Models

In this section, we quantify the observed bias and explore the factors causing it.

**Claim-only Classification**  
Claim-only aware classifiers can significantly outperform all baselines described by \[272\].\footnote{We evaluate on the development set as the test set is hidden. Hyper-parameter fine-tuning is performed on a 20% split of the training set, which is finally joined to the remaining 80% for training the best setting.} BERT, for instance, attains an accuracy of 61.7%, which is just 8% behind NSMN. We hypothesize that these results are due to two factors: (1) idiosyncrasies distorting performance and (2) word embeddings revealing world knowledge.
Idiosyncrasies Distorting Performance  We investigate the correlation between phrases in the claims and the labels. In particular, we look at the n-gram distribution in the training set. We use Local Mutual Information (LMI) \cite{64} to capture high frequency n-grams that are highly correlated with a particular label, as opposed to \( p(l|w) \) that is biased towards low frequency n-grams. LMI between \( w \) and \( l \) is defined as follows:

\[
LMI(w, l) = p(w, l) \cdot \log \left( \frac{p(l|w)}{p(l)} \right),
\]

where \( p(l|w) \) is estimated by \( \frac{\text{count}(w,l)}{\text{count}(w)} \), \( p(l) \) by \( \frac{\text{count}(l)}{|D|} \), \( p(w, l) \) by \( \frac{\text{count}(w,l)}{|D|} \) and \( |D| \) is the number of occurrences of all n-grams in the dataset.

Table\ref{table:LMI} shows that the top LMI-ranked n-grams that are highly correlated with the \textsc{Refutes} class in the training set exhibit a similar correlation in the development set. Most of the n-grams express strong negations, which, in hindsight, is not surprising as these idiosyncrasies are induced by the way annotators altered the original claims to generate fake claims.

\begin{table}[h]
\centering
\begin{tabular}{|l|c|c||c|c|}
\hline
\textbf{Bigram} & \textbf{Train} & \textbf{Development} \\
 & \textbf{LMI-}10^{-6} & \textbf{\( p(l|w) \)} & \textbf{LMI-}10^{-6} & \textbf{\( p(l|w) \)} \\
\hline
united states & 271 & 0.64 & 268 & 0.44 \\
least one & 269 & 0.90 & 267 & 0.77 \\
at least & 256 & 0.72 & 163 & 0.48 \\
person who & 162 & 0.90 & 135 & 0.61 \\
stars actor & 143 & 0.86 & 111 & 0.71 \\
won award & 133 & 0.80 & 50 & 0.56 \\
american actor & 126 & 0.79 & 55 & 0.45 \\
starring movie & 100 & 0.88 & 34 & 0.80 \\
from united & 100 & 0.82 & 108 & 0.67 \\
from america & 96 & 0.89 & 108 & 0.74 \\
who acts & 96 & 0.97 & 74 & 1.00 \\
american actress & 82 & 0.81 & 54 & 0.67 \\
s birth & 77 & 0.88 & 54 & 0.67 \\
on television & 76 & 0.73 & 175 & 0.71 \\
released album & 75 & 0.71 & 8 & 0.50 \\
\hline
\end{tabular}
\caption{Top 15 LMI-ranked bigrams in the train set of FEVER for \textsc{Support}.}
\end{table}
Table 3.4: Top 15 LMI-ranked bigrams in the train set of FEVER for NOT ENOUGH INFO. * denotes computationally infeasible, as there are no occurrences.

| Bigram               | Train LMI $10^{-6}$ | Development LMI $10^{-6}$ | p(l|w) | p(l|w) |
|----------------------|---------------------|---------------------------|-------|-------|
| worked with          | 221                 | 129                       | 0.40  | 0.56  |
| s name               | 99                  | 106                       | 0.59  | 0.65  |
| award winning        | 98                  | 208                       | 0.52  | 0.79  |
| wyatt earp           | 96                  | *                         | 0.42  | 0.00  |
| finished college     | 86                  | 10                        | 0.68  | 0.42  |
| and it               | 86                  | 254                       | 0.42  | 0.73  |
| will ferrell         | 79                  | *                         | 0.46  | 0.00  |
| can be               | 75                  | 72                        | 0.35  | 0.48  |
| and he               | 74                  | 52                        | 0.38  | 0.59  |
| tim rice             | 70                  | *                         | 0.41  | 0.00  |
| on boat              | 69                  | 104                       | 0.93  | 0.92  |
| won lottery          | 62                  | 41                        | 0.97  | 1.00  |
| named on             | 61                  | 51                        | 0.84  | 1.00  |
| starred twilight     | 56                  | 21                        | 0.78  | 1.00  |
| to extract           | 55                  | *                         | 0.77  | 0.00  |

Static World Knowledge Word embeddings encompass world knowledge, which might augment the performance of claim-only classifiers. To factor out the contribution of world knowledge, we trained two versions of claim-only InferSent [211] on the FEVER claims: one with GloVe embeddings [203] and the other with random embeddings. The performance with random embeddings was 54.1%, compared to 57.3% with GloVe, which is still far above the majority baseline (33.3%). We conjecture that world knowledge is not the main reason for the success of the claim-only classifier.

3.4 Towards Unbiased Evaluation

Based on the analysis above, we conclude that an unbiased verification dataset should exclude ‘give-away’ phrases in one of its inputs and also not allow the system to solely rely on world knowledge. The dataset should enforce models to validate the claim with respect to the retrieved evidence. Particularly, the truth of some claims might change as the evidence varies over time.

3We use InferSent because BERT, being pretrained on Wikipedia, comprises world knowledge [265].
For example, the claim “Halep failed to ever win a Wimbledon title” was correct until July 19. A fact-checking system that retrieves information from Halep’s Wikipedia page should modify its answer to “false” after the update that includes information about her 2019 win.

Towards this goal, we create a Symmetric Test Set. For an original claim-evidence pair, we manually generate a synthetic pair that holds the same relation (i.e. SUPPORTS or REFUTES) while expressing a fact that contradicts the original sentences. Combining the ORIGINAL and GENERATED pairs, we obtain two new cross pairs that hold the inverse relations (see Figure 3-1). Examples of generated sentences are provided in Table 3.5.

This new test set completely eliminates the ability of models to rely on cues from claims. Considering the two labels of this test set\(^4\) the probability of a label given the existence of any n-gram in the claim or in the evidence is \(p(l|w) = 0.5\), by construction.

Also, as the example in Figure 3-1 demonstrates, in order to perform well on this dataset, a fact verification classifier may still take advantage of world knowledge (e.g. geographical locations), but reasoning should only be with respect to the context.

Symmetric Test Set  The full Symmetric Test Set consists of 956 claim-evidence pairs, created following the procedure described in section 3.4. The new pairs originated from 99 SUPPORTS and 140 REFUTES pairs that were randomly picked from the cases which NSMN correctly predicts\(^5\). After its generation, we asked two subjects to annotate randomly sampled 285 claim-evidence pairs (i.e. 30% of the total pairs in Symmetric Test Set) with one label among SUPPORTS, REFUTES or NOT ENOUGH INFO, flagging non-grammatical cases. They agreed with the dataset labels in 94% of cases, attaining a Cohen \(\kappa\) of 0.88 [52]. Typos and small grammatical errors were reported in 2% of the cases. Given the small size of this dataset, we only use it as a test set.

\(^4\)NOT ENOUGH INFO cases are easy to generate so we focus on the two other labels.
\(^5\)Due to our focus on the performance drop with respect to the newly generated pairs rather than on the intention of multiplying the difficulties for the top performing model.
<table>
<thead>
<tr>
<th>Source</th>
<th>Claim</th>
<th>Evidence</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEVER</td>
<td>Tim Roth is an English actor.</td>
<td>Timothy Simon Roth (born 14 May 1961) is an English actor and director.</td>
<td>SUP</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Tim Roth is an American actor.</td>
<td>Timothy Simon Roth (born 14 May 1961) is an American actor and director.</td>
<td>SUP</td>
</tr>
<tr>
<td>FEVER</td>
<td>Aristotle spent time in Athens.</td>
<td>At seventeen or eighteen years of age, he joined Plato’s Academy in Athens and remained there until the age of thirty-seven (c. 347 BC).</td>
<td>SUP</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Aristotle did not visit Athens.</td>
<td>At seventeen or eighteen years of age, he missed the opportunity to join Plato’s Academy in Athens and never visited the place.</td>
<td>SUP</td>
</tr>
<tr>
<td>FEVER</td>
<td>Telemundo is a English-language television network.</td>
<td>Telemundo (telemundo) is an American Spanish-language terrestrial television network owned by Comcast through the NBCUniversal division NBCUniversal Telemundo Enterprises.</td>
<td>REF</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Telemundo is a Spanish-language television network.</td>
<td>Telemundo (telemundo) is an American English-language terrestrial television network owned by Comcast through the NBCUniversal division NBCUniversal Telemundo Enterprises.</td>
<td>REF</td>
</tr>
<tr>
<td>FEVER</td>
<td>Magic Johnson did not play for the Lakers.</td>
<td>He played point guard for the Lakers for 13 seasons.</td>
<td>REF</td>
</tr>
<tr>
<td>Symmetric</td>
<td>Magic Johnson played for the Lakers.</td>
<td>He played for the Giants and no other team.</td>
<td>REF</td>
</tr>
</tbody>
</table>

Table 3.5: Examples of pairs from the Symmetric Dataset and, above each line, the related pair from the FEVER dataset. Each claim-evidence pair holds the relation described in the right column. Crossing the Symmetric sentences with the FEVER ones creates two additional claim-evidence pairs with an opposite label (see Figure 3.1).

### 3.5 Debiasing Statistical Cues with Importance Weighting

Creating a large dataset for training is discussed in Chapter 4. Section 3.6 presents a method for automatically generating synthetic symmetric pairs. Here, we propose an algorithmic solution to alleviate the bias introduced by ‘give-away’ n-grams present in the claims. We re-weight the instances in the dataset to flatten the correlation of claim n-grams with respect to the labels. Specifically, for ‘give-away’ phrases of a particular label, we increase the importance of claims with different labels containing those phrases.

We assign an additional (positive) balancing weight $\alpha^{(i)}$ to each training example $\{x^{(i)}, y^{(i)}\}$, determined by the words in the claim.
Bias in the Re-Weighted Dataset  For each n-gram \( w_j \) in the vocabulary \( V \) of the claims, we define the bias towards class \( c \) to be of the form:

\[
\begin{align*}
    b^c_j & = \frac{\sum_{i=1}^n I_{[w_j(i)]}(1 + \alpha^{(i)}) I_{[y^{(i)}=c]}}{\sum_{i=1}^n I_{[w_j(i)]}(1 + \alpha^{(i)})},
\end{align*}
\]

where \( I_{[w_j(i)]} \) and \( I_{[y^{(i)}=c]} \) are the indicators for \( w_j \) being present in the claim from \( x^{(i)} \) and label \( y^{(i)} \) being of class \( c \), respectively.

Optimization of the Overall Bias  Finding the \( \alpha \) values which minimize the bias leads us to solving the following objective:

\[
\begin{align*}
    \min \left( \sum_{j=1}^{|V|} \max_c (b^c_j) + \lambda \|\vec{\alpha}\|_2 \right),
\end{align*}
\]

Re-Weighted Training Objective  We calculate the \( \alpha \) values separately from the model optimization, as a pre-processing step, by optimizing Equation 3.3 Using these values, the training objective is re-weighted from the standard \( \sum_{i=1}^n L(x^{(i)}, y^{(i)}) \) to

\[
\begin{align*}
    \sum_{i=1}^n (1 + \alpha^{(i)}) L(x^{(i)}, y^{(i)}).
\end{align*}
\]

This re-weighting is independent of the model architecture and can be easily added to any objective, similar to [120] where they learn instance weights to address labeling bias in datasets.

3.5.1 Experiments  We use the Symmetric Test Set to (1) investigate whether top performing sequence classification models trained on the FEVER dataset are actually verifying claims in the context of evidence; and (2) measure the impact of the re-weighting method described in section 3.5 over a classifier.

To achieve the first goal, we use three classifiers. The first is a pre-trained, current FEVER state-of-the-art classifier, NSMN [191] which is a variation of the ESIM [41] model,
with a number of additional features, such as contextual word embeddings [207]. In addition, we train our own ESIM model with GloVe embeddings, using the available code from [78]. The third is a BERT classifier that we fine-tune for 3 epochs to classify the relation based on the concatenation of the claim and evidence (with a delimiter token). To measure the impact of our regularization method, we also train the ESIM and BERT models with the re-weighting method.

Table 3.6 summarizes the performance of the three models on the SUPPORTS and REFUTES pairs from the FEVER DEV set and on the created SYMMETRIC TEST SET pairs. All models perform relatively well on FEVER DEV but achieve less than 60% accuracy on the synthetic ones. We conjecture that the drop in performance is due to training data bias that is also observed in the development set (see section 3.3) but not in the generated symmetric cases.

Our re-weighting method (section 3.5) helps to reduce the bias in the claims. In Table 3.7, we revisit the give-away bigrams from Table 3.2. Applying the weights obtained by optimizing Equation 3.3, the weighted distribution of these phrases being associated with a specific label in the training set is now roughly uniform.

The re-weighting method increases the accuracy of the ESIM and BERT models by an absolute 3.4% and 3.3% respectively. One can notice that this improvement comes at a cost in the accuracy over the FEVER DEV pairs. Again, this can be explained by the bias in the training data that translates to the development set, allowing FEVER-trained models to leverage it. Applying the regularization method, using the same training data, helps to train a more robust model that performs better on our test set, where verification in context is a

<table>
<thead>
<tr>
<th>Model</th>
<th>FEVER DEV BASE</th>
<th>FEVER DEV R.W</th>
<th>GENERATED BASE</th>
<th>GENERATED R.W</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSMN</td>
<td>81.8</td>
<td>-</td>
<td>58.7</td>
<td>-</td>
</tr>
<tr>
<td>ESIM</td>
<td>80.8</td>
<td>76.0</td>
<td>55.9</td>
<td>59.3</td>
</tr>
<tr>
<td>BERT</td>
<td><strong>86.2</strong></td>
<td><strong>84.6</strong></td>
<td><strong>58.3</strong></td>
<td><strong>61.6</strong></td>
</tr>
</tbody>
</table>

Table 3.6: Classifiers’ accuracy on the SUPPORTS and REFUTES cases from the FEVER DEV set and on the GENERATED pairs for the SYMMETRIC TEST SET in the setting of without (BASE) and with (R.W) re-weight.

[https://github.com/huggingface/pytorch-pretrained-BERT]
| Bigram                | R.W LMI·10^{-6} | R.W $p(l|w)$ |
|-----------------------|-----------------|--------------|
| did not               | 144             | 0.35         |
| yet to                | 30              | 0.33         |
| does not             | 67              | 0.35         |
| refused to           | 55              | 0.35         |
| failed to            | 31              | 0.33         |
| only ever            | 9               | 0.31         |
| incapable being      | 32              | 0.33         |
| to be                | 8               | 0.30         |
| unable to            | 10              | 0.32         |
| not have             | 41              | 0.35         |

Table 3.7: Re-weighted LMI and conditional probabilities ($l = \text{REFUTES}$) for the bigrams from Table 3.2. The weights were obtained following the optimization of Equation 3.3 on the training set which contains three labels.

3.6 Automatic Data Augmentation with Fact-guided Evidence Modification

Online text resources like Wikipedia contain millions of articles that must be continually updated. Some updates involve expansions of existing articles, while others modify the content. In this work, we are interested in the latter scenario where the modification contradicts the current articles. Such changes are common in online sources and often cover a broad spectrum of subjects ranging from the changing of dates for events to modifications of the relationship between entities. In these cases, simple solutions like negating the original text or concatenating it with the new information would not apply. In this work, our goal is to automate these updates. Specifically, given a claim and an outdated sentence from an article, we rewrite the sentence to be consistent with the given claim while preserving non-contradicting content.

Consider the Wikipedia update scenario depicted in Figure 3-2. The claim, informing that 23 of 43 minority stakeholdings are significant, contradicts the old information in the Wikipedia sentence, requiring modification. Directly learning a model for this task
would demand supervision, i.e. demonstrated updates with the corresponding claims. For Wikipedia, however, the underlying claims which drive the changes are not easily accessible. Therefore, we need to utilize other available sources of supervision.

In order to make the corresponding update, we develop a two step solution: (1) Identify and remove the contradicting segments of the text (in this case, *28 of their 42 minority stakeholdings*); (2) Rewrite the residual sentence to include the updated information (e.g. fraction of significant stakeholdings) while also preserving the rest of the content.

For the first step, we utilize a neutrality stance classifier as indirect supervision to identify the polarizing spans in the target sentence. We consider a sentence span as polarizing if its absence increases the neutrality of the claim-sentence pair. To identify and mask such sentence spans, we introduce an interpretability-inspired [155] neural architecture to effectively explore the space of possible spans. We formulate our objective in a way that the masking is minimal, thus preserving the context of the sentence.

For the second step, we introduce a novel, two-encoder decoder architecture, where two encoders fuse the claim and the residual sentence with a more refined control over their interaction.

We apply our method to two tasks: automatic fact-guided modifications and data augmentation for fact-checking. On the first task, our method is able to generate corrected Wikipedia sentences guided by unstructured textual claims. Evaluation on Wikipedia mod-
fications demonstrates that our model’s outputs were the most successful in making the requisite updates, compared to strong baselines. On the FEVER fact-checking dataset, our model is able to successfully generate new claim-evidence supporting pairs, starting with claim-evidence refuting pairs — intended to reduce the bias in the dataset. Using these outputs to augment the dataset, we attain a 13% decrease in relative error on an unbiased evaluation set.

3.6.1 Model

Problem Statement We assume access to a corpus \( \mathcal{D} \) of claims and knowledge-book sentences. Specifically, \( \mathcal{D} = \{\{C_1, ..., C_n\}, \{S_1, ..., S_m\}\} \), where \( C \) is a short factual sentence (claim), and \( S \) is a sentence from Wikipedia. Each pair of claim and Wikipedia sentence has a relation \( \text{rel}(S, C) \), of either agree (Agr), disagree (Dis) or neutral (N). In this corpus, a Wikipedia sentence \( S \) is defined as outdated with respect to \( C \) if \( \text{rel}(S, C) = \text{Dis} \) and updated if \( \text{rel}(S, C) = \text{Agr} \). The neutral relation holds for pairs in which the sentence doesn’t contain specific information about the claim.

Our goal is to automatically update a given sentence \( S \), which is outdated with respect to a \( C \). Specifically, given a claim and a pair for which \( \text{rel}(S, C) = \text{Dis} \), our objective is to apply minimal modifications to \( S \) such that the relation of the modified sentence \( S^+ \) will be: \( \text{rel}(S^+, C) = \text{Agr} \). In addition, \( S^+ \) should be structurally similar to \( S \).

Framework Currently, to the best of our knowledge, there is no large dataset for fact-guided modifications. Instead, we utilize a large dataset with pairs of claims and sentences that are labeled to be consistent, inconsistent or neutral. In order to compensate the lack of direct supervision, we develop a two-step solution. First, using a pretrained fact-checking classifier for indirect supervision, we identify the polarizing spans of the outdated sentence and mask them to get a \( S^0 \) such that \( \text{rel}(S^0, C) = \text{N} \). Then, we fuse this pair to generate the updated sentence which is consistent with the claim. This is done with a sequence-to-sequence model trained with consistent pairs through an auto-encoder style objective. The two steps are trained independently to simplify optimization (see Figure 3-4).
Masker: Eliminate Polarizing Spans

In this section we describe the module to identify the polarizing spans within a Wikipedia sentence. Masking these spans ensures that the residual sentence-claim pairs attain a neutral relation. Here, neutrality is determined by a classifier trained on claim and Wikipedia sentence pairs as described below. Using this classifier, the masking module is trained to identify the polarizing spans by maximizing the neutrality of the residual-sentence and claim pairs. In order to preserve the context of the original sentence, we include optimization constraints to ensure minimal deletions. This approach is similar to neural rationale-based models [155], where a module tries to identify the spans of the input that justify the model’s prediction.

Neutrality Masker  Given a knowledge-book sentence \((S)\) and a claim \((C)\), the masker’s goal is to create \(S^0\) such that \(\text{rel}(S^0, C) = N\). For the original sentence with \(l\) tokens, \(S = \{x_i\}_{i=1}^l\), the output is a mask \(m \in [0, 1]^l\). The neutral sentence \(S^0\) is constructed as:

\[
S^0_i = \begin{cases} 
  x_i, & \text{if } m_i = 0 \\
  *, & \text{otherwise}
\end{cases}
\]

Figure 3-3: Illustrating the flow of the masker module.
where ⋆ is a special token. The details of the masker architecture are stated below and depicted in Figure 3-3.

**Encoding** We encode \( S \) with a sequence encoder to get \( e_i = f(x; w_f)_i \). Since the neutrality of the sentence needs to be measured with respect to a claim, we also encode the claim and enhance \( S \)’s representations with that of \( C \) using attention mechanism. Formally, we compute

\[
z_i = e_i + \sum_{j=1}^{n} a_{i,j} \cdot c_j,
\]

where \( c_j \) are the encoded representations of the claim and \( a_{i,j} \) are the parameterized bilinear attention \([136]\) weights computed by:

\[
a_{i,j} = \text{softmax}_j(\text{atten}(e_i, c_j)),
\]

\[
\text{atten}(e_i, c_j) = e_i W c_j^T + b.
\]

Finally, the aggregated representations are used as input to a sequence encoder \( g(\cdot; w_g) \).

**Masking** The encoded sentence is used to predict a per token masking probability:

\[
p(m_i = 1) = \sigma(g(z; w_g)_i).
\]

Then, the mask is applied to achieve the residual sentence:

\[
S^\emptyset = S \odot (1 - m),
\]

where \( \odot \) denotes element-wise multiplication. During training, we perform soft deletions over the token embeddings and add the out-of-vocabulary embedding in place. During inference, the values of \( m \) are rounded to create a discrete mask.

\[\text{[136]}\text{The special token is treated as an out-of-vocabulary token for the following models.}\]
**Training** A pretrained fact-checking neutrality classifier’s prediction \( \text{rel}(S, C) \) is used to guide the training of the masker. In order to encourage maximal retention of the context, we utilize a regularization term to minimize the fraction of the masked words. The joint objective is to minimize:

\[
\mathcal{L}(S, C, m) = -\log \left( p(\text{rel}(S^0, C) = N) \right) + \frac{\lambda}{l} \sum_{i=1}^{l} m_i. \tag{3.11}
\]

**Fact-checking Neutrality Classifier** Our fact-checking classifier is pretrained on agreeing and disagreeing \( (S, C) \) pairs from \( D \), in addition to neutral examples constructed through negative sampling. For each claim we construct a neutral pair by sampling a random sentence from the same paragraph of the polarizing sentence, making it contextually close to the claim, but unlikely to polarize it. We pretrain the classifier on these examples and fix its parameters during the training of the masker.

**Optional Syntactic Regularization** Currently the model is trained with distant supervision, so, we pre-compute a valid neutrality mask as additional signal, when possible. To this end, we parse the original sentences using a constituency parser and iterate over continuous syntactic phrases by increasing length. For each sentence, the shortest successful neutrality mask (if any) is selected as a target mask. In the event of successfully finding such a mask, the masking module is regularized to emulate the target mask by adding the following term to Equation (3.11):

\[
\frac{1}{l} \| m - m' \|^2, \tag{3.12}
\]

where \( m' \) is the target mask.

Empirically, we find that the model can perform well even without this regularization, but it can help to stabilize the training. Additional details and analysis are available in the appendix.

**Two-encoder Pointer Generator: Constructing a Fact-updated Sentence**

In this section we describe our method to generate an output which agrees with the claim. If the earlier masking step is done perfectly, the merging boils down to a simple fusion task.

\[\text{If there are several successful masks of the same length, we use the one with the highest neutrality score.}\]
Figure 3-4: A summary of our pipeline. Given a sentence that is inconsistent with a claim, a masker is applied to mask out the contradicting parts from the original text while preserving the rest of the content. Then, the residual neutral text and claim are fused to create an updated text that is consistent with the claim. The Masker and the Two-Encoder Generator are trained separately.

However, in certain cases, especially ones with a strong contradiction, our minimal deletion constraint might leave us with some residual contradictions in \( \mathcal{S}^0 \). Thus, we develop a model which can control the amount of information to consider from either input.

We extend the pointer-generator model of [249] to enable multiple encoders. While sequence-to-sequence models support the encoding of multiple sentences by simply concatenating them, our use of a per input encoder allows the decoder to better control the use of each source. This is especially of interest to our task, where the context of the claim must be translated to the output while ignoring contradicting spans from the outdated Wikipedia sentence.

Next, we describe the details of our generator’s architecture. Here, we use one encoder for the outdated sentence and one encoder for the claim. In order to reduce the size of the model, we share the parameters of the two encoders. The model can be similarly extended to any number of encoders.

**Encoding** At each time step \( t \), the decoder output \( h^t \), is a function of a weighted combination of the two encoders’ context representations \( r^t \), the decoder output in the previous step \( h^{t-1} \) and the representation of the word output at the end of the previous step \( \text{emb}(y^{t-1}) \):

\[
h^t = \text{RNN}([r^t, \text{emb}(y^{t-1})], h^{t-1}).
\]  

(3.13)

As the decoder should decide at each time step which encoder to attend more, we introduce an encoder weight \( \alpha \). The shared encoder context representation \( r^t \) is based on their individual representations \( r^t_1 \) and \( r^t_2 \):
The context representation \( r^t_i \) \((i \in \{1, 2\})\) is the attention score over the encoder representation \( r_i \) for a particular decoder state \( h^{t-1} \):

\[
z^t_j = u^T \tanh(r_{i,j} + h^{t-1}),
\]

\[
a^t_i = \text{softmax}(z^t),
\]

\[
r^t_i = \sum_j a^t_{i,j} r_{i,j}.
\]

### Decoding

Following standard copy mechanism, predicting the next word \( y^t \), involves deciding whether to generate \( (p_{gen}) \) or copy, based on the decoder input \( x^t = [r^t, \text{emb}(y^{t-1})] \), the decoder state \( h^t \) and context vector \( r^t \):

\[
p_{gen} = \sigma(v_x^T x^t + v_h^T h^t + v_r^T r^t).
\]

In case of copying, we need an additional gating mechanism to select between the two sources:

\[
p_{enc1} = \sigma(u_x^T x^t + u_h^T h^t + u_r^T r^t).
\]

When generating a new word, the probability over words from the vocabulary is computed by:

\[
P_{vocab} = \text{softmax}(V^T[h^t, r^t]).
\]

The final output of the decoder at each time step is then computed by:

\[
P(w) = p_{gen}P_{vocab}(w) +
(1 - p_{gen})(p_{enc1}) \sum_{j : w_j = w} a^t_{1,j} +
(1 - p_{gen})(1 - p_{enc1}) \sum_{j : w_j = w} a^t_{2,j},
\]

\[
y^t = \arg\max_w P(w).
\]
where $a^t$ are the input sequence attention scores from Equation 11.

**Training** Since we have no training data for claim guided sentence updates, we train the generator module to reconstruct a sentence $S$ to be consistent with an agreeing claim $C$. The training input is the residual up-to-date neutral sentence $S^0$ and the guiding claim $C$.

During inference, we utilize only guiding claims and residual outdated sentences $S^0$ to create $S^+$. While generating the updated sentences $S^+$, we would like to preserve as much context as possible from the contradicting sentence, while ensuring the correct relation with the claim. Therefore, for each case, if the later goal is not achieved, we gradually increase the focus on the claim by increasing $\alpha$ and $p_{enc1}$ values until the output $S^+$ satisfies $\text{rel}(S^+, C) = \text{Agr}$, or until a predefined maximum weight.

### 3.6.2 Experimental Setting

We evaluate our model on two tasks: (1) Automatic fact updates of Wikipedia sentences, where we update outdated wikipedia sentences using guiding fact claims; and (2) Generation of synthetic claim-evidence pairs to augment an existing biased fact-checking dataset in order to improve the performance of trained classifiers on an unbiased dataset.

**Datasets**

**Training Data from FEVER** We use FEVER [272], the largest available Wikipedia based fact-checking dataset to train our models for both of our tasks. This dataset contains claim-evidence pairs where the claim is a short factual sentence and the evidence is a relevant sentence retrieved from Wikipedia. We use these pairs as our claim-sentence samples and use the “refutes”, “not enough information”, “supports” labels of that dataset as our Dis, N, Agr relations, respectively.

**Evaluation Data for Automatic Fact Updates** We evaluate the automatic fact updates task on an evaluation set based on part of the symmetric dataset from [246] and the fact-based cases from a Wikipedia updates dataset [312]. For the symmetric dataset, we use the modified Wikipedia sentences with their guiding claims to generate the true Wikipedia
sentence. For the cases from the updates dataset, we have human annotators write a guiding claim for each update and use it, together with the outdated sentence, to generate the updated Wikipedia sentence. Overall we have a total of 201 tuples of fact update claims, outdated sentences and updated sentences.

**Evaluation Data for Augmentation**  To measure the proficiency of our generated outputs for data augmentation, we use the unbiased FEVER-based evaluation set of [246]. As shown by [246], the claims in the FEVER dataset contain give-away phrases that can make FEVER-trained models overly rely on them, resulting in decreased performance when evaluated on unbiased datasets.

The classifiers trained on our augmented dataset are evaluated on the unbiased symmetric dataset of [246]. This dataset (version 0.2) contains 531 claim-evidence pairs for validation and 534 claim-evidence pairs for testing.

In addition, we extend the symmetric test set by creating additional FEVER-based pairs. We hired crowd-workers on Amazon Mechanical Turk and asked them to simulate the process of generating synthetic training pairs. Specifically, for a “refutes” claim-evidence FEVER pair, the workers were asked to generate a modified supporting evidence while preserving as much information as possible from the original evidence. We collected responses of workers for 500 refuting pairs from the FEVER training set. This process extends the symmetric test set (+TURK) by 1000 cases — 500 “refutes” pairs, and corresponding 500 “supports” pairs generated by turkers.

**Implementation Details**

**Masker**  We implemented the masker using the AllenNLP framework [78]. For a neutrality classifier, we train an ESIM model [41] to classify a relation of Agr, Dis or N. To train this classifier, we use the Agr and Dis pairs from the FEVER dataset and for each claim we add a neutral sentence which is sampled from the sentences in the same document as the polarizing one. The classifier and masker are trained with GloVe [203] word embeddings. We use BiLSTM [235] encoders with hidden dimensions of 100 and share the parameters of the claim and original sentence encoders. The model is trained for up to 100 epochs with a
<table>
<thead>
<tr>
<th>MODEL</th>
<th>Automatic Evaluation</th>
<th>Human’s Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SARI</td>
<td>KEEP</td>
</tr>
<tr>
<td>Fact updates:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split-no-Copy</td>
<td>15.1</td>
<td>36.9</td>
</tr>
<tr>
<td>Paraphrase</td>
<td>15.9</td>
<td>18.7</td>
</tr>
<tr>
<td>Claim Ext.</td>
<td>12.9</td>
<td>22.6</td>
</tr>
<tr>
<td>M. Concat</td>
<td>26.5</td>
<td>61.7</td>
</tr>
<tr>
<td>Ours</td>
<td>31.5</td>
<td>45.4</td>
</tr>
<tr>
<td>Human</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data augmentation:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paraphrase</td>
<td>18.2</td>
<td>12.5</td>
</tr>
<tr>
<td>Claim Ext.</td>
<td>12.2</td>
<td>9.8</td>
</tr>
<tr>
<td>M. Concat</td>
<td>22.1</td>
<td>71.6</td>
</tr>
<tr>
<td>Ours</td>
<td>34.4</td>
<td>33.0</td>
</tr>
<tr>
<td>Human</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.8: Human evaluation results for our model’s outputs for the fact update task (top) and for the data augmentation task (bottom). The left part of the table shows the geometric SARI score with the three F1 scores that construct it. The right part shows the human’s scores in a 1-5 Likert scale on grammatically of the output sentence and on agreement with the given claim.

patience value of 10, where the stopping condition is defined as the highest delta between accuracy and deletion size on the development set ($\Delta$ in Table 3.10).

For syntactic guidance, we use the constituency parser of [262] and consider continuous spans of length 2 to 10 as masking candidates (without combinations). By doing so, we obtain valid neutrality masks for 38% of the Agr and Dis pairs from the FEVER training dataset. These masks are used for Equation 3.12.

Two-Encoder Pointer Generator  We implemented our proposed multi-sequence-to-sequence model, based on the pointer-generator framework. We use a one layer BiLSTM for encoding and decoding with a hidden dimension of 256. The parameters of the two encoders are shared. The model is trained with batches of size 64 for a total of 50K steps.

BERT Fact-Checking Classifier  We use a BERT [56] classifier, which takes in as input a (claim-evidence) pair separated by a special token, to predict out of 3 labels (Agr, Dis or N). The model is fine-tuned for 3 epochs, which is sufficient to perform well on the task.
**Evidence Regeneration**  Since we are interested in using the generated supporting pairs for data augmentation, we add machine generated cases to the Agr set of the dataset. Adding machine generated sentences to only one of the labels in the data can be ineffective. Therefore, we balance this by regenerating paraphrased refuting evidence for the false claims. This is then added along with all models’ outputs for a balanced augmentation.

**Baselines**

We consider the following baselines for constructing a fact-guided updated sentence:

- **Copy Claim** The sentence of the claim is copied and used as the updated sentence for itself (used only for data augmentation).

- **Paraphrase** The claim is paraphrased using the back-translation method of [303][9] and the output is used as the updated sentence.

- **Claim Extension [Claim Ext.]** A pointer-generator network is trained to generate the updated sentence from an input claim alone. The model is trained on FEVER’s agreeing pairs and applied on the to-be-updated claims during inference.

- **Masked Concatenation [M. Concat]** Instead of our Two-Encoder Generator, we use a pointer-generator network. The residual sentence (output from the masker module) and the claim are concatenated and used as input.

- **Split Encoder without Copy [Split-no-Copy]** Our Two-Encoder Generator, without the copy mechanism. The original text and contradicting claim are passed through each of the encoders.

**3.6.3 Results**

We report the performance of the model outputs for automatic fact-updates by comparing them to the corresponding correct wikipedia sentences. We also have crowd workers score the outputs on grammar and for agreeing with the claim. Additionally, we report the results

[9]: https://github.com/vsuthichai/paraphraser
Table 3.9: Classifiers’ accuracy on the symmetric DEV and TEST splits. The right column (+TURK) shows the accuracy on the TEST set extended to include the 500 responses of turkers for the simulated process and the refuted pairs that they originated from. The BERT classifiers were trained on the FEVER training dataset augmented by outputs of the different methods.

<table>
<thead>
<tr>
<th>MODEL</th>
<th>DEV</th>
<th>TEST</th>
<th>+TURK</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Augmentation</td>
<td>62.7</td>
<td>66.1</td>
<td>77.0</td>
</tr>
<tr>
<td>Paraphrase</td>
<td>60.8</td>
<td>64.6</td>
<td>77.4</td>
</tr>
<tr>
<td>Copy Claim</td>
<td>62.1</td>
<td>63.6</td>
<td>77.4</td>
</tr>
<tr>
<td>Claim Ext.</td>
<td>62.5</td>
<td>65.0</td>
<td>76.8</td>
</tr>
<tr>
<td>M. Concat</td>
<td>60.1</td>
<td>63.7</td>
<td>78.5</td>
</tr>
<tr>
<td>Ours</td>
<td>63.8</td>
<td>67.8</td>
<td>80.0</td>
</tr>
</tbody>
</table>

Table 3.8 reports the automatic and human evaluation results. Our model gets the highest SARI score, showing that it is the closest to humans in modifying the text for the corresponding tasks. Humans also score our outputs the highest for consistency with the claim, an essential criterion of our task. In addition, the outputs are more grammaticality sound compared to those from other methods.

Fact Updates. Following recent text simplification work, we use the SARI [311] method. The SARI method takes 3 inputs: (i) original sentence, (ii) human written updated sentence and (iii) model output. It measures the similarity of the machine generated and human reference sentences based on the deletions, additions and kept n-grams with respect to the original sentence. For human evaluation of the model’s outputs, 20% of the evaluation dataset was used. Crowd-workers were provided with the model outputs and the corresponding supposably consistent claims. They were instructed to score the model outputs from 1 to 5 (1 being the poorest and 5 the highest), on grammaticality and agreement with the claim.

Table 3.8 reports the automatic and human evaluation results. Our model gets the highest SARI score, showing that it is the closest to humans in modifying the text for the corresponding tasks. Humans also score our outputs the highest for consistency with the claim, an essential criterion of our task. In addition, the outputs are more grammaticality sound compared to those from other methods.

Examining the gold answers, we notice that many of them include very minimal and

---

10We use the default up to 4-grams setting.

11Following [83] we use the F1 measure for all three sets, including deletions. The final SARI score is the geometric mean of the ADD, DEL and KEEP score.
local modifications, keeping much of the original sentence. The M. Concat model keeps most of the original sentence as is, even at the cost of being inconsistent with the claim. This corresponds to a high *KEEP* score but a lower SARI score overall, and a low human score on supporting the claim. Claim Ext. and Paraphrase do not maintain the structure of the original sentence, and perform poorly on *KEEP*, leading to a low SARI score. The Split-no-Copy model has the same low ADD score as Claim Ext. since instead of copying the accurate information from the claim, it generates other tokens.

**Data Augmentation.** For 41850 Dis pairs in the FEVER training data, our method generates synthetic evidence sentences leading to 41850 Agr pairs. We train the BERT fact-checking classifier with this augmented data and report the performance on the symmetric dataset in Table 3.9. In addition, we repeat the human evaluation process on the generated augmentation pairs and report it in Table 3.8.

Our method’s outputs are effective for augmentation, outperforming a classifier trained only on the original biased training data by an absolute 1.7% on the TEST set and an absolute 3.0% on the +TURK set. The outputs of the Paraphrase and Copy Claim baselines are not Wikipedia-like, making them ineffective for augmentation. All the baseline approaches augment the false claims with a supported evidence. However, the success of our method in producing supporting evidence while trying to maintain a Wikipedia-like structure, leads to more effective augmentations.

**Masker Analysis.** To evaluate the performance of the masker model, we test its capacity to modify Agr and Dis pairs from the FEVER development set to a neutral relation. We measure the accuracy of the pretrained classifier in predicting neutral versus the percentage of masked words from the sentence. For a finer evaluation, we manually annotated 75 Agr and 76 Dis pairs with the minimal required mask for neutrality and compute the per token F1 score of the masker against them.

The results for different values of the regularization coefficient are reported in Table 3.10 (upper part - with regularization). Increasing the regularization coefficient helps to minimize the mask size and to improve the precision while maintaining the classifier accuracy and
<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>ACC</th>
<th>SIZE</th>
<th>$\Delta$</th>
<th>PREC</th>
<th>REC</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>With syntactic regularization:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.5</td>
<td>5.1</td>
<td>0.0</td>
<td>5.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>.4</td>
<td>80.0</td>
<td>26.3</td>
<td>54</td>
<td>27.2</td>
<td>75.1</td>
<td><strong>39.9</strong></td>
</tr>
<tr>
<td>.3</td>
<td>77.0</td>
<td>27.5</td>
<td>50</td>
<td>25.9</td>
<td>71.6</td>
<td>38.0</td>
</tr>
<tr>
<td>.2</td>
<td>81.6</td>
<td>31.1</td>
<td>51</td>
<td>23.1</td>
<td>74.8</td>
<td>35.3</td>
</tr>
<tr>
<td>.1</td>
<td>80.5</td>
<td>34.7</td>
<td>46</td>
<td>21.9</td>
<td>77.8</td>
<td>34.2</td>
</tr>
<tr>
<td>0</td>
<td>80.0</td>
<td>37.1</td>
<td>43</td>
<td>22.6</td>
<td>81.7</td>
<td>35.5</td>
</tr>
<tr>
<td><strong>Without syntactic regularization:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.5</td>
<td>5.1</td>
<td>0.0</td>
<td>5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>.4</td>
<td>87.8</td>
<td>25.0</td>
<td>63</td>
<td>25.9</td>
<td>68.5</td>
<td><strong>37.6</strong></td>
</tr>
<tr>
<td>.3</td>
<td>5.1</td>
<td>0.0</td>
<td>5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>.2</td>
<td>90.1</td>
<td>35.0</td>
<td>55</td>
<td>22.4</td>
<td>78.7</td>
<td>34.8</td>
</tr>
<tr>
<td>.1</td>
<td>91.2</td>
<td>48.9</td>
<td>42</td>
<td>17.0</td>
<td>85.3</td>
<td>28.4</td>
</tr>
<tr>
<td>0</td>
<td>91.6</td>
<td>100</td>
<td>-8</td>
<td>9.3</td>
<td>100</td>
<td>17.1</td>
</tr>
</tbody>
</table>

Table 3.10: Results of different values of $\lambda$ for the masker with and without syntactic guidance. The left three columns describe the accuracy and average mask size (% of the sentence) over the FEVER development set with the masked evidence and a neutral target label. The right three columns contain the precision, recall and F1 of the masks that we have human annotations for.

the mask recall. However, setting $\lambda$ too large, can collapse the solution to no masking at all. The generation experiments use the outputs of the $\lambda = 0.4$ model.

The masker model makes finding a valid mask in the space of $2^l$ options tractable. However, as mentioned in [13], training an objective of the type shown in Equation 3.11 is unstable. An alternative tractable approach is to enumerate a set of syntactic components of the evidence and score them as potential masks for neutrality. Although this approach is insufficient and might not always work, the cases where the continuous spans satisfy neutrality can help guide the masker training.

Table 3.10 shows results for the masker model with and without syntactic regularization. The syntactic regularization helps to stabilize the performance, allowing a reasonable solution even without any additional constraint on the mask size. Without syntactic regularization, better accuracy can be achieved, but the learning is very unstable and can lead to solutions that mask the whole sentence or keep it as is.
Example Outputs. Examples of outputs from different models are provided in Table 3.11. For the first 3 examples, our model produces a perfect update. In the last example, even though our model gets the year 1930 correct, it modifies the month and nationality to made-up, incorrect values. This is a result of a too aggressive deletion by the masker. The Claim Ext. model typically produces wrong and non-grammatical sentences. The Concat model doesn’t capture the polarizing relation between the two inputs and mostly ignores the claim. The M. Concat model tends to overly generate made-up content instead of copying it from the claim.

3.7 Conclusion

We showed that the FEVER dataset contains idiosyncrasies that can be easily exploited by fact-checking classifiers to obtain high classification accuracies. Evaluating the claim-evidence reasoning of these models necessitates unbiased datasets. Therefore, we suggest a way to turn the evaluation FEVER pairs into symmetric combinations for which a decision that is solely based on the claim is equivalent to a random guess. Tested on these pairs, FEVER-trained models show degraded performance.

Moving forward, we suggest using our symmetric dataset in addition to the current retrieval-based FEVER evaluation pipeline. This way, models could be tested both for their evidence retrieval and classification accuracy and for performing the reasoning with respect to the evidence.

To address this problem, we propose two approaches: (1) importance weighting and (2) automatic data augmentation with fact-guided sentence modification. Our method for sentence modification overcomes the challenges of this conditional generation task by breaking it into two steps. First, we identify the polarizing components in the original sentence and mask them. Then, using the residual sentence and the claim, we generate a new sentence which is consistent with the claim.
<table>
<thead>
<tr>
<th>Original Text</th>
<th>Born in Lawton, Oklahoma and raised in Anaheim, California, Hillenburg became fascinated with the sky as a child and also developed an interest in art.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim</td>
<td>Stephen Hillenburg was fascinated with the ocean as a child.</td>
</tr>
<tr>
<td>Claim Ext.</td>
<td>He in Huntington, Trinidad City Tommy in the, Hillenburg developed he became the of the stage, a senior. business in the adopted in 1847.</td>
</tr>
<tr>
<td>Concat</td>
<td>Born in Lawton, Oklahoma and raised in Anaheim, California, Hillenburg became fascinated with the sky as a child and also developed an interest in art.</td>
</tr>
<tr>
<td>M. Concat</td>
<td>Born in Lawton, Oklahoma and raised in Anaheim, California, Hillenburg became the with the United as the condition and also developed an interest in art.</td>
</tr>
<tr>
<td>Ours</td>
<td>Born in Lawton, Oklahoma and raised in Anaheim, California, Hillenburg became fascinated with the ocean as a child and also developed an interest in art.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original Text</th>
<th>German Startups Group considers 28 of their 42 minority stakeholdings in operationally active companies to be of particular significance to the group.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim</td>
<td>It considers 23 of 43 minority stakeholdings to be significant.</td>
</tr>
<tr>
<td>Claim Ext.</td>
<td>The - soon are the days the eighth capital, is the spending, find divided active by's the original.</td>
</tr>
<tr>
<td>Concat</td>
<td>German Startups Group considers 28 of their their minority stakeholdings in operationally active companies to be of particular significance to the group.</td>
</tr>
<tr>
<td>M. Concat</td>
<td>German Startups Group considers 23 of 18 minority million in operationally active companies to be of particular significance to the group.</td>
</tr>
<tr>
<td>Ours</td>
<td>German Startups Group considers 23 of 43 minority stakeholdings beginning in operationally active companies to be of particular significance to the group.</td>
</tr>
</tbody>
</table>

| Original Text | A sequel, Rio 2, was released on April 11, 2012.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim</td>
<td>Rio’s sequel was released on April 11, 2014.</td>
</tr>
</tbody>
</table>
| Claim Ext.    | In series, Rio is is is released on January 4, 2014,  
| Concat        | A sequel, Rio Rio 2, was released on April 11, 2012  
| M. Concat     | A sequel, Rio 2, was released on August 11, 2014.  
| Ours          | A sequel, Rio 2, was released on April 11, 2014.                                                                 |

| Original Text | Albert S. Ruddy -LRB- born March 28, 1940 -RRB- is a Canadian - born film and television producer.  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim</td>
<td>In 1930, Albert S. Ruddy is born.</td>
</tr>
<tr>
<td>Concat</td>
<td>Albert S. Ruddy -LRB- born March March, 1940 -RRB- is a Canadian - born film and television producer.</td>
</tr>
<tr>
<td>M. Concat</td>
<td>Albert S. Ruddy -LRB- born Hiram 12, 1930 -RRB- is a German - American film and television producer.</td>
</tr>
</tbody>
</table>

Table 3.11: We compare our model outputs against different models. Each example is showing the two input sentences following the output of each model. The Concat model setting is similar to the M. Concat one but the original text is left unmasked. For the Claim Ext. model, only the claim sentence is given as input.
Chapter 4

Robust Fact Verification with Contrastive Evidence

Typical fact verification models use retrieved written evidence to verify claims. Evidence sources, however, often change over time as more information is gathered and revised. In order to adapt, models must be sensitive to subtle differences in supporting evidence. We present VitAMiNC, a benchmark infused with challenging cases that require fact verification models to discern and adjust to slight factual changes. We collect over 100,000 Wikipedia revisions that modify an underlying fact, and leverage these revisions, together with additional synthetically constructed ones, to create a total of over 400,000 claim-evidence pairs. Unlike previous resources, the examples in VitAMiNC are contrastive, i.e., they contain evidence pairs that are nearly identical in language and content, with the exception that one supports a given claim while the other does not. We show that training using this design increases robustness—improving accuracy by 10% on adversarial fact verification and 6% on adversarial natural language inference (NLI). Moreover, the structure of VitAMiNC leads us to define additional tasks for fact-checking resources: tagging relevant words in the evidence for verifying the claim, identifying factual revisions, and providing automatic edits via factually consistent text generation.
4.1 Introduction

Determining the truthfulness of factual claims by comparing them to textual sources of evidence has received intense research interest in recent years. An underlying, but often overlooked, challenge for this paradigm, however, is the dynamic nature of today’s written resources. An extraordinary amount of new information becomes available daily; as a result, many consequential facts are established, changed, or added to over time. We argue that the quality of fact verification systems should be measured by how well they adjust to new evidence. In this way, we seek to advance fact verification by requiring that models remain reliable and robust to the change present in practical settings.

To this end, we focus on fact verification with contrastive evidence. That is, we infuse the standard fact verification paradigm with challenging cases that require models to be sensitive to factual changes in their presented evidence (hereon referred to interchangeably as “context”). We present VITAMINC, a new large-scale fact verification dataset that is based on factual revisions to Wikipedia. The key concept is exemplified in Figure 4-1: there a factual revision yields a contrastive pair of contexts that are nearly identical in language and content—except that one context refutes the given claim, while the other supports it.

This type of contrastive structure exposes existing deficiencies in model behavior. To illustrate this, we train a classifier on the popular FEVER fact verification dataset [272] and evaluate it on contrastive claim-evidence pairs. We find that the model flips its prediction

\[^{1}\text{Etymology of VITAMINC:} \text{Contrastive evidence keeps fact verification models robust and healthy, hence \textit{“Vitamin C.”}}\]

Figure 4-1: In VITAMINC, we focus on Wikipedia revisions in which the factual content changes. This example revision now supports an initially refuted claim.
from the original verdict on only 56% of the contrastive cases. When examples from \textsc{Vitamin}C are included during training, however, the model’s sensitivity increases, flipping on 86% of contrastive cases.

Such context-sensitive inference has two main benefits. First, it ensures that the model considers the provided evidence rather than relying on built-in static knowledge, such as that obtained via language model pre-training \cite{209, 225}. This is particularly important for scenarios in which the source of truth is mutable (e.g., the current US president, or new declarations as in Figure 4.1). Second, this setting discourages certain biases and idiosyncrasies—such as exploiting differences in how true vs. false claims are posed—that are common in similar crowd-sourced datasets \cite{211, 246}. Indeed, we show that augmenting both fact verification models and NLI models with \textsc{Vitamin}C data improves their robustness to adversarial inputs.

Furthermore, our emphasis on contrastive contexts allows us to expand on the scope of commonly considered tasks. Most of the fact verification literature focuses on resolving claims to be true or false \cite{214, 270, 299}. The surrounding ecosystem, however, includes additional challenges, some of which we explore here: Documents such as Wikipedia articles are updated frequently; which edits represent factual changes? For a given claim and (refuting or supporting) evidence pair, which words or phrases in the evidence are most relevant? If we know that a certain claim is true, can we modify an out-dated document to be consistent with it? We show that the unique structure of our \textsc{Vitamin}C dataset can be leveraged to provide both supervised and distantly supervised data for these new questions.

Our key contributions are as follows:

1. We pose a contrastive fact verification paradigm that requires sensitivity to changes in data;

2. We introduce \textsc{Vitamin}C, a new large-scale dataset that supports this paradigm;

3. We demonstrate that training on \textsc{Vitamin}C leads to better performance on standard tasks;

4. We show how \textsc{Vitamin}C opens the door to additional research directions in fact verification.
4.2 Related Work

We provide a more extensive review of previous work on fact verification and annotation bias in Section 3.2. We summarize here the recent work most related to VitaminC.

4.2.1 Fact Verification

The FEVER dataset [272] fueled the development of many fact-checking models [e.g., see 98, 190, 191, 316, *inter alia*]. The claim creation process, however, required crowd-workers to write claims related to Wikipedia articles, and was found to engender biases that allow an evidence-agnostic model to achieve unexpectedly high performance [246]. Other recent datasets cover verification against tables [44], relational databases [123], Wikipedia references [240], multiple articles [122], and search snippets [9]. These resources all assume static ground truths. [42] collected subjective claims and opinions paired with references that provide different levels of agreement. In contrast, *VITAMINC* compares objective claims to a dynamic source of truth, and requires models to change their verdicts accordingly.

4.2.2 Annotation Bias

Annotation artifacts are common in many NLP datasets, and affect performance on adversarial and contrastive examples [77, 224, 231]. Sentence-pair inference tasks such as fact verification [201, 246] and NLI [95, 179, 211, 281] are no exception. Alleviating this bias requires either modeling solutions [128, 217, 252, 271, 285], which have limited effectiveness [284], or adversarially removing troublesome training examples [26] or manually collecting new ones [192, 273], which is model specific. Instead, our dataset design avoids single-sentence artifacts and provides model-agnostic challenging examples that increase the robustness of trained models.

4.2.3 Explainability

Current fact verification datasets provide sentence-level rationales [57, 208] but do not enforce the model’s verdict to rely on them—leading to a potential discrepancy. In contrast,
<table>
<thead>
<tr>
<th>Factual</th>
<th>Wikipedia sentences before and after a revision, presented with VITAMIN C claims if the revision is factual.</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ Before</td>
<td>More stringent actions were taken in China once the seriousness of the outbreak became apparent, such as quarantining entire cities affecting 60 million individuals in Hubei, and strict travel bans.</td>
</tr>
<tr>
<td>After</td>
<td>More drastic actions were taken in China once the severity of the outbreak became apparent, such as quarantining entire cities affecting 60 million individuals in Hubei, and strict travel bans.</td>
</tr>
<tr>
<td>✓ Before</td>
<td>In animals, spaying involves an invasive removal of the ovaries, but rarely has major complications other than that spayed animals tend to gain weight.</td>
</tr>
<tr>
<td>After</td>
<td>In animals, spaying involves an invasive removal of the ovaries, but rarely has major complications; the superstition that it causes weight gain is not based on fact.</td>
</tr>
<tr>
<td>Claim 1</td>
<td>Spayed animals gain weight.</td>
</tr>
<tr>
<td>Claim 2</td>
<td>Weight gain in spayed animals is a superstitious myth.</td>
</tr>
<tr>
<td>✗ Before</td>
<td>As of 16 March, more than 182,000 cases of the disease have been reported in over 160 countries and territories, resulting in around 79,000 recoveries and more than 7,100 deaths.</td>
</tr>
<tr>
<td>After</td>
<td>As of 16 March, more than 182,000 cases of the disease have been reported in over 160 countries and territories, resulting in more than 7,100 deaths and around 79,000 recoveries.</td>
</tr>
<tr>
<td>✓ Before</td>
<td>Global hybrid sales are led by the Prius family, with sales of 4.7 million units representing 66.8% of TMC worldwide sales of 7,053 million Lexus and Toyota units through September 2014.</td>
</tr>
<tr>
<td>After</td>
<td>Global hybrid sales are led by the Prius family, with sales of 5,264 million units representing 65.4% of TMC worldwide sales of 8,048 million Lexus and Toyota units delivered through July 2014.</td>
</tr>
<tr>
<td>Claim 1</td>
<td>Prius sold less than 5 million units, representing over 65.5% of TMC worldwide sales.</td>
</tr>
<tr>
<td>Claim 2</td>
<td>Prius sold more than 5 million units, representing less than 65.5% of TMC worldwide sales.</td>
</tr>
</tbody>
</table>

Table 4.1: Examples of non-factual revisions vs. factual revisions, and the claims associated with the later. Factual updates change the outcome (i.e., true or false) of a claim that might be in question. Accordingly, the verdict of a classifier should change based on the version presented. Modified words are underlined and colored.

VITAMIN C ensures the verdict is conditioned on the retrieved evidence by making the model adjust its predictions accordingly. Moreover, we use the revision history as distant supervision for word-level rationales, allowing for finer-grained explanations [33, 155, 215, 274].

### 4.2.4 Factually Consistent Generation

Generating texts that match given facts is a known challenge [66, 142, 158, 198, 252, 277] as language models tend to degenerate and hallucinate [109, 245, 324]. Moreover, evaluation is non-trivial, and usually manual. VITAMIN C includes supervised data for training sequence-to-sequence models, and provides automatic evaluation via the fact verification classifier.
4.3 The VITAMIN C Dataset

VITAMIN C (abbreviated VitC) is based on revisions to English Wikipedia. Wikipedia has become a comprehensive online resource that is rigorously maintained by a large and active community [19]. While adversaries do try to insert disinformation, popular pages are usually quickly corrected [143]. Furthermore, Wikipedia’s policies dictate that its content should be written from a neutral perspective—or should otherwise objectively state all points of view. These properties make Wikipedia a suitable source of evidence for fact verification models. In the following section, we outline our process for mining factual revisions from Wikipedia.

4.3.1 Collecting Factual Revisions

We collected the 5K most-viewed English Wikipedia articles as of January 2020, along with any additional articles referred from them (on average 100 per article). We also included all articles from the FEVER dataset [272]. For each article, we retrieved up to 500 of its most recent revisions. In May 2020, we added all COVID-19 related articles and all of their 41K revisions at the time. Combined together, this resulted in a total of ~200 million revisions. For each revision, we identified all of the modified sentences and stored two versions: (1) before, and (2) after the edit.

In our task, we are only interested in edits made with an intent to introduce a factual modification—i.e., a change for which one can make a claim that is supported by one sentence, but not by the other. To expedite annotation, we trained a BERT classifier [56] on a small labeled set of revised sentences determined to be factual [312], and used this model to select the top 305K edited sentences from the corpus for manual annotation. Trained human annotators were then presented with the sentence pairs, and were asked to mark the ones that indeed represented a factual change. Sentences lacking self-contained context

---

5 Many edits only reflect grammatical corrections, paraphrasing, or “Wikification” (text formatting/page linking).

https://wikimediafoundation.org/covid19

102
were filtered (e.g., short expressions from tables or bulleted lists). Example annotations are presented in Table 4.1. Note that these annotations can also be recursively recycled for re-training the automated BERT classifier in the future to expand the corpus further (we also introduce this as a task, see §4.4.1).

### 4.3.2 Writing Claims

The factual Wikipedia revisions guide us in creating challenging claims for fact verification. For each revision, annotators were asked to write two symmetric claims related to the same edit:

1. The first should be supported by the *original* sentence and refuted by the *revised* sentence;
2. The second should be supported by the *revised* sentence and refuted by the *original* sentence.

When an explicit contradiction was not possible, a *not enough information* (NEI) relation was used. A group of 70 native English speakers wrote and reviewed claims. During the annotation period, annotations were delivered in weekly batches, from which we examined random samples to provide feedback and request corrections. Annotators were instructed to write short and self-contained claims. Furthermore, annotators were instructed to avoid copying exact phrases and values when possible, in order to avoid a bias for substantially higher word overlap in supporting pairs over refuting pairs. For example, rather than stating, "there are \( x \) confirmed cases of coronavirus in the US", one can write "there are more than \( z \) confirmed cases of coronavirus in the US", which is supported if \( x > z \) and refuted otherwise. For revisions that only add new information or that remove outdated facts without replacing them, annotators wrote a single claim.

### 4.3.3 Adding Synthetic Revisions

Naturally, the real Wikipedia revisions we collect mostly describe facts that frequently change over time, or that are prone to mistakes and corrections (such as quantitative values,

---

We sourced our annotators through TransPerfect.

---
see Appendix A.1.1). Sensitivity to contrastive contexts, however, is desirable behavior for any claim. This can both ensure consistency with external sources of truth, and improve the model’s faithfulness via connecting the verdict with a specific evidence. For example, we require the model to not only classify the claim “Tom Hanks was honored by a president” as true, but to also change its verdict to false if paired with a (fictional) contrasting evidence. As a result, we can verify that the model prioritizes sentence-pair inference over memorization, which can help it generalize better. Therefore, we use the FEVER dataset to augment VITAMINC with synthetic revisions to Wikipedia sentences.

We follow the setting of 246 to expand claim-evidence pairs from FEVER. Specifically, given a false claim from FEVER, we ask annotators to edit the sentence that refutes it so that it will then support the originally false claim. Additionally, we ask them to write a new claim that is refuted by the new, modified sentence, but that is supported by the original version. Following this method, we obtain two claims where each can be supported or refuted by the original, or the synthetically revised, sentence. We follow the same process for constructing synthetic examples using true claims, but with flipped labels.

### 4.3.4 Dataset Statistics

In total, 304,671 revised Wikipedia sentences were examined by annotators, of which 107,056 (35%) were found to express a factual modification and were passed to the group of expert annotators for claim writing. As two symmetric claims with opposing facts were created (when possible) for each revision, this resulted in a total of 325,724 total claim-evidence pairs. We collected 163,180 additional pairs following the synthetic process. The data was partitioned as shown in Table. The assignment was done randomly by article, and is consistent with FEVER for overlapping articles. Appendix A.1 contains additional details.
### Table 4.2: Number of claim-evidence pairs in VITAMINC.

Breakdowns of real vs. synthetic revisions are presented on the left and right of each cell, respectively.

<table>
<thead>
<tr>
<th>Split</th>
<th>Supports</th>
<th></th>
<th>Refutes</th>
<th></th>
<th>NEI</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>Syn</td>
<td>Real</td>
<td>Syn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Train</td>
<td>124,864</td>
<td>60,850</td>
<td>71,108</td>
<td>60,850</td>
<td>52,981</td>
<td>-</td>
</tr>
<tr>
<td>Dev</td>
<td>21,102</td>
<td>10,382</td>
<td>12,146</td>
<td>10,382</td>
<td>9,042</td>
<td>-</td>
</tr>
<tr>
<td>Test</td>
<td>17,306</td>
<td>10,358</td>
<td>9,907</td>
<td>10,358</td>
<td>7,268</td>
<td>-</td>
</tr>
</tbody>
</table>

The outbreak was first identified in Wuhan, Hubei, China in December 2019 and recognized [...].

### 4.4 VITAMINC Tasks

The unique structure of VITAMINC allows us to derive annotations that provide a novel source of supervision for several fact-verification-related tasks. We describe the four main tasks we consider in this work, along with baseline models: (1) factual revision flagging, (2) fact verification, (3) word-level rationales, and (4) factually consistent generation. Figure 4-2 illustrates an example from VITAMINC. We use the following notations:

- \( \mathcal{C} \) is the space of short sentences that express an arbitrary factual statement that can potentially be verified or debunked by external sources.
- \( \mathcal{S} \) is the space of sentences that can be found in a trusted online resource (Wikipedia in this study).
- \( (s_{t-1}, s_t) \) denotes the two versions of a sentence that was revised from \( s_{t-1} \) to \( s_t \in \mathcal{S} \).
- \( \text{rel}(c, s) \) denotes the relation between the claim \( c \in \mathcal{C} \) and observed evidence \( s \in \mathcal{S} \)—which can either support \( c \) (SUP), refute it (REF), or not contain enough information (NEI).
4.4.1 Factual Revision Flagging

Online resources like Wikipedia are continuously changing. In order to remain a reliable and neutral source for recent information, its active community of users must constantly verify and correct the revisions of others. We define factual revision flagging as the task of identifying revisions that introduce a factual change—e.g., by either modifying a certain fact, adding a new one, or removing an existing one. Such an automated detection process can help the community moderate important articles by serving as a watchdog for factual revisions. Furthermore, tracking factual revisions to certain articles can potentially help keep reliant articles consistent (e.g., citing articles, or non-English versions).

We pose factual revision flagging as a binary classification function $f_{\text{flag}}: \mathcal{S} \times \mathcal{S} \rightarrow \{0, 1\}$, where for a revision $(s_{t-1}, s_t)_i$, we set $y_i = 1$ iff there exists a claim in $\mathcal{C}$ whose label (SUP or REF) changes as a result of the edit (i.e., SUP $\rightarrow$ \{REF, NEI\} or REF $\rightarrow$ \{SUP, NEI\}). Table 4.1 provides example factual and non-factual revisions. We evaluate the following baseline models:

**Edit Distance.** We measure the edit distance between $s_{t-1}$ and $s_t$, assuming that larger edits are more likely to represent substantive changes. We tune a decision threshold on the validation set.

**BOW.** We use an MLP on top of a bag-of-words representation. Each sentence is encoded as $e_s$, the average fastText [22] word embedding of its edited words (i.e., that were removed or modified in the revision). The MLP input is then taken as $[e_{t-1}; e_t; |e_t - e_{t-1}|; e_t \cdot e_{t-1}]$.

**ALBERT.** We train the ALBERT transformer [146] using either only the edited words (diff), or the full sentence pair (full).
4.4.2 Fact Verification

Our basic setting is similar to the inference task of the FEVER dataset. We predict the verdict for a claim given an observed evidence, \( f_{\text{verdict}} : \mathcal{C} \times \mathcal{S} \rightarrow \{\text{SUP}, \text{REF}, \text{NEI}\} \). The FEVER dataset, however, contains independent claim-evidence pairs. In our setting, we have claims paired with revisions such that \( \text{rel}(c_i, s_{t-1}) \neq \text{rel}(c_i, s_t) \), creating contrastive triplets. For example, the claim in Figure 4-2 states that the COVID-19 outbreak was identified before December. VITAMINC matches it with two different contexts (before and after the presented revision), that can either support or refute that claim.

Our baseline model is an ALBERT sentence-pair classifier that predicts \( \text{rel}(c, s) \). Compared to BERT [56], it uses fewer parameters by shrinking the embedding size and sharing layers, which we find to improve robustness.

4.4.3 Word-level Rationales

Word-level rationales provide useful explanations for predictions of neural models [155]. Such explanations can be particularly useful for semi-automated fact verification, since they allow users to quickly interpret and trust the model’s verdict. In Figure 4-2, for example, the date of the first identified case can explain the verdict for the claim.

As first proposed by [155], the standard definition of extractive rationales asks for selecting the minimal set of input tokens that is sufficient for preserving the model’s prediction. Here we use a slightly modified definition following [252], where we identify the minimal set of evidence tokens where removing them will change the input’s label to NEI.

We pose this task as conditional masking, where we learn a function \( f_{\text{rationale}} : \mathcal{C} \times \mathcal{S} \rightarrow \{0, 1\}^n \), where \( n \) is the length of an evidence \( s \in \mathcal{S} \). Given an evidence \( s = (x_1, \ldots, x_n) \) and a claim \( c \), where \( \text{rel}(c, s) \in \{\text{SUP}, \text{REF}\} \), we want to find a mask \( m \) such that \( \text{rel}(c, s \odot m) \neq \text{rel}(c, s) \).

---

7To focus on the inference task, as opposed to a full end-to-end system, we assume that we have access to an oracle retriever.

8[228] showed that explanations can increase the agreement between users and expert fact-checkers.
\( m \) = NEI, where
\[
s \odot m = \begin{cases} 
    x_i & \text{if } m[i] = 0; \\
    <\text{mask}> & \text{if } m[i] = 1.
\end{cases}
\]

Moreover, we want \( m \) to be as sparse as possible. Intuitively, \( s \odot m \) could be viewed as an incomplete revision in which the masked words that have not yet been filled in will determine the relation with the claim. We say that \( m \) reveals the most responsible words in \( s \) for resolving \( c \). Following \cite{252}, we formulate an unsupervised objective as
\[
\min \sum_{i=1}^{n} m_i \text{ s.t. } \text{rel}(c, s \odot m) = \text{NEI}.
\] (4.1)

We evaluate the quality of \( m \) by comparing it in terms of F1 to both (1) \( m_{\text{edit}} \), the non-stopwords removed or replaced in the true revision (i.e., \textit{edit prediction}), and (2) \( m_{\text{manual}} \), a manually annotated “human” reference, (i.e., \textit{rationale prediction}). We implement the following two baselines:

\textbf{Unsupervised.} As in \cite{252}, we optimize a Lagrangian relaxation of Eq. 4.1, where
\[
\mathcal{L}_{\text{us}} := -\log p(\text{rel}(c, s \odot m) = \text{NEI}) + \frac{\lambda}{n} \sum_{i=1}^{n} m_i.
\]

We keep the \textit{rel} classifier (from §4.4.2) fixed, and train a separate ALBERT model to predict the mask \( m \) using a Gumbel softmax \cite{117}.

\textbf{Distantly Supervised.} By leveraging opposing claims present in \textsc{Vitaminc}, we are able to identify \( m_{\text{edit}} = \text{diff}(s_{t-1}, s_t) \)—i.e., the non-stopwords that are deleted or replaced in \( s_{t-1} \) when compared to \( s_t \). We then use \( m_{\text{edit}} \) as distant supervision for \( m \), where
\[
\mathcal{L}_{\text{ds}} = -\frac{\gamma}{n} \sum_{i=1}^{n} \log p(m_i = m_{\text{edit},i}).
\]
We combine both the \( \mathcal{L}_{\text{us}} \) and \( \mathcal{L}_{\text{ds}} \) losses.

\subsection*{4.4.4 Factually Consistent Generation}

As facts change, the sources reporting them must change as well to reflect the most recent information. In \textsc{Vitaminc}, this is reflected via the active revisions to Wikipedia. We
simulate automating this process by considering two generation tasks:

**Automatic Revisions.** Given an outdated context \( s_{t-1} \) and an updated claim \( c \), we learn \( f_{\text{revise}} : S \times C \to S \) to produce a new context \( s_t \) that minimally modifies \( s_{t-1} \) to agree with \( c \). For example, one can change \( s_{t-1} \) in Figure 4-2 to state “before December” in order to agree with the claim.

**Claim Extraction.** Given a revision \( (s_{t-1}, s_t) \), we learn \( f_{\text{extract}} : S \times S \to C \) to produce a short claim \( c \) that expresses the factual change.

In both tasks, the output should satisfy \( \text{rel}(c, s_t) = \text{SUP} \) while \( \text{rel}(c, s_{t-1}) = \text{REF} \). We use \( f_{\text{verdict}} \) (§4.4.2) to evaluate this requirement. We experiment with both BART-base [157] and T5-base [219] sequence-to-sequence transformer-based generators. For the revision task, we concatenate \( s_{t-1} \) and \( c \) with a separator and train the model to predict \( s_t \). For the claim extraction task, we combine the input pair \( (s_{t-1}, s_t) \) into a single sentence that visualizes the revision (e.g., “sales of \{4.7 \to 5.4\} million”).

### 4.5 Experiments

We present and analyze results for the models described in Section 4.4. Our analysis attempts to evaluate several questions: (1) How well can the current state-of-the-art models perform on the VITAMINC tasks? (2) Does VITAMINC increases the robustness of models against adversarial examples? (3) Can VITAMINC improve interpretability by providing supervision for anchoring words?

#### 4.5.1 Related Datasets

In addition to VITAMINC, we train and evaluate on several related datasets, which we briefly describe:

**FEVER** [272]: A popular fact verification dataset based on Wikipedia. We use the provided SUP and REF claim-evidence pairs. For NEI claims, we randomly sample neutral evidence from the article with the highest BM25 score [71].
MNLI [304]: A large and diverse dataset for natural language inference. The three-way sentence-pair entailment prediction is similar to fact verification. We use the hypothesis as the claim and the premise as the evidence and evaluate on the “mismatched” evaluation set.

Symmetric [246]: A set of challenging symmetric, synthetic extensions to FEVER’s evaluation set that avoid claim-only bias.

Adversarial [276]: Adversarial examples created by participants of the FEVER 2.0 shared task. Teams were asked to create claims that break FEVER-trained models. We take all SUP and REF claims and their gold evidence sentences.

Triggers [8]: A set of 186 FEVER claims paraphrased adversarially to contain universal adversarial triggers [297]. Its small size leads to high variance results.

ANLI [192]: An adversarial dataset for MNLI- and FEVER-based models. The creation was performed in three iterative rounds in which a model was trained, and then crowdworkers devised adversarial inputs, and the process repeated.

PAWS [319]: A dataset of altered Wikipedia sentences using word swapping and back-translation. Human annotators labeled whether the modified sentence is a paraphrase or not. We evaluate whether a PAWS-trained classifier can be used for our factual revision flagging task.

4.5.2 Factual Revision Flagging

Table 4.3 shows the results of our baseline models on the factual revision flagging task. First, we notice that a model trained on the PAWS dataset (reaching 93.42 F1 score on PAWS test) does not transfer well to the flagging task, and performs on par with a simple edit distance heuristic. We hypothesize that this is a result of the entity scrambling technique used to synthetically revise sentences in PAWS, which is different from the edits introduced by real, factual Wikipedia revisions in practice.
Table 4.3: Factual revision flagging scores for models aware of the full sentence-pair (full) and aware only of the modified words (diff). We use ALBERT-base.

Table 4.4: Test accuracy of fact verification and NLI models. VITAMINC-trained models are more robust to adversarial examples and more sensitive to contrastive contexts. The rightmost column shows the percent of FEVER claims in which the prediction flipped when presented with contrastive contexts.

Second, we see that the performance of neural models trained on the VITAMINC flagging task increases with richer inputs and more advanced models—demonstrating the complexity of the task. The ALBERT (diff) model that uses only the modified word sequences from each sentence (i.e., contextual within a subspan) improves the AUC by 10 points over a BOW model that gets a similar input. The ALBERT (full) model that receives the full sentences as input (i.e., has access to even more context), further improves the AUC by 2 points. Nevertheless, the best model still only reaches 83 macro-F1, indicating the difficulty of this task.
4.5.3 Fact Verification

Table 4.4 summarizes the results for classifiers trained on fact verification and NLI datasets. Verifying claims against real revisions proves to be the hardest. The best model achieves 89% accuracy, lower than that on either VITAMINC’s synthetic cases or the original FEVER examples. Including VITAMINC examples in the training data drastically increases models’ sensitivity to contrastive examples (rightmost column)—while preserving the in-domain accuracy (only −0.42% for FEVER and +0.12% for MNLI with ALBERT-xlarge). Another evidence for the generalization properties conferred by VITAMINC is its zero-shot performance to both other datasets. An ALBERT-xlarge model trained only on VITAMINC reaches 76% and 79% accuracy on FEVER and MNLI, respectively. In contrast, the transfer accuracy for MNLI→FEVER is 70% and for FEVER→MNLI is only 38%.

Most importantly, models trained with VITAMINC perform better on challenging adversarial datasets. On the other hand, simply augmenting FEVER data with MNLI data has a limited effect on adversarial examples. We conjecture that the contrastive nature of VITAMINC helps models better learn the relations between the claims and evidences—and to avoid relying on certain artifacts that do not generalize well.

We’ve also tried augmenting FEVER with ANLI for an ALBERT-xlarge model and find it to achieve only 73%, 91%, and 34% on Adver., Sym., and Triggers, respectively.
Table 4.5: The distant token-level supervision of VITAMINC improves the edit prediction, and as result identifies the anchoring words (rationales) more accurately.

To further probe the value of VITAMINC examples compared to FEVER ones (SUP and REF only), we compose training sets of 100K examples using different ratios of the two datasets. As shown in Figure 4-3 including more VITAMINC pairs continuously improves the performance on the challenging adversarial and symmetric evaluation sets.

As an additional qualitative experiment, given the recent successes of huge language models such as GPT-3 [29], we explore whether such models develop sufficient context sensitivity on their own. Appendix A.3 shows the results of classifying several claims using a few-shot GPT-3 model. We find that GPT-3 still largely under-performs our VITAMINC-trained models in terms of sensitivity—demonstrating the importance of using VITAMINC’s unique structure during training.

4.5.4 Word-level Rationales

Table 4.5 shows the results of our baseline models for identifying word-level rationales (i.e., anchoring words in the evidence). While our unsupervised model is able to uncover some patterns, directly leveraging the structure of VITAMINC to obtain distant supervision for likely anchoring words (i.e., token labels) improves both the edit prediction and the word-level rationale prediction performance. Example predictions are provided in Appendix A.5.

4.5.5 Factually Consistent Generation

Table 4.6 presents the results on factually consistent generation. We find BART to perform better in both of our generation tasks (though we only tried the default setting). The BLEU score [197] is lower in the claim extraction task since there is freedom in how to phrase the

\[10\] We evaluate rationales using a manually annotated test set of 300 examples (150 each from VitC real and VitC synthetic).
Table 4.6: Factually consistent generation results. Higher is better for all scores and the max value is 100 (except for BLEURT). $f_{\text{verdict}}$ is the score of our VITAMINC-trained ALBERT-base model on the outputs. For manual evaluation, outputs were rated by their grammaticality and by how much the evidence supports the claim (SUP). For reference, human-written pairs received 75.75 and 76.0 average scores for Grammar and SUP, respectively.

The revision generator aims to modify sentences so that they agree with a given claim. According to our fact verification model’s verdict, it succeeds in doing so 76% of the time. Furthermore, revisions should resemble real ones, and preserve the remaining content that is unrelated to the claim. The SARI KEEP F1 of 75 shows that the model and the reference mostly agree on parts of the sentence that should be kept unchanged.

We find that the token-based measures and our $f_{\text{verdict}}$ metric agree well with human (manual) evaluation scores. We randomly sampled 100 generated and human-written sentences per task, and asked workers on Amazon MTurk to rate their grammaticality and whether the evidence $s_t$ supports the claim. The scores of the generated sentences were on par with the human-written ones, indicating the high-quality of our outputs.

Table 4.7 presents two example generations for the claim extraction task (we provide additional qualitative examples in Appendix A.5). Our model is able to efficiently extract a self-contained claim that expresses the correct fact after the edit. As in §4.5.3, we also explore how GPT-3 handles this task (we provide two demonstrations in the prompt). Compared to the BART model trained on VITAMINC, GPT-3 appears to make more factually inconsistent or unsupported generations (see Appendix A.3 for more details). Encouragingly, our $f_{\text{verdict}}$ classifier is still able to pick up on this—as demonstrated by the predictions in
<table>
<thead>
<tr>
<th>((s_{t-1}, s_t))</th>
<th>2020 coronavirus pandemic in Germany: there have been(*)349 \to 444) confirmed cases and 16 recoveries.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BART (VitC)</td>
<td>More than 400 people have tested positive for COVID-19 in Germany. (f_{\text{verdict}}(c, s_t) = \text{SUP})</td>
</tr>
<tr>
<td>GPT-3 (T=0)</td>
<td>As of 14 March, there have been more than 350 confirmed cases of the virus in Germany</td>
</tr>
<tr>
<td>GPT-3 (T=0.7)</td>
<td>As of March 12, there have been more than 400 confirmed cases and 20 reported deaths</td>
</tr>
<tr>
<td>Reference</td>
<td>There have been more than 400 confirmed coronavirus cases in Germany.</td>
</tr>
<tr>
<td>((s_{t-1}, s_t))</td>
<td>Diego Corrales: Corrales was born to a(<em>)Puerto Rican \to African American) father and a (</em>)Dominican \to Mexican) mother.</td>
</tr>
<tr>
<td>BART (VitC)</td>
<td>Diego Corrales’ father is African American and his mother is Mexican. (f_{\text{verdict}}(c, s_t) = \text{SUP})</td>
</tr>
<tr>
<td>GPT-3 (T=0)</td>
<td>Corrales was born to a Puerto Rican father and a Mexican mother (f_{\text{verdict}}(c, s_t) = \text{REF})</td>
</tr>
<tr>
<td>GPT-3 (T=0.7)</td>
<td>Corrales was born to a father from Puerto Rico and a mother from the Dominican Republic (f_{\text{verdict}}(c, s_t) = \text{REF})</td>
</tr>
<tr>
<td>Reference</td>
<td>Diego Corrales’ father was African American and his mother Mexican. (f_{\text{verdict}}(c, s_t) = \text{SUP})</td>
</tr>
</tbody>
</table>

Table 4.7: Example outputs for expressing claims that reflect the factual changes in a single Wikipedia revision. The BART-base model is trained on VITAMIN\(C\) data, while GPT-3 is applied in a 2-shot setting with a temperature of 0 or 0.7 (see Appendix A.3). The revision \((s_{t-1}, s_t)\) is given to the model as a single sentence, where the edits are between curly brackets. The human-written claim is provided for reference. The rightmost column contains the prediction of our ALBERT-xlarge \(f_{\text{verdict}}(c, s_t)\) model (trained on VITAMIN\(C\)) when using the generated claim.

For example, classifying the report about 20 deaths as \text{NEI} since it is not part of the source. Once again, this serves to qualitatively demonstrate the effectiveness of leveraging VITAMIN\(C\).

### 4.6 Conclusion

We presented VITAMIN\(C\), a large-scale dataset for training and evaluating fact verification models using contrastive contexts. Our novel method of leveraging factual revisions to Wikipedia enabled us to create challenging examples in which a claim is paired with contexts that are lexically similar, yet factually opposing. Our results illustrated that training on VITAMIN\(C\) improves classifier sensitivity to subtle changes in evidence, and increases their robustness to adversarial examples.

Furthermore, we formulated several new, important tasks for fact verification that VITAMIN\(C\) allows us to test. We showed how the dataset’s unique “before and after”
structure lends itself to training classifiers to flag factual revisions. In addition, for factual revisions, the edits reveal which words in the evidence are the most critical—which helps supervise word-level rationale models for better interpretability. Finally, we demonstrated that VITAMINC can help with factually consistent text generation. We hope that this work and the range of tasks it presents will motivate and support the fact verification field in developing reliable models that can adapt to dynamically changing evidence.
Chapter 5

Efficient Evidence Retrieval via Cascaded Conformal Prediction

In this chapter, we present a novel approach for conformal prediction (CP), in which we aim to identify a set of promising prediction candidates—in place of a single prediction. This set is guaranteed to contain a correct answer with high probability, and is well-suited for many open-ended classification tasks. In the standard CP paradigm, the predicted set can often be unusably large and also costly to obtain. This is particularly pervasive in settings where the correct answer is not unique, and the number of total possible answers is high. We first expand the CP correctness criterion to allow for additional, inferred “admissible” answers, which can substantially reduce the size of the predicted set while still providing valid performance guarantees. Second, we amortize costs by conformalizing prediction cascades, in which we aggressively prune implausible labels early on by using progressively stronger classifiers—again, while still providing valid performance guarantees. We demonstrate the empirical effectiveness of our approach for multiple applications in natural language processing and computational chemistry for drug discovery.

5.1 Introduction

The ability to provide precise performance guarantees is critical to many classification tasks [6,121,119]. Yet, achieving perfect accuracy with only single guesses is often
out of reach due to noise, limited data, insufficient modeling capacity, or other pitfalls. Nevertheless, in many applications, it can be more feasible and ultimately as useful to hedge predictions by having the classifier return a set of plausible options—one of which is likely to be correct.

Consider the example of information retrieval (IR) for fact verification. Here the goal is to retrieve a snippet of text of some granularity (e.g., a sentence, paragraph, or article) that can be used to verify a given claim. Large resources, such as Wikipedia, can contain millions of candidate snippets—many of which may independently be able to serve as viable evidence. A good retriever should make precise snippet suggestions, quickly—but do so without excessively sacrificing sensitivity (i.e., recall).

Conformal prediction (CP) is a methodology for placing exactly that sort of bet on which candidates to retain \[294\]. Concretely, suppose we have been given \(n\) examples, \((X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n\), as training data, that have been drawn exchangeably from an underlying distribution \(P\). For instance, in our IR setting, \(X\) would be the claim in question, \(Y\) a viable piece of evidence that supports or refutes it, and \(\mathcal{Y}\) a large corpus (e.g., Wikipedia). Next, let \(X_{n+1}\) be a new exchangeable test example (e.g., a new claim to verify) for which we would like to predict the paired \(y \in \mathcal{Y}\). The aim of conformal prediction is to construct a set of candidates \(C_n(X_{n+1})\) likely to contain \(Y_{n+1}\) (e.g., the relevant evidence) with distribution-free marginal coverage at a tolerance level \(\epsilon \in (0, 1)\):

\[
P( Y_{n+1} \in C_n(X_{n+1})) \geq 1 - \epsilon; \quad \text{for all distributions } P. \tag{5.1}
\]

The marginal probability above is taken over all the \(n+1\) calibration and test points \\{\((X_i, Y_i)\)\}_{i=1}^{n+1}. A classifier is considered to be valid if the frequency of error, \(Y_{n+1} \notin C_n(X_{n+1})\), does not exceed \(\epsilon\). In our IR setting, this would mean including the correct snippet at least \(\epsilon\)-fraction of the time. Not all valid classifiers, however, are particularly useful (e.g., a trivial classifier that merely returns all possible outputs). A classifier is considered to have good predictive efficiency if \(\mathbb{E}[|C_n(X_{n+1})|] \) is small (i.e., \(\ll |\mathcal{Y}|\)). In our IR setting, this would mean not returning too many irrelevant articles—or in IR terms, maximizing precision while holding the level of recall at \(\geq 1 - \epsilon\) (assuming \(Y\) is a single answer). In practice, in domains where the number of outputs to choose from is large and the
Figure 5-1: A demonstration of our conformalized cascade for \( m \)-step inference with set-valued outputs, here on an IR for claim verification task. The number of considered articles is reduced at every level—red frames are filtered, while green frames pass on. We only care about retrieving \textit{at least one} of the admissible articles (starred) for resolving the claim.

“correct” one is not necessarily unique, classifiers derived using conformal prediction can suffer dramatically from both poor predictive and computational efficiency \cite{30, 293, 295}. Unfortunately, these two conditions tend to be compounding: large label spaces \( \mathcal{Y} \) both (1) often place strict constraints on the set of tractable model classes available for consideration, and (2) frequently contain multiple clusters of labels that are difficult to discriminate between, especially for a low-capacity classifier.

In this chapter, we present two effective methods for improving the efficiency of conformal prediction for classification tasks with large output spaces \( \mathcal{Y} \), in which several \( y \in \mathcal{Y} \) might be admissible—i.e., acceptable for the purposes of our given task. First, in Section 5.4 we describe a generalization of Eq. 5.1 to an expanded admission criteria, where \( \mathcal{C}_n(X_{n+1}) \) is considered valid if it contains at least one admissible \( y \) with high probability. For example, in our IR setting, given the claim “\textit{Michael Collins took part in the Apollo mission to the moon},” any of the articles “\textit{Apollo 11},” “\textit{Michael Collins (astronaut)},” or “\textit{Apollo 11 (2019 film)}” have enough information to independently support it (see Figure 5-1)—and are therefore all admissible. When \( Y_{n+1} \) is not unique, forcing the classifier to hedge for the worst case, in which a specific realization of \( Y_{n+1} \) must be contained in \( \mathcal{C}_n(X_{n+1}) \), is too strict and can lead to conservative predictions. We theoretically and empirically show that optimizing for an expanded admission criteria yields classifiers with significantly better predictive efficiency.

Second, in Section 5.5 we present a technique for conformalizing \textit{prediction cascades}...
to progressively filter the number of candidates with a sequence of increasingly complex classifiers. This allows us to balance predictive efficiency with computational efficiency during inference. Importantly, we also theoretically show that, in contrast to other similarly motivated pipelines, our method filters the output space in a manner that still guarantees marginal coverage. Figure 5-1 illustrates our combined approach. We demonstrate that, together, these two approaches serve as complementary pieces of the puzzle towards making CP more efficient. We empirically validate our approach on information retrieval for fact verification, open-domain question answering, and in-silico screening for drug discovery.

**Contributions.** In summary, our main results are as follows:

- A theoretical extension of validity (Eq. 5.1) to allow for inferred admissible answers.
- A principled framework for conformalizing computationally efficient prediction cascades.
- Consistent empirical gains on three diverse tasks demonstrating up to 4.6× better predictive efficiency AUC (measured across all \( \epsilon \)) when calibrating for expanded admission, with computation pruning factors of up to \( 1/m \), where \( m \) is the number of models, when using prediction cascades.

### 5.2 Related Work

#### 5.2.1 Uncertainty Estimates

Methods for obtaining precise uncertainty estimates have received intense interest in recent years. A significant body of work is concerned with calibrating model confidence—measured as \( p_\theta(\hat{y}_{n+1}|x_{n+1}) \)—such that the true accuracy, \( y_{n+1} = \hat{y}_{n+1} \), is indeed equal to the estimated probability \([189, 76, 145, 151]\). In theory, these estimates could be leveraged to create confident prediction sets \( C_n(X_{n+1}) \). Unlike CP, however, these methods are not guaranteed to be accurate, and often still suffer from miscalibration in practice—especially for modern neural networks \([91, 7, 107]\). **Selective classification** \([62, 80]\), where models have the option to abstain from answering when not confident, is similar in motivation to
Eq. [5.1] In fact, it can be considered as a special case in which the classifier chooses to abstain unless $|C_n(X_{n+1})| = 1$.

### 5.2.2 Conformal Prediction

As validity is already guaranteed by design in conformal prediction, most efforts in CP focus on improving various aspects of efficiency. Mondrian CP [294] accounts for the fact that some classes are harder to model than others, and leverages class-conditional statistics. Similarly, several recent studies have built towards conditional—as opposed to marginal—coverage through various adaptive approaches, such as conformalizing quantile functions or working with conditional distributions that vary with $x$ [see 36, 48, 138, 230, 229, inter alia]. [36] also directly model dependencies among $y$ variables for use in multi-label prediction.

Our method for expanded admission, on the other hand, aggregates statistics for equivalent single labels by example and across classes. Though we only provide marginal guarantees, the ideas expressed in those related works are complementary, and can be applied here as well. Inductive CP [196] is also complementary extension that dramatically reduces the cost of computing $C_n(X_{n+1})$ in the general case; we make use of it here. Most similar to our work, trimmed [46] and discretized [45] CP trade predictive efficiency for computational efficiency in regression tasks, where the label space is infinite. A key distinction of our method is that we do not force the same trade-off: in fact, we empirically show that our conformalized cascades can at times result in better predictive efficiency alongside a pruned label space.

### 5.2.3 Prediction Cascades

The idea of balancing cost with accuracy by using multi-step inference has been explored extensively for many applications [38, 54, 74, 125, 159, 233]. Some of these methods use fixed rules with no performance guarantees, such as greedy pipelines where the top $k$ predictions are passed on to the next level [39, 70]. Closer to our work, [301] optimize their cascades for overall pruning efficiency, and not for top-1 prediction. While they also analyze error bounds for filtering, their guarantees are specific to linear classifiers with
bounded $L_2$ norm, whereas our conformalized approach only assumes data exchangeability. Furthermore, they assume a target filtering loss before training—our tolerance level $\epsilon$ is defined at test time, which allows for much greater flexibility.

5.3 Background

We begin with a brief review of conformal prediction [see 251]. Here, and in the rest of the chapter, upper-case letters ($X$) denote random variables; lower-case letters ($x$) denote scalars, and script letters ($\mathcal{X}$) denote sets, unless otherwise specified. Proofs are deferred to the appendix.

At the core of conformal prediction is a simple statistical hypothesis test: for each candidate $y \in \mathcal{Y}$ we must either accept or reject the null hypothesis that $(X_{n+1} = x_{n+1}, Y_{n+1} = y)$ is a correct pairing. Formally, we rely on a nonconformity measure $S((x_{n+1}, y), \mathcal{D})$ to serve as the test statistic, where a higher value of $S$ reflects that $(x_{n+1}, y)$ is less “conforming” to the distribution specified by dataset $\mathcal{D}$. For instance, $S$ could be computed as $-\log p_\theta(y|x)$, where $\theta$ is a model fit to $\mathcal{D}$.

**Definition 5.3.1** (Nonconformity measure). Let $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ be the space of examples $(X, Y)$, and let $\mathcal{Z}^{(\ast)} := \bigcup_{d \geq 1} (\mathcal{X} \times \mathcal{Y})^d$ be the space of datasets of examples $\mathcal{D}$, of any size $d \geq 1$. A nonconformity measure $S$ is then a measurable mapping $S: \mathcal{Z} \times \mathcal{Z}^{(\ast)} \rightarrow \mathbb{R}$, that assigns a real-valued score to any example $(X, Y)$, indicating how different it is from a reference dataset $\mathcal{D}$. Furthermore, in order to retain exchangeability, $S$ is symmetric with respect to permutations of its input data.

To be specific, exact or full CP takes $\mathcal{D}$ to be all of the examples seen so far, including the candidate $(x_{n+1}, y)$. Thus, the nonconformity measure $S$ has to be re-trained each time. An alternative—which we use in this chapter w.l.o.g. —is the inductive or split CP variant [196] which assumes that $\mathcal{D}$ is a proper training set, independent of any of the subsequent $n + 1$ exchangeable examples used for CP. Dropping $\mathcal{D}$ for ease of notation, we denote the score for example $(X, Y)$ as the random variable $S(X, Y)$. The degree of nonconformity can then be quantified using a p-value.

---

1The definition of “different” here is intentionally vague, as any metric will technically work.
Lemma 5.3.2 (Smoothed p-value). Assume that the random variables $V_1, \ldots, V_{n+1}$ are exchangeable. We define the smoothed empirical p-value $\text{pvalue}(V_{n+1}, V_1:n)$ as

$$\text{pvalue}(V_{n+1}, V_1:n) := \frac{|\{i \in [1, n]: V_i > V_{n+1}\}| + \tau \cdot |\{i \in [1, n]: V_i = V_{n+1}\}| + 1}{n + 1},$$

where $\tau \sim U(0, 1)$. Then, for any $\epsilon \in (0, 1)$, we have $P(\text{pvalue}(V_{n+1}, V_1:n) \leq \epsilon) \leq \epsilon$.

To construct the final conformal prediction, the classifier uses the p-values to include all $y$ for which the null hypothesis—i.e., that the candidate pair $(x_{n+1}, y)$ is conformal—is not rejected.

Theorem 5.3.3 (CP; [294], see also [153]). Assume that the random variables $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n+1$ are exchangeable. For any nonconformity measure $S$, and $\epsilon \in (0, 1)$, define the conformal label set (based on the first $n$ samples) at $x_{n+1} \in \mathcal{X}$ as

$$\mathcal{C}_n(x_{n+1}) := \{y \in \mathcal{Y}: \text{pvalue}(S(x_{n+1}, y), S(x_{1:n}, y_{1:n})) > \epsilon\}.$$  

Then $\mathcal{C}_n(X_{n+1})$ satisfies Eq. 5.1 where $P(Y_{n+1} \in \mathcal{C}_n(X_{n+1})) \geq 1 - \epsilon$.

5.4 Conformal Prediction with Expanded Admission

We now introduce our strategy for improving the predictive efficiency of CP classifiers. What might it mean for an alternative label $y$ to be “good enough?” Among other factors, this depends on the task, the label space $\mathcal{Y}$, and even the input $x$. For example, in IR, two different texts might independently provide sufficient information for claim $x$ to be resolved. Formally, we pose the underlying setting as a set-valued function $f : \mathcal{X} \rightarrow 2^\mathcal{Y}$, where the ground truth is defined as the expanded set of all admissible answers $f(X)$ for input $X$ (e.g., given our notions of semantic equivalence or error tolerance). Unlike ranking or multi-label classification, however, our evaluation only demands retrieving a single element of this ground truth set (and without any preference as to which element).

It is often the case that this underlying function $f$ remains unknown—after all, exhaustively annotating all possible admissible answers can quickly become intractable for many problems. Rather, we assume that what we observe in our dataset are samples, $(X_i, Y_i)$,
from the underlying ground truth sets \( f(X_i) \) via some additional observation process (e.g., influenced by which annotator wrote the answer to the question). In this view, the distribution \( P \) governing each pair \((X_i, Y_i)\) is an induced distribution from this set-valued function, together with the unknown observation process. We can then use the provided dataset reference \( Y_i \) to seed a label-expansion operation, in an attempt to approximate \( f(X_i) \). More concretely, for some choice of admission function \( g: (\mathcal{X} \times \mathcal{Y}) \times \mathcal{Y} \to \{0, 1\} \), we construct a set of inferred admissible labels \( \mathcal{A}_g \) given the seed reference \((X = x, Y = y)\), i.e.,

\[
\mathcal{A}_g(x, y) := \left\{ \bar{y} \in \mathcal{Y} : g(x, y, \bar{y}) = 1 \right\}.
\] (5.4)

**Assumption 5.4.1.** For any reference label \( Y \in f(X) \), we have that \( \mathcal{A}_g(X, Y) \) contains \( Y \) and is a subset of the full ground truth. That is, \( \mathcal{A}_g(X, Y) \) obeys \( Y \in \mathcal{A}_g(X, Y) \subseteq f(X) \).

In this work we assume that \( g \) is given to us (not learned), and is a deterministic function that has no inherent error in its outputs. For many tasks, this is a quite natural assumption, as it is often feasible for the user to define a set of rules that qualify label admission—e.g., syntactic normalization rules in NLP, or expanding some small \( \delta \)-neighborhood of the original \( y \) given some metric.

Given \( g, x, \) and \( y \), any prediction that is a member of the derived set \( \mathcal{A}_g \) is then considered to be admissible, i.e., a success. For example, in IR for claim verification, let \( h \) be the verification model (or a human) that, given the claim \( x \) and evidence \( y \), outputs a score for the final verdict (that the claim is true or false). The admission might then be defined as \( g(x, y, \bar{y}) := 1\{ |h(y, x) - h(\bar{y}, x)| \leq \delta \} \), where \( \delta \) is a small slack parameter. This then leads us to a helpful definition of expanded admission:

**Definition 5.4.2** (Expanded admission). Given any label admission function \( g \) and data points \( \{(X_i, Y_i)\}_{i=1}^n \) drawn exchangeably from an underlying distribution \( P \), a conformal predictor producing admissible predictions \( \mathcal{C}_n(X_{n+1}) \) (based on the first \( n \) points) for a new exchangeable test example \( X_{n+1} \) is considered to be valid under expanded admission if for any \( \epsilon \in (0, 1) \), \( \mathcal{C}_n \) satisfies

\[
P\left( \left| \mathcal{A}_g(X_{n+1}, Y_{n+1}) \cap \mathcal{C}_n(X_{n+1}) \right| \geq 1 \right) \geq 1 - \epsilon; \quad \text{for all distributions } P. \] (5.5)
Recall that by predictive efficiency we mean that we desire $|C_n(X_{n+1})|$ to be small. A CP that simply returns all possible labels is trivially valid, but not useful. By Definition 5.4.2 we only need to identify at least one $\bar{Y}_{n+1}$ that is deemed admissible according to $g$. As such, we propose a modification that allows the classifier to be more discriminative—and produce smaller $C_n(X_{n+1})$—when testing the null hypothesis that $y$ is not just conforming, but that it is the most conforming admissible $y$ for $x$. For each data point $(X_i = x_i, Y_i = y_i)$, we then define the minimal nonconformity score as

$$S_{g\min}^\text{min}(x_i, y_i) := \min \left\{ S(x_i, \bar{y}) : \bar{y} \in \mathcal{A}_g(x_i, y_i) \right\},$$  

and use these random variables to compute the p-values for our conformal predictor.

**Theorem 5.4.3 (CP with expanded admission).** Assume that $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, $i = 1, \ldots, n+1$ are exchangeable. For any non-conformity measure $S$, label admission function $g$, and $\epsilon \in (0, 1)$, define the conformal set (based on the first $n$ samples) at $x_{n+1} \in \mathcal{X}$ as

$$C_{g\min}^\text{min}(x_{n+1}) := \left\{ y \in \mathcal{Y} : \text{pvalue}(S(x_{n+1}, y), S_{g\min}^\text{min}(x_{1:n}, y_{1:n})) > \epsilon \right\}.$$  

Then $C_{g\min}^\text{min}(X_{n+1})$ satisfies Eq. 5.5. Furthermore, we have $\mathbb{E} \left[ |C_{g\min}^\text{min}(X_{n+1})| \right] \leq \mathbb{E} \left[ |C_n(X_{n+1})| \right]$.

In Section 5.7 we demonstrate empirically that this simple modification can yield large improvements in efficiency, while still maintaining the desired coverage.

## 5.5 Conformal Prediction Cascades

We now introduce our strategy for improving the computational efficiency of conformal prediction. As is apparent from Eq. 5.3, the cost of conformal prediction is linear in $|\mathcal{Y}|$. In practice, this can limit the tractable choices available for the nonconformity measure $S$, particularly in domains where $|\mathcal{Y}|$ is large. Furthermore, predictive and computational efficiency are coupled, as being forced to use weaker $S$ reduces the statistical power of the CP. Our approach balances the two by leveraging prediction cascades [237, 301] inter alia], where $m$ models of increasing power are applied sequentially. At each stage, the number of considered outputs is iteratively pruned. Critically, we conformalize the cascade, which
Algorithm 1  Cascaded inductive conformal prediction with distribution-free marginal coverage.

Definitions: \((S_1, \ldots, S_m)\) is a sequence of nonconformity measures. \(\mathcal{M}\) is a monotonic correction controlling for family-wise error. \(x_{n+1} \in \mathcal{X}\) is the given test point. \(x_{1:n} \in \mathcal{X}^n\) and \(y_{1:n} \in \mathcal{Y}^n\) are the previously observed calibration examples and their labels, respectively. \(\mathcal{Y}\) is the label space. \(\epsilon\) is the tolerance level.

1: function \textsc{predict}(\(x_{n+1}, (x_{1:n}, y_{1:n}), \epsilon\))
2: \(\mathcal{C}_n^0 \leftarrow \mathcal{Y}\) \hfill \triangleright Initialize with the full label set.
3: \(p_1(y) = p_2(y) = \ldots = p_m(y) \leftarrow 1, \forall y \in \mathcal{Y}\) \hfill \triangleright Conservatively set unknown p-values.
4: for \(j = 1\) to \(m\) do
5: \(\mathcal{C}_n^j \leftarrow \{\}\) \hfill \triangleright Initialize the current output.
6: for \(y \in \mathcal{C}_n^{j-1}\) do \hfill \triangleright Iterate through the previous label set.
7: \(p_j(y) \leftarrow \text{pvalue}(S_j(x_{n+1}, y), S_j(x_{1:n}, y_{1:n}))\) \hfill \triangleright Update the \(j\)-th p-value for \((x_{n+1}, y)\).
8: \(\tilde{p}_j(y) \leftarrow \mathcal{M}(p_1(y), \ldots, p_m(y))\) \hfill \triangleright Correct the current p-values for MHT.
9: if \((\tilde{p}_j(y) > \epsilon)\) then \hfill \triangleright Keep \(y\) iff the corrected p-value supports it.
10: \(\mathcal{C}_n^j \leftarrow \mathcal{C}_n^j \cup \{y\}\)
11: return \(\mathcal{C}_n^m\) \hfill \triangleright Return the final output of the cascade.

preserves marginal coverage. For clarity of presentation, in this section we return to the standard setting without expanded admission (i.e., where \(\mathcal{C}_n(X_{n+1})\) satisfies Eq. [5.1]), but emphasize that the method applies to either case.

When constructing \(\mathcal{C}_n(X_{n+1})\) via Eq. [5.3] a nonconformity score and corresponding p-value is computed for every candidate \(y \in \mathcal{Y}\). Different \(y\), however, might be much easier to reject than others, and can be filtered using simple metrics. For example, in IR, wholly non-topical sentences (of which there are many) can be discarded using fast keyword matching algorithms such as TFIDF or BM25. On the other hand, more ambiguous sentences—perhaps those on the same topic but with insufficient information—might require a more sophisticated scoring mechanism, such as a neural network.

Assume that we are given a sequence of progressively more discriminative, yet also more computationally expensive, nonconformity measures \((S_1, \ldots, S_m)\). When applied in order, we only consider \(y \in \mathcal{C}_n^i(X_{n+1})\) as candidates for inclusion in \(\mathcal{C}_n^{i+1}(X_{n+1})\). Thus, \(\mathcal{C}_n^m(X_{n+1}) \subseteq \mathcal{C}_n^{m-1}(X_{n+1}) \subseteq \ldots \subseteq \mathcal{C}_n^1(X_{n+1}) \subseteq \mathcal{Y}\). In this way, the amortized cost of evaluating \(m\) measures over parts of \(\mathcal{Y}\) can be lower than the cost of running one expensive measure over all of it. For example, in IR, we can use BM25 \((S_1)\) to prune the label space passed to a neural model \((S_2)\). Furthermore, combining multiple nonconformity
measures together can also lead to better predictive efficiency when using complementary measures—similar to ensembling [278].

Naïvely applying multiple tests to the same data, however, leads to the multiple hypothesis testing (MHT) problem. This results in an increased family-wise error rate (i.e., false positives), making the CP invalid. Many corrective procedures exist in the literature (e.g., see [163]). Formally, given \( m \) p-values \((P_1, \ldots, P_m)\) for a pair \((X, Y)\), we denote as \( \mathcal{M} \) some such correction satisfying

\[
\tilde{P} = \mathcal{M}(P_1, \ldots, P_m) \quad \text{s.t.} \quad \mathbb{P}\left(\tilde{P} \leq \epsilon \mid Y \text{ is correct}\right) \leq \epsilon. \tag{5.8}
\]

Furthermore, we require \( \mathcal{M} \) to be element-wise monotonic\(^2\) i.e. (where \( \preceq \) operates element-wise):

\[
(P_1, \ldots, P_m) \preceq \left(\hat{P}_1, \ldots, \hat{P}_m\right) \implies \mathcal{M}(P_1, \ldots, P_m) \leq \mathcal{M}(\hat{P}_1, \ldots, \hat{P}_m). \tag{5.9}
\]

We consider several options for \( \mathcal{M} \), namely the Bonferroni and Simes corrections (see Appendix B.3). In order to exit the test early at cascade \( j \) before all the p-values (i.e., for measures \( k > j \)) are known, we compute an upper bound for the corrected p-value by conservatively assuming that \( P_k = 1 \), \( \forall k > j \). The full procedure is demonstrated in Algorithm 1 and formalized in Theorem 5.5.1.

**Theorem 5.5.1 (Cascaded CP).** Assume that \((X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n + 1\) are exchangeable. For any sequence of nonconformity measures \((S_1, \ldots, S_m)\) yielding p-values \((P_1, \ldots, P_m)\), and \( \epsilon \in (0, 1) \), define the conformal set for step \( j \) (based on the first \( n \) samples) at \( x_{n+1} \in \mathcal{X} \) as

\[
C^j_n(x_{n+1}) := \left\{ y \in \mathcal{Y} : \tilde{P}_j^{(y)} > \epsilon \right\}, \tag{5.10}
\]

where \( \tilde{P}_j^{(y)} \) is the conservative p-value for candidate \( y \) at step \( j \), \( \mathcal{M}(P_1^{(y)}, \ldots, P_j^{(y)}, 1, \ldots, 1) \), with \( P_k^{(y)} := 1 \). Then \( \forall j \in [1, m] \), \( C^j_n(x_{n+1}) \) satisfies Eq. 5.1 and \( C^m_n(x_{n+1}) \subseteq C^j_n(x_{n+1}) \).

Theorem 5.5.1 also easily extends to the setting of Eq. 5.5. An important result is that early pruning will not affect the validity of the final result, \( C^m_n \).

\(^2\)Note that we are unaware of any common \( \mathcal{M} \) beyond contrived examples satisfying (5.8) but not (5.9).
5.6 Experimental Setup

We empirically evaluate our method on three different tasks with standard, publicly available datasets. In this section, we briefly give a high-level outline of each task and our conformalized approach to it. We also describe our evaluation methodology. We defer the technical details for each task, such as data preprocessing, training, and nonconformity measure formulations, to Appendix B.1.

5.6.1 Tasks

**Open-domain question answering (QA).** Open-domain question answering focuses on using a large-scale corpus $\mathcal{D}$ to answer arbitrary questions via search combined with reading comprehension. We use the open-domain setting of the Natural Questions dataset [144]. Following [39], we first retrieve relevant passages from Wikipedia using a document retriever, and then select an answer span from the considered passages using a document reader. We use a Dense Passage Retriever model [130] for the retriever, and a BERT model [56] for the reader. The BERT model yields several score variants—we use multiple in our cascade (see Table B.1.1). Any span from any retrieved passage that matches any of the annotated answer strings when lower-case and stripped of articles and punctuation is considered to be correct.

**Information retrieval for fact verification (IR).** As introduced in §5.1, the goal of IR for fact verification is to retrieve a sentence that can be used to support or refute a given claim. We use the FEVER dataset [272], in which evidence is sourced from a set of $\sim$40K sentences collected from Wikipedia. A sentence that provides enough evidence for the correct verdict (true/false) is considered to be acceptable (multiple are labeled in the dataset). Our cascade consists of (1) a fast, non-neural BM25 similarity score between a given claim and sentence, and (2) the score of an ALBERT model [147] trained to directly predict if a given claim and sentence are related.

**In-silico screening for drug discovery (DR).** In-silico screening of chemical compounds is a common task in drug discovery/repurposing, where the goal is to identify possibly effective drugs to manufacture and test [263]. Using the ChEMBL database [177],
we consider the task of screening molecules for combinatorial constraint satisfaction, where
given a specified constraint such as “has property A but not property B”, we want to iden-
tify at least one molecule from a given set of candidates that has the desired attributes.
Our cascade consists of (1) the score of a fast, non-neural Random Forest (RF) applied
to binary Morgan fingerprints [226], and (2) the score of a directed Message Passing NN
ensemble [313].

5.6.2 Evaluation Metrics

For each task, we use a proper training, validation, and test set. We use the training set to
learn all nonconformity measures $S$. We perform model selection specifically for CP on
the validation set, and report final numbers on the test set. For all CP methods, we report
the marginalized results over 20 random trials, where in each trial we partition the data into
80% calibration ($x_{1:n}$) and 20% prediction points ($x_{n+1}$). In order to compare the aggregate
performance of different CPs across all tolerance levels, we plot each metric as a function
of $\epsilon$, and compute the area under the curve (AUC). In all plots, shaded regions show the
16-84th percentiles across trials. We use the following metrics:

Predictive accuracy. We measure the accuracy rate as the rate at which at least one
admissible prediction is in $C_n$, i.e., $|A_g(X_{n+1}, Y_{n+1}) \cap C_n(X_{n+1})| \geq 1$ (see Eq. 5.5). To be
valid—the key criteria in this work—a classifier should have an accuracy rate $\geq 1 - \epsilon$, and
AUC $\geq 0.5$. Note that more is not necessarily better: higher success rates than required can
lead to poor efficiency (i.e., the size of $C_n$ can afford to decrease at the expense of accuracy).

Predictive efficiency ($\downarrow$). We measure predictive efficiency as the size of the prediction
set out of all candidates: $|C_n| \cdot |Y|^{-1}$. The goal is to make the predictions more precise while
still maintaining validity. Lower predictive efficiency is better ($\downarrow$), as it means that the size
of $C_n$ is relatively smaller.

Amortized computation cost ($\downarrow$). We measure the amortized computation cost as the ratio of pvalue computations required to make a cascaded prediction with early pruning,
compared to when using a simple combination of CPs (no pruning, same number of mea-
sures). In this work we do not measure wall-clock times as these are hardware-specific, and
depend heavily on optimized implementations. Lower amortized cost is better, as it means
Figure 5-2: Does calibrating using minimal admissible nonconformity scores (§5.4) improve predictive efficiency? We show predictive efficiency and success rates as a function of $\epsilon$. The efficiency of the $\min$-calibrated CPs (both cascaded and non-cascaded) is significantly better (i.e., lower) than standard CP across all tasks, especially at critical values of $\epsilon$. The $\min$-calibrated CP’s accuracy hugs the diagonal, allowing it to remain valid, but be far less conservative (resulting in better efficiency).

that the relative number of p-value computations required to construct $C^m_m$ (for a given $m$) is smaller.

5.7 Experimental Results

In the following, we address several key research questions relating to our two combined conformal prediction advancements, and their impact on overall performance. In all of the following experiments, we report results on the test set, using cascade configurations selected based on the validation set performance. The QA and IR cascades use the Simes correction for MHT, while the DR cascades uses the Bonferroni correction. Additional results, details, and analysis are included in Appendix B.2.

Predictive efficiency of expanded admission. We begin by testing how deliberately calibrating for expanded admission affects predictive efficiency. That is, we use minimal admissible nonconformity scores $\{S_{g}^{\min}(X_i, Y_i)\}_{i=1}^{n}$ for calibration (Eq. 5.6) to create
Table 5.1: CP results for specific tolerance levels $\epsilon$. Each line shows the empirical accuracy (Acc.) and the (raw) size of the prediction set for our two methods as compared to regular CP, per target accuracy rate $(1 - \epsilon)$. The amortized computation cost of the cascade, a consequence of pruning, is also given.

$C_n(X_{n+1})$ using Eq. 5.7. Figure 5-2 shows the predictive efficiency of our min-calibrated CPs with expanded admission across all $\epsilon$, while Table 5.1 shows results for select values of $\epsilon$. We compare both cascaded and non-cascaded CPs, as Theorem 5.4.3 applies to both settings. The non-cascaded CP uses the last nonconformity measure of the cascaded CP. Across all tasks, the efficiency of the min-calibrated CP is significantly better than the baseline CP method, and results in tighter prediction sets—giving up to $4.6 \times$ smaller AUC. Naturally, this effect depends on the qualities of the admission function $g$ and resulting admissible label sets $A_g$. For example, the most dramatic gains are seen on QA. This task has a relatively large variance among admissible label nonconformity scores (i.e., some admissible answer spans are much easier to identify than others), and thus calibrating for the most conforming spans has a large impact.

Validity of minimal nonconformity calibration. As per Theorem 5.4.3, we observe that our min-calibrated CP still creates valid predictors, with an accuracy rate close to $1 - \epsilon$ on average. We see that the min-calibration allows the predictor to reject more wrong predictions (lower predictive efficiency, which is better) while, with high enough probability, still accept at least one that is correct. The standard CP methods, however, are more conservative—and result in prediction sets and accuracy rates larger than necessary.

<table>
<thead>
<tr>
<th>Task</th>
<th>Target Acc. (1 − $\epsilon$)</th>
<th>Baseline CP</th>
<th>min-CP</th>
<th>Cascaded min-CP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc. $</td>
<td>C_n</td>
<td>$</td>
<td>Acc. $</td>
</tr>
<tr>
<td>QA</td>
<td>0.90</td>
<td>0.98</td>
<td>1245.7</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>0.94</td>
<td>453.5</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>0.91</td>
<td>227.8</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>0.87</td>
<td>127.8</td>
<td>0.60</td>
</tr>
<tr>
<td>IR</td>
<td>0.99</td>
<td>1.00</td>
<td>33.6</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>1.00</td>
<td>20.9</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.98</td>
<td>13.8</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>0.95</td>
<td>6.7</td>
<td>0.80</td>
</tr>
<tr>
<td>DR</td>
<td>0.90</td>
<td>0.99</td>
<td>429.8</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>0.97</td>
<td>305.8</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>0.96</td>
<td>216.6</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>0.94</td>
<td>145.6</td>
<td>0.60</td>
</tr>
</tbody>
</table>
Figure 5-3: How effective is cascaded conformal prediction (§5.5) at pruning the candidate space? Incorporating early rejection via the CP cascade reduces the fraction of required p-value computations. Larger tolerance levels ($\epsilon$) allow for more aggressive pruning rates. This effect is most pronounced at smaller $\epsilon$ (e.g., $< 0.2$).

**Computational efficiency of conformalized cascades.** Figure 5-3 shows the amortized cost, in terms of percentage of p-value computations skipped, achieved using our cascaded CP algorithm. Our method reduces the number of p-value computations by up to a factor of $1/m$ for an $m$-layer cascade. The effect of conformalization is clear: the more strict an $\epsilon$ we demand, the fewer labels we can prune, and vice versa. To simplify comparisons, our metric gives equal weight to all p-value computations. In practice, however, the benefits of early pruning will generally grow by layer, as the later p-values are assumed to be increasingly expensive to compute. Again, exactly quantifying this trade-off in terms of absolute cost is model- and implementation-specific; we leave it for future work.

**Conservativeness of conformalized cascades.** A limitation of the cascade approach is the conservative nature of the MHT-corrected p-value (i.e., the Bonferroni effect), which can reduce the statistical power of the CP as the number of cascade layers grows. This effect is especially present if the cascaded measures are highly dependent. In general, however, in both Figure 5-2 and Table 5.1 we see that the benefits of combining complementary cascaded models largely make up for this drop in statistical power, as our cascaded min-calibrated CPs nearly matches the predictive efficiency of our non-cascaded models. Importantly, this is achieved while still improving computational efficiency.

**Relation to heuristic methods.** For completeness, in Appendix B.2.2 we also compare CP to common heuristic methods for producing set-valued predictions—namely, taking the top-$k$ predictions and taking all predictions for which the model’s score exceeds a
threshold \( \tau \). We show that while CP is more general, it can be (practically) reduced to each of these methods with the appropriate choice of nonconformity measure. In some cases, the flexibility of CP even allows for better predictive efficiency, even while those heuristics do not amortize cost or guarantee coverage in finite samples.

5.8 Conclusion

Conformal prediction can afford remarkable theoretical performance guarantees to important applications for which high accuracy and precise confidence estimates are key. Naively applying CP, however, can be inefficient in practice. This is especially true in realistic domains in which the correct answers are not clearly delineated, and in which the computational cost of discriminating between options starts to become a limiting factor. In this chapter, we proposed two novel methods that provide two more pieces of the puzzle. Our results show that (1) calibration using expanded admission consistently improves empirical predictive efficiency, and (2) conformal prediction cascades yield better computational efficiency—and thereby enable the use of more powerful classifiers.
Chapter 6

Efficient Classification via Confident Adaptive Transformers

We develop a novel approach for confidently accelerating inference in the large and expensive multilayer Transformers that are now ubiquitous in natural language processing (NLP). Amortized or approximate computational methods increase efficiency, but can come with unpredictable performance costs. In this chapter, we present CATs—Confident Adaptive Transformers—in which we simultaneously increase computational efficiency, while guaranteeing a specifiable degree of consistency with the original model with high confidence. Our method trains additional prediction heads on top of intermediate layers, and dynamically decides when to stop allocating computational effort to each input using a meta consistency classifier. To calibrate our early prediction stopping rule, we formulate a unique extension of conformal prediction. We demonstrate the effectiveness of this approach on four classification and regression tasks.

6.1 Introduction

Large pre-trained language models have become the de facto standard approach for solving natural language processing tasks [56, 167]. Despite their impressive performance, however, their often massive computational burden makes them costly to run [247, 254]. Concerns about their efficiency have kindled a large body of research in the field [236, 248, 65]. For
multilayered architectures such as the Transformer, a popular approach is adaptive early exiting [248, 308, inter alia]. Early exiting takes advantage of the observation that task instances vary in complexity. In this setting, “early” classifiers are added on top of the simpler features of intermediate layers in the base model, and can trigger a prediction before the full model is executed. Naively deciding when to preempt computation, however, can result in unpredictable decreases in model accuracy.

Quantifying the uncertainty in a prediction in order to decide when additional computation is needed (or not) is critical to making predictions quickly without excessively sacrificing performance. In this chapter, we present Confident Adaptive Transformers (CATs), a general method for increasing Transformer-based model efficiency while remaining confident in the quality of our predictions. Specifically, given a fixed, expensive $l$-layer model $\mathcal{F}(x)$, we create an amortized model $\mathcal{G}(x)$ that includes early classifiers $\{\mathcal{F}_1, \ldots, \mathcal{F}_l\}$\[^1\] We then make $\mathcal{G}$ provably consistent with the original $\mathcal{F}$ with arbitrarily high probability (e.g., 95% of the time). This process is illustrated in Figure 6-1.

Our approach builds on conformal prediction (CP), a model-agnostic and distribution-
**Claim**: All airports in Guyana were closed for all international passenger flights until 1 May 2020.

**Evidence**: Airports in Guyana are closed to all international passenger flights until 1 May 2020.

**Claim**: Deng Chao broke sales record for a romantic drama.

**Evidence**: The film was a success and broke box office sales record for mainland-produced romance films.

Figure 6-2: Confidence levels given by our meta model regarding the consistency of our prediction as computation progresses. Ex.1 from the VitaminC fact verification dataset is “easy”, and is classified consistently by all early classifiers $F_k$ (Supports). The meta confidence captures this, and increases with time. Ex.2 is harder—and the prediction changes (Refutes/NEI) as it propagates through the Transformer layers. Appropriately, the meta confidence is low. The exact exit layer of $G$ is determined as a function of a user-specified tolerance $\epsilon$, see Eq. Equation 6.1.

free framework for creating well-calibrated predictions [294]. Concretely, suppose we have been given $n$ examples, $X_i \in \mathcal{X}, i = 1, \ldots, n$, as unlabeled calibration data, that have been drawn exchangeably from some underlying distribution $P$. Let $X_{n+1} \in \mathcal{X}$ be a new exchangeable test example for which we would like to make a prediction. The aim of our method is to construct $G$ such that it agrees with $F$ with distribution-free marginal coverage at a tolerance level $\epsilon \in (0, 1)$, i.e.,

$$\mathbb{P}(G(X_{n+1}) = F(X_{n+1})) \geq 1 - \epsilon. \quad (6.1)$$

We consider $G$ to be $\epsilon$-consistent if the frequency of error, $G(X_{n+1}) \neq F(X_{n+1})$, does not exceed $\epsilon$. By design, this ensures that $G$ preserves at least $(1 - \epsilon)$-fraction of $F$’s original performance. Within these constraints, the remaining challenge is to make $G$ relatively efficient (e.g., a consistent, but vacuous, model is simply the identity $G \triangleq F$).

\[2\]For regression, we define equality as $|G(\cdot) - F(\cdot)| \leq \tau$. 

137
In order to support an efficient $G$, we need a reliable signal for inferring whether or not the current prediction is likely to be stable or not. Past work [e.g., 248] rely on potentially poorly correlated metrics such as the early classifier’s softmax response. We address this challenge by instead directly learning meta “consistency predictors” for each of the $l - 1$ early classifiers of our $l$ layer model, by leveraging patterns in past predictions. Figure 6-2 demonstrates the progression of meta confidence scores across layers when applied to “easy” versus “hard” instances from the VitaminC fact verification task [242].

We pair the scores of our meta classifier for each layer with a stopping rule that is calibrated using a unique twist on standard conformal prediction. Traditionally, CP is used to construct prediction sets that cover the desired target (e.g., $Y_{n+1}$) with high probability. We invert the CP problem to first infer the multi-label set of inconsistent layers, and then exit at the first layer that falls in its complement. We then demonstrate that this can be reduced to setting a simple (but well-calibrated) exit threshold for the meta classifier scores. Our resulting algorithm is (1) fast to compute in parallel to the main Transformer, (2) requires only unlabeled data, and (3) is statistically efficient in practice, in the sense that it finds low exit layers on average while still maintaining the required predictive consistency.

We validate our method on four diverse NLP tasks—covering both classification and regression, different label space sizes, and varying amounts of training data. We find that it constitutes a simple-yet-effective approach to confident adaptive prediction with minimal interventions and desirable theoretical guarantees. In short, we provide:

1. A novel theoretical extension of conformal prediction to accommodate adaptive prediction;

2. An effective meta consistency classifier for deriving a confident “early exiting” model;

3. A demonstration of the utility of our framework on both classification and regression tasks, where we show significant efficiency improvements, while guaranteeing high consistency.

---

3We refer to the meta aspect of the classifier, not the optimization process (i.e., not to be confused with meta-learning).
6.2 Related Work

6.2.1 Adaptive Computation

Reducing the computational cost of neural models has received intense interest from the machine learning community. Consequently, many efficiency improvements have been proposed in recent years.

A popular approach is model compression—either by distillation [236], in which a smaller model is trained to imitate the larger one, or by model pruning [65, 180], in which parameters are removed from the original model. Both approaches result in a single model that is used for all future inputs.

Adaptive approaches adjust the amount of computation per example to amortize the total inference cost [266, 89, 112, 134]. As discussed in §6.1, our method is inspired by the approach of Schwartz et al. [248] and others [164, 81, 326], where they preempt computation if the softmax value of any early classifier is above a predefined threshold. Yet unlike our approach, their model is not guaranteed to be accurate, even after softmax calibration [92]. Several approaches to early exiting also include fine-tuning stages to improve efficiency [164, 81, 326]. Some methods attempt to preserve the top classifier’s performance while improving the accuracy of the early classifiers by applying a weighted loss [81]. In practice, however, the top performance still decreases for some tasks, and they do not provide any guarantee on the performance. In this work, we choose to avoid this step in order to allow our model to be widely applied with minimal overhead and to preserve the performance of the top classifier. Several methods use the softmax entropy as the confidence measure instead of the max value [309, 164], but we find it to be less accurate compared to the difference between the max value and second max for classification tasks.

Regression tasks lack a softmax layer and therefore are unsupported by the above softmax-based confidence estimates. In concurrent work, [310] propose a meta confidence classifier similar to ours. As in previous work, however, they do not address how to calibrate their model to guarantee a desired performance level.

In a recent work, the number of consecutive similar predictions by Transformer layers
was used as an exiting decision rule\textsuperscript{326}. For some tasks and settings, this approach was empirically observed to even provide slightly better accuracy than the full model. However, they do not provide performance guarantees, and in some tasks observe a decrease in accuracy. Also, for this model to properly work, all early classifiers should be trained jointly with the top one. As mentioned above, we assume that the large model—usually containing many parameters—was already trained and we only need to train our added early classifiers. Also, we include history features for our meta-classifier to also leverage the propagation information for our confidence estimates.

### 6.2.2 Confident Prediction

A large amount of research has been dedicated towards calibrating the model posterior, $p_\theta(\hat{y}_{n+1}|x_{n+1})$, such that the accuracy, $y_{n+1} = \hat{y}_{n+1}$, is indeed equal to the estimated probability [189, 76, 91]. In theory, these estimates could be leveraged to create confident early exits—e.g., similar to [248]. Ensuring calibrated probabilities of this form is hard, however, and existing methods often still suffer from miscalibration. Additionally, many methods exist for bounding the true error of a classifier [148, 199], but do not give end-users opportunities to control it. More similar to our work, selective classification [80] allows the model to abstain from answering when not confident, in order to maintain a target error rate only over answered inputs. Our work gives a different and statistically efficient technique applied to consistent prediction.

Alternatives such as Bayesian approaches can be useful and satisfy different desiderata [186, 88, 103, 76], but do not necessarily give well-calibrated probabilities (e.g., due to model misspecification, choice of prior, or approximation issues). Our CP-based method, on the other hand, gives well-calibrated distribution-free guarantees—even in finite samples.

### 6.2.3 Conformal Prediction for Multi-label Predictions

CP\textsuperscript{294} typically is formulated in terms of prediction sets $\mathcal{C}(X_{n+1})$, where finite-sample, distribution-free guarantees can be given over the event that $\mathcal{C}$ contains $Y_{n+1}$. As we discuss in \S 6.4, internally our method follows a similar approach in which we try to conservatively
identify the inadmissible set of all layers that are inconsistent (and exit at the first layer that falls in that set’s complement). Most relevant to our work, [36] presents algorithms for conformal multi-label predictions. We leverage similar methods in our model, but formulate our solution in terms of the complement of a multi-label set of inconsistent predictions. Our work adds to several recent directions that explore CP in the context of risk-mitigating applications [154, 229, 15, 71, inter alia], or meta-learning settings [72].

6.3 Early Exiting Transformers

In the following, we describe our dynamic early exiting model. We summarize early classification (following previous work) for convenience (§6.3.1), and then present our novel meta consistency classifier (§6.3.2). We focus on classification and regression tasks, given a model \( F(x) = y \). We assume that \( F \) maps the input \( x \in \mathcal{X} \) into a series of feature representations before making the prediction \( y \in \mathcal{Y} \). Here, \( F \) is a multilayered Transformer [288] composed of \( l \) layers (although our method can be applied to any multilayer network).

For all downstream tasks we follow standard practice and assume that the input contains a [CLS] token whose representation is used for prediction. For classification, we use a task-specific head, \( \text{softmax}(W_o(\phi(W_p h_{(CLS)}))) \), where \( h_{(CLS)} \in \mathbb{R}^d \) is the hidden representation of the [CLS] token, \( \phi \) is a nonlinear activation, and \( W_p \) are linear projections, where \( W_p \in \mathbb{R}^{d \times d} \) and \( W_o \in \mathbb{R}^{\left|\mathcal{Y}\right| \times d} \). Regression is treated similarly, but uses a 1-d output projection, \( w_o \cdot h_{(CLS)} \).

6.3.1 Early Predictors

\( F \)'s structure yields a sequence of hidden [CLS] representations, \( \{h_{(CLS)}^{(1)}, \ldots, h_{(CLS)}^{(l)}\} \), where \( h_{(CLS)}^{(k)} \in \mathbb{R}^d \) is the representation after applying layer \( k \). After each intermediate layer \( k < l \), we train an early classification head that is similar to the head used in \( F \), but reduce the dimensionality of the first projection to \( W_p^{(k)} \in \mathbb{R}^{d_c \times d} \) (this is purely for efficiency). The final \( F_l \) is unchanged from \( F \). These extra parameters \( ((l - 1) \times (d_c \times d + d_c \times |\mathcal{Y}|)) \) are quick to tune on top of a fixed \( F \), and we can reuse \( F \)'s training data as \( D_{tune} \).

\[4\] If \( D_{tune} \) is unlabeled, we can use \( F(x) \) as labels.
<table>
<thead>
<tr>
<th>Meta Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{y}_k$</td>
<td>The current prediction.</td>
</tr>
<tr>
<td>history</td>
<td>The past $k - 1$ predictions, $\hat{y}_{1:k-1}$.</td>
</tr>
<tr>
<td>$p^\text{max}_k$</td>
<td>Probability of the prediction, $p_k(\hat{y}</td>
</tr>
<tr>
<td>$p^\text{diff}_k$</td>
<td>Difference in probability of top predictions, $p_k(\hat{y}</td>
</tr>
</tbody>
</table>

Table 6.1: Additional meta features used as input to the meta early exit classifier, $\mathcal{M}_k$. Where specified, the probability $p_k$ is taken from the model’s early softmax. $p^\text{max}_k$ and $p^\text{diff}_k$ are only used for classification tasks.

classifier $\mathcal{F}_k(x) = \text{softmax}(W^{(k)}_p(\phi(W^{(k)}_p h^{(k)}_{\text{CLS}})))$ is then used after layer $k$ to get an early prediction candidate. Early regression is handled similarly.

### 6.3.2 Meta Early Exit Classifier

To decide when to accept the current prediction and stop computation, we require some signal as to how likely it is that $\mathcal{F}_k(x) = \mathcal{F}(x)$. Previous work relies on intrinsic measures (e.g., softmax response). Here, we present a meta classifier to explicitly estimate the consistency of an early predictor. Given fixed $\mathcal{F}_k$ and $\mathcal{F}$, we train a small binary MLP, $\mathcal{M}_k(x) \in \mathbb{R}$, on another unlabeled (limited) sample of task in-domain data, $\mathcal{D}_{\text{meta}}$. As input, we provide the current “early” hidden state $\phi(W^{(k)}_p h^{(k)}_{\text{CLS}})$, in addition to several processed meta features, see Table 6.1. We then train $\mathcal{M}_k$ with a binary cross entropy objective, where we maximize the likelihood of predicting $1\{\mathcal{F}_k(x_i) = \mathcal{F}(x_i)\}$ for $x_i \in \mathcal{D}_{\text{meta}}$.

Using the trained $\mathcal{F}_k$ and $\mathcal{M}_k$, we define the full adaptive model $\mathcal{G}$ using the prediction rule

$$
\mathcal{G}(x; \tau) := \begin{cases} 
\mathcal{F}_1(x) & \text{if } \mathcal{M}_1(x) > \tau_1, \\
\mathcal{F}_2(x) & \text{else if } \mathcal{M}_2(x) > \tau_2, \\
& \vdots \\
\mathcal{F}_l(x) & \text{otherwise,}
\end{cases}
$$

(6.2)

where $\tau = (\tau_1, \ldots, \tau_{l-1})$ are confidence thresholds. The key challenge is to calibrate $\tau_k$ such that $\mathcal{G}$ guarantees $\epsilon$-consistent performance per Eq. Equation 6.1.
6.3.3 Warmup: Development Set Calibration

A simple approach to setting $\tau$ is to optimize performance on a development set $\mathcal{D}_{\text{dev}}$, subject to a constraint on the empirical inconsistency:

$$\tau^* := \minimize_{(\tau_1, \ldots, \tau_{l-1})} \hat{E}_{\text{dev}}[\text{exit}(G(X; \tau))]$$

subject to

$$\hat{E}_{\text{dev}}[\{G(X; \tau) = F(X)\}] \geq 1 - \epsilon,$$

(6.3)

where $\text{exit}(\cdot)$ measures the exit layer, and $\hat{E}_{\text{dev}}$ is simply the average over $\mathcal{D}_{\text{dev}}$. Using a standard error bound [148] over a separate split, $\mathcal{D}_{\text{cal}}$, we can then derive the following guarantee:

**Proposition 6.3.1.** Let $X_i, i = 1, \ldots, n$ be an i.i.d. sample with $s = \sum_{i=1}^{n} 1\{G(X_i; \tau) = F(X_i)\}$. Then, up to a confidence level $\delta$, we have that

$$P(P(G(X; \tau) = F(X)) \geq 1 - \tilde{\epsilon}) \geq 1 - \delta,$$

(6.4)

where $\tilde{\epsilon}$ is the solution to $\text{Beta}(s, n - s + 1) = \delta$.

Though in practice $\tilde{\epsilon}$ might be close to $\epsilon$ for most well-behaved distributions, unfortunately Eq. Equation 6.4 does not give a fully specifiable guarantee as per Eq. Equation 6.1. Readjusting $\tau$ based on $\mathcal{D}_{\text{cal}}$ requires correcting for multiple testing in order to remain theoretically valid, which can quickly become statistically inefficient. In the next section, we provide a novel calibration approach that allows us to guarantee a target performance level with strong statistical efficiency.

6.4 Conformalized Early Exits

We now formulate the main contribution of this chapter, which is a distribution-free and model-agnostic method based on CP for guaranteeing any performance bound an end-user chooses to specify.^[251]

^[251]See [251] for a concise review of CP.
6.4.1 Conformal Formulation

Let \( \mathcal{I}(x) := \{ i : \mathcal{F}_i(x) \neq \mathcal{F}(x) \} \) be the index set of layers that are inconsistent with the final model’s prediction. To maintain \( \epsilon \)-consistency, we must avoid using any of the predictions specified by this set, \( \mathcal{F}_i(x) \) where \( i \in \mathcal{I}(x) \), more than \( \epsilon \)-fraction of the time for \( x \in \mathcal{X} \). In §6.4.2, we show how \( \mathcal{M}_{1:l-1} \) can be paired with a conformal procedure to obtain calibrated thresholds \( \boldsymbol{\tau} = (\tau_1, \ldots, \tau_{l-1}) \) such that we obtain a conservative prediction of \( \mathcal{I}(x) \),

\[
\mathcal{C}_n(x) := \{ k : \mathcal{M}_k(x) \leq \tau_k \},
\]

where we ensure that \( \mathcal{I}(x) \subseteq \mathcal{C}_n(x) \) with probability at least \( 1 - \epsilon \). Proposition 6.4.1 states our guarantee when \( \boldsymbol{\tau} \) is paired with \( \mathcal{G} \) following Eq. Equation 6.2.

**Proposition 6.4.1.** Assume that examples \( X_i, i = 1, \ldots, n + 1 \) are exchangeable. For any \( \epsilon \in (0, 1) \), let the index set \( \mathcal{C}_n \) (based on the first \( n \) examples) be the output of conformal procedure satisfying

\[
\mathbb{P}(\mathcal{I}(X_{n+1}) \subseteq \mathcal{C}_n(X_{n+1})) \geq 1 - \epsilon.
\]

Define \( K := \min \{ j : j \in \mathcal{C}_n(X_{n+1}) \} \), the first exit layer selected by \( \mathcal{G} \) following Eq. Equation 6.2. Then

\[
\mathbb{P}(\mathcal{F}_K(X_{n+1}) = \mathcal{F}(X_{n+1})) \geq 1 - \epsilon.
\]

**Remark 6.4.2.** Note that Eq. Equation 6.6 is stricter than necessary. Fundamentally, we only require that \( \mathbb{P}(K \in \mathcal{I}^c(X_{n+1})) \geq 1 - \epsilon \). Nevertheless, Eq. Equation 6.6 is easier to calibrate, and leads to strong empirical results despite being theoretically conservative.

**Remark 6.4.3.** During inference we do not fully construct \( \mathcal{C}_n \); it is only used to calibrate \( \boldsymbol{\tau} \) beforehand.

6.4.2 Conformal Calibration

We now describe our conformal procedures for calibrating \( \boldsymbol{\tau} \). Conformal prediction is based on hypothesis testing, where for a given input \( x \) and possible output \( y \), a statistical test is

\[\text{Here } A^c \text{ denotes the complement index set } \{ i : i \notin A \}.\]
performed to accept or reject the null hypothesis that the pairing \((x, y)\) is correct. In our setting, we consider the null hypothesis that layer \(k\) is inconsistent, and we use \(M_k(x)\) as our test statistic. Since \(M_k\) is trained to predict \(1\{F_k(x_i) = F(x_i)\}\), a high value of \(M_k(x)\) indicates how “surprised” we would be if layer \(k\) was in fact inconsistent with layer \(l\) for input \(x\). Informally, a low level of surprise indicates that the current input “conforms” to past data. To rigorously quantify the degree of conformity via the threshold \(\tau_k\) for predictor \(M_k\), we use a held-out set of \(n\) unlabeled, exchangeable examples, \(D_{\text{cal}}\).

**Independent calibration**

As a first approach, we construct \(C_n(x)\) by composing \(l - 1\) separate tests for \(F_k(x) \neq F(x)\), each with significance \(\alpha_k\), where \(\alpha_k\) are corrected for multiple testing. Let \(v_k^{(1:n,\infty)}\) denote the inflated empirical distribution of inconsistent layer scores,

\[
\{M_k(x_i): x_i \in D_{\text{cal}}, F_k(x_i) \neq F(x_i)\} \cup \{\infty\}.
\]

Inflating the empirical distribution is critical to our finite sample guarantee. We then define \(\tau_k^{\text{ind}} = \text{Quantile}(1 - \alpha_k, v_k^{(1:n,\infty)})\), and predict the inconsistent index set at \(x \in \mathcal{X}\) as

\[
C_e^{\text{ind}}(x) = \{k: M_k(x) \leq \tau_k^{\text{ind}}\}. \quad (6.8)
\]

The following theorem states how to set each \(\alpha_k\) such that the quantiles \(\tau_k^{\text{ind}}\) yield a valid \(C_n^{\text{ind}}\).

**Theorem 6.4.4.** Let \(\alpha_k = \omega_k \cdot \epsilon\), where \(\omega_k\) is a weighted Bonferroni correction, i.e., \(\sum_{k=1}^{l-1} \omega_k = 1\). Then \(C_n^{\text{ind}}(X_{n+1})\) is a valid set that satisfies Eq. Equation 6.6.

**Remark 6.4.5.** \(\omega_{1:l-1}\) can be tuned on a development set \(D_{\text{dev}}\) as long as \(D_{\text{dev}}\) is distinct from \(D_{\text{cal}}\).

**Shared calibration**

\(C_n^{\text{ind}}\) has the advantage of calibrating each layer independently. As \(l\) grows, however, \(\alpha_k\) will tend to 0 in order to retain validity (as specified by Theorem 6.4.4). As a result, \(C_n^{\text{ind}}\) will...
lose statistical efficiency. Following a similar approach to [36] and [71], we compute a new test statistic, $M_{\text{max}}$, as

$$M_{\text{max}}(x) = \max_{k \in [l-1]} \{ M_k(x) : F_k(x) \neq F(x) \}. \quad (6.9)$$

We discard ill-defined values when $M_{\text{max}}(x) = \max \emptyset$. $M_{\text{max}}(x)$ reflects the worst-case confidence across inconsistent layers for input $x$ (i.e., where $M_k(x)$ predicts a high consistency likelihood for layer $k$ when layer $k$ is, in fact, inconsistent). This worst-case statistic allows us to keep a constant significance level $\epsilon$, even as $l$ grows. Let $m^{(1:n,\infty)}$ denote the inflated empirical distribution,

$$\{ M_{\text{max}}(x_i) : x_i \in D_{\text{cal}}, \exists k \ F_k(x_i) \neq F(x_i) \} \cup \{ \infty \}.$$ 

We then define a single threshold shared across layers, $\tau_{\text{share}} = \text{Quantile}(1 - \epsilon, m^{(1:n,\infty)})$, and predict the inconsistent index set at $x \in X$ as

$$C_{\epsilon}^{\text{share}}(x) = \{ k : M_k(x) \leq \tau_{\text{share}} \} \quad (6.10)$$

**Theorem 6.4.6.** For any number of layers $l \in \mathbb{N}^+$, $C_{\epsilon}^{\text{share}}(X_{n+1})$ is a valid set that satisfies Eq. Equation 6.6

### 6.5 Experimental Setup

For our main results, we use an Albert-xlarge model [147] with 24 Transformer layers. Experiments using an Albert-base model and a RoBERTa-large model [167] are reported in Appendix C.2. As discussed in §6.3, our methods can be applied to any multilayered model such as BERT [56], GPT [29], ResNet [102], and others. See Appendix C.1 for implementation details.
Table 6.2: Task dataset and label space sizes. The rightmost column reports either test accuracy (classification) or Pearson-correlation (regression). *We downsample the 63K public development set to expedite validation.

| Dataset  | |Y|  | Train  | Dev. | Test  | F test perf. |
|----------|--------|---------|--------|------|-------|-------------|
| IMDB     | 2      | 20K     | 5K     | 25K  | 94.0  |
| VitaminC | 3      | 370K    | 10K*   | 55K  | 90.6  |
| AG News  | 4      | 115K    | 5K     | 7.6K | 94.4  |
| STS-B    | ∞      | 5.7K    | 1.5K   | 1.4K | 89.8  |

6.5.1 Tasks

We evaluate our methods on three classification tasks with varying label space size |Y| and difficulty: IMDB [172] sentiment analysis on movie reviews, VitaminC [242] fact verification with Wikipedia articles, and AG [90] [318] news topic classification. We also evaluate on the STS-B [37] semantic textual similarity regression task where Y ∈ [0, 5] ⊂ R. Dataset statistics, along with the test set performance of our original F model (Albert-xlarge), are contained in Table 6.2.

6.5.2 Baselines

In addition to our main methods discussed in §6.4.2, we compare to several non-CP baselines. Note that the following methods are not guaranteed to give well-calibrated performance (as our CP ones are).

**Static.** We use the same number of layers for all inputs. We choose the exit layer as the first one that obtains the desired consistency on average on D_cal.

**Softmax threshold.** Following [248], we exit on the first layer where p_k^{max} ≥ 1 − ε, where p_k^{max} denotes the maximum softmax response of our early classifier. Softmax values are calibrated using temperature scaling [92] on another held-out (labeled) data split, D_scale.

**Meta threshold.** Even if perfectly calibrated, p_k^{max} from softmax thresholding is not measuring consistency likelihood P(G(X) = F(X) | X = x), but rather P(G(X) = Y | X = x). This is equivalent if F is an oracle, but breaks down when F is not. We also
experiment with thresholding the confidence value of our meta classifier (§6.3.2) in a similar way (i.e., exiting when it exceeds $1 - \epsilon$).

6.5.3 Evaluation

For each task, we use a proper training, validation, and test set. We use the training set to learn $\mathcal{F}$ and $\mathcal{G}$. We perform model selection on the validation set, and report final numbers on the test set. For all methods, we report the marginalized results over 25 random trials, where in each trial we partition the data into 80% $\mathcal{D}_{\text{cal}}(x_1:n)$ and 20% $\mathcal{D}_{\text{test}}(x_{n+1})$. In order to compare different methods across all tolerance levels, we plot each metric as a function of $\epsilon$. Shaded regions show the 16-84th percentiles across trials. We report the following metrics:

**Consistency.** We measure the percent of inputs for which the prediction of the CAT model $\mathcal{G}$ is the same as the full Transformer on our test prediction, i.e., $\mathcal{G}(X_{n+1}) = \mathcal{F}(X_{n+1})$. For regression tasks, we count a prediction as consistent if it is within a small margin $\tau$ from the reference (we use $\tau = 0.5$). As discussed in §6.1, if $\mathcal{G}$ is $\epsilon$-consistent, we can also derive an average performance lower bound: it will be at least $(1 - \epsilon) \times \mathcal{F}$’s average performance.\footnote{In practice, the performance is likely to be higher than this lower bound, since inconsistencies with $\mathcal{F}$ could lead to a correct prediction when $\mathcal{F}$ would have otherwise been wrong.}

**Layers ($\downarrow$).** We report the computational cost of the model as the average number of Transformer layers used. Our goal is to improve the efficiency (i.e., use fewer layers) while preserving $\epsilon$-consistency. We choose this metric over absolute run-time to allow for implementation-invariant comparisons, but we provide a reference analysis next, to permit easy approximate conversions.

6.5.4 Absolute Runtime Analysis

The exact run-time of $\mathcal{G}$ depends on the efficiency of the hardware, software, and implementation used. Ideally, the early and meta classifiers can run in parallel with the following Transformer layer (layer $k + 1$). As long as they are faster to compute concurrently than
Figure 6-3: Classification results (dev). While both our CP-based methods give valid consistencies (above diagonal), *shared calibration* generally results in earlier exits. This advantage is especially pronounced at smaller tolerance levels (right-hand side), where it significantly outperforms other approaches. Our meta-learned confidence measure $\mathcal{M}_k$ improves over using the softmax response as a drop-in replacement, especially for tasks with larger $|\mathcal{Y}|$. Note that we care more about the right-hand side behavior, (i.e., larger $1 - \epsilon$), as it corresponds to higher consistency.

a single layer, this will avoid incurring any additional time cost. An alternative naive synchronous implementation could lead to inefficiencies when using a small tolerance $\epsilon$.

We provide a reference timing for the IMDB task implemented with the Transformers [305] library, PyTorch 1.8.1 [200], and an A100-PCIE-40GB Nvidia GPU with CUDA 11.2. A full forward path of an Albert-xlarge takes 22.32ms per input, 0.85ms $\times$ 24 for the transformer layers and 1.95ms for the embedding layer and top classifier. Our early classifier takes 0.20ms and the meta classifier takes 0.11ms. Therefore, with a naive implementation, a CAT model $\mathcal{G}$ with an average exit layer less than 17.6 with the meta classifier, or 19.5 without, will realize an overall reduction in wall-clock time relative to the full $\mathcal{F}$.
Table 6.3: Classification results (test) for specific tolerance levels. We report the accuracy lower bound guaranteed by our CP methods in parentheses. Shared/ Meta is reliably the most efficient method (and is $\epsilon$-consistent). Greyed rows reflect approaches without guarantees; our CAT approaches with guarantees are presented below them.

### Experimental Results

We present our main results. We experiment with both our meta classifier $M_k$ confidence score (Meta, §6.3.2), and, for classification tasks, the early classifier’s softmax response, $p_k^\text{max}$ (SM), as a drop-in replacement for $M_k$ (at no additional computational cost). Appendix C.2 reports results with other drop-in $M_k$ replacements, in addition to results using our naive development set calibration approach (§6.3.3). Appendix C.3 provides qualitative examples.

#### Classification Results

Figure 6-3 summarizes the average consistency and number of layers used by $G$ as a function of $\epsilon$, while Table 6.3 presents results for specific $\epsilon$ on task test sets. Independent calibration proves to be quite conservative due to the loss of statistical power from the loose union bound of the Bonferroni correction for large $l$ (here $l = 24$). At some levels of $\epsilon$, non-CP baselines perform competitively, however, they lack formal guarantees. Overall, for the most critical tolerance levels (small $\epsilon$, right-hand side of the plots), our shared method leads...
to significant efficiency gains while still maintaining the desired level of consistency (above the diagonal).

The effectiveness of our meta predictor, $M_k$, is most pronounced for tasks with $|\mathcal{Y}| > 2$, where the drop-in softmax score (SM) becomes less indicative of consistency. Both SM and Meta are relatively well-calibrated for IMDB and VitaminC, which makes the threshold-based exit rule a competitive baseline. Still, our Shared/ Meta method provides both reliable and significant gains.

The computational advantage of our CAT model is dependent on the average difficulty of the task and the implementation. As Table 6.3 shows, allowing up to an $\epsilon$ of 10% inconsistency, for two of the tasks we cut down the average Transformer layer to only 9 out of 24 using our Shared/ Meta model. Following the analysis in §6.5.4, this leads to an approximate speedup of $1.8 \times$ with a synchronous implementation and of $2.7 \times$ with a concurrent one, compared to running the full model. Moreover, Figure 6-4 illustrates the user’s control over available computational resources via modulating $\epsilon$. Decreasing $\epsilon$ increases the confidence level required before committing to the early classifier’s prediction (thereby increasing the average number of required layers), and vice-versa.

### 6.6.2 Regression Results

Table 6.4 and Figure 6-5 present results for our regression task, where we see similar trends. Here, an attractive advantage of our meta confidence predictor is its generalizability to
Table 6.4: Test results for the STS-B regression task.

<table>
<thead>
<tr>
<th>Method</th>
<th>Consist.</th>
<th>Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - (\epsilon) = 0.95:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>100.00</td>
<td>24.00</td>
</tr>
<tr>
<td>Thres./ Meta</td>
<td>99.87</td>
<td>19.19</td>
</tr>
<tr>
<td>Indep./ Meta</td>
<td>99.29</td>
<td>23.60</td>
</tr>
<tr>
<td>Shared/ Meta</td>
<td>96.42</td>
<td><strong>17.64</strong></td>
</tr>
<tr>
<td>1 - (\epsilon) = 0.90:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>92.51</td>
<td>20.00</td>
</tr>
<tr>
<td>Thres./ Meta</td>
<td>99.19</td>
<td>18.53</td>
</tr>
<tr>
<td>Indep./ Meta</td>
<td>97.77</td>
<td>20.26</td>
</tr>
<tr>
<td>Shared/ Meta</td>
<td>92.65</td>
<td><strong>17.29</strong></td>
</tr>
</tbody>
</table>

multiple task output types. Notice that the event space of \(1\{G(X) = F(X)\} = \{0, 1\}\) always, regardless of the original \(\mathcal{Y}^8\). This allows it to be easily adapted to tasks beyond classification, such as regression, where traditional softmax-based confidence measures (as used in, e.g., [248]) are absent.

\(^8\)As long as equality is suitably defined, e.g., for STS-B we define consistent outputs as being within \(\tau = 0.5\) away.
6.7 Conclusion

The ability to make predictions quickly without excessively degrading performance is critical to production-level machine learning systems. In fact, being capable of quantifying the uncertainty in a prediction and deciding when additional computation is needed (or not) is a key challenge for any intelligent system (e.g., see the System 1 vs. System 2 dichotomy explored in [126]).

In this work, we addressed the challenge of deciding when to sufficiently trust an early prediction of Transformer-based multilayer models by learning from their past predictions. Our Confident Adaptive Transformers (CATs) framework leverages meta predictors to accurately assess whether or not the prediction of a simple, early classifier trained on an intermediate Transformer representation is likely to already be consistent with that of the full model \(\mathcal{F}(X)\) (i.e., after all \(l\) layers of \(\mathcal{F}\) are computed). Importantly, we develop a new conformal prediction approach for calibrating the confidence of the meta classifier that is (1) simple to implement, (2) fast to compute alongside the Transformer, (3) requires only unlabeled data, and (4) provides statistically efficient marginal guarantees on the event that the prediction of the faster, amortized CAT model is consistent with that of the full \(\mathcal{F}\). Our results on multiple tasks demonstrate the generality our approach, and its effectiveness in consistently improving computational efficiency—all while maintaining a reliable margin of error.
Chapter 7

Conclusions

In this thesis, we have presented several methods for advancing automatic fact verification models for preventing misinformation online. In the first part, we focused on improving the robustness and sensitivity of models, as well as their interpretability. In the second part, we focused on improving the efficiency of the models to allow them to operate over comprehensive free text information resources.

In order to improve the robustness of the claim-evidence classifier, we present a contrastive evidence training paradigm that requires models to be sensitive to small changes in the evidence and avoid any statistical cues in the claim. We have presented two main approaches for achieving contrastive evidence. First, we developed a rationale-based denoising pipeline where we designed a model to extract the most important words in the evidence. We train a generator to recreate the original supporting evidence, and then apply it to modify refuting evidence for claims to supporting one, obtaining the desired contrastive pairs. Secondly, we presented a data curation methodology that naturally achieves the contrastive design by leveraging revisions to information resources. We collected Wikipedia revisions and created a large-scale dataset with claims that are paired with at least two contexts that reflect different conclusions about the claim’s veracity. We demonstrated the effectiveness of this paradigm by showing a significant increase in evidence sensitivity and robustness against adversarial examples for multiple sentence-pair classifiers. Furthermore, we showed how our techniques can also help in extracting word-level rationales, providing valuable fine explanations for the prediction of the classifier.
To reliably improve the efficiency of retrieval models and classifiers, we presented extensions to the conformal prediction framework. Building on conformal calibration, our methods provide marginal coverage guarantees. For evidence retrieval, we provide a user-specified arbitrary significance level for the existence of a sufficient evidence sentence for evaluating the claim in the retrieved set. As naive calibration is computationally expensive and could result in large prediction sets, we showed that our method provides substantial efficiency improvements, allowing smaller prediction sets with less computational effort. For sentence-pair classification, we presented a method for dynamically adapting the computational effort by introducing intermediate confidence measurements as the input representation propagates through a multi-layered classifier.

7.1 Limitations

We note that there are some limitations to our work which we hope to address in future work. Some concrete directions are detailed below. In addition, other than some human-evaluation experiments, we did not test how human readers and professional fact-checkers could interact with automatic fact verification systems. We expect that a complete system that meets all the specifications listed in Chapter 1 would be beneficial in reducing the number of times readers believe and share false statements. However, it remains to explore the right strategy for integrating such automatic systems in the online environment. We hypothesize that the best usage of fact verification systems is to identify disagreements between sources and to provide the user with recommendations backed by interpretable explanations and evidence. Since automatic systems will always have some degree of error, we believe that the end-user should make the final verdict. The goal of the models should be to assist the user towards making the right decisions.

Furthermore, we note that the conformal prediction framework that we relied on in Chapters 5 and 6 has some limitations. First, the method requires a held-out set of exchangeable examples for calibration. In Chapter 6, we only use unlabeled examples which make it less costly. Future work might explore reducing the need for large calibration sets. One direction is to utilize exchangeable auxiliary tasks if available [72]. Also, conformal prediction
provides marginal guarantees. In practice that could mean that the model underperforms on one group of examples while doing well on another. Specifying different groups of examples and controlling for conditional coverage, for example with Mondrian conformal prediction, could alleviate this issue. Other solutions with more local guarantees could also be useful.

### 7.2 Future Work

Following the work presented in this thesis, we list future related directions:

- **Combining multiple evidence pieces.** Most of the classifiers in this work rely on a single evidence sentence. We first focus on enforcing classifiers to attend to any context beyond the claim itself since, as we showed, this is not easily granted. Other models usually provide a trivial extension to the single evidence setting, either by concatenating all evidence sentences as a pre-process or by adding learned aggregation modules on top of the single evidence classifiers [84, 161, 168, 191]. These methods, however, mostly ignore the origin of each evidence [111], including when and by whom it was given. As we explored in Chapter 4 and is also discussed in [268], some evidence pieces can contradict each other. Utilizing richer features in our models can help reach a better verdict in such scenarios.

- **Information resources beyond English Wikipedia.** In this work, we focused on Wikipedia as our main evidence resource. Indeed, Wikipedia proved itself as a comprehensive and reliable resource that users tend to follow [19, 143]. For many languages, however, Wikipedia articles are more limited. The recent advancements in machine translation models can help in filling this gap but do not provide a complete solution. In some cases, articles in different languages disagree about certain facts. This could be a result of an outdated article in one language, or due to more fundamental disagreements between groups. In both cases, the previous point of combining multiple views is again relevant. A more complete solution, however, will also address at least the former case by automatically identifying disagreements between related articles and assisting in correcting them. Our methods for factual revision flagging and automatic
sentence modification can potentially help in this direction, especially if extended to multilingual applications [244]. Extending models to support other information resources, such as news outlets [259] could also be beneficial.

- **Identifying check-worthy claims.** Finally, in this work, we focus on fact-checking individual claims in the form of short textual assertions. This form of claim is most commonly seen in social media posts. We would like to extend our fact verification tools to many other applications. For example, many articles include multiple implicit claims. Extracting these claims and deciding which of them is check-worthy could be extremely valuable. The same is true for public speeches, either operating over the transcript or including voice features. Also, in some cases, a broader context is required to understand the claim, such as an image, video, or background information. In all these cases, being able to distill concrete claims will provide an important complementary extension to fact verification models.
Appendix A

Contrastive Evidence Pairs

A.1 VITAMINC: Complementary details

We provide additional details about the VITAMINC dataset.

A.1.1 Claim Statistics

**Topic Distribution.** Figure A.1.1 shows the distribution of claims in the VITAMINC dataset by the topic of the Wikipedia article they are based on. The information was collected from DBpedia, retrieving the parent class of the pages. Labels for about 25% of the articles were missing, and left blank in the diagram.

The “synthetic” part of VITAMINC, which is based on the claims of the FEVER dataset, contains many claims about specific human entities. About 15% of the claims in VITAMINC real are about COVID-19.

<table>
<thead>
<tr>
<th>Claim category</th>
<th>real</th>
<th>synthetic</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantitative</td>
<td>48%</td>
<td>9%</td>
<td>The COVID-19 pathogen may last less than 10 days on some surfaces.</td>
</tr>
<tr>
<td>Calendrical</td>
<td>9%</td>
<td>15%</td>
<td>Italy surpassed the 10,000 coronavirus-related deaths on a Saturday.</td>
</tr>
<tr>
<td>Entity</td>
<td>23%</td>
<td>58%</td>
<td>Mary of Teck was queen-consort.</td>
</tr>
<tr>
<td>Event</td>
<td>14%</td>
<td>14%</td>
<td>In the last EFL Cup, Manchester defeated Chelsea.</td>
</tr>
<tr>
<td>Other</td>
<td>6%</td>
<td>4%</td>
<td>Most genes need further research to better understand the function of their RNA products.</td>
</tr>
</tbody>
</table>

Table A.1.1: Estimated distribution of claims in the VITAMINC, based on manual annotations of 100 randomly sampled claims from the development split of the real and synthetic subsets. An example claim from each category is provided for reference.

Figure A.1.1: Distribution of claims in the VITAMINC dataset by the topic of the originated Wikipedia article.

**Category Distribution.** We sample 100 examples from the “real” and “synthetic” subsets of VITAMINC and manually categorize their claims. The results are presented in Table A.1.1. Due to the creation methodology of VITAMINC real, its claims mostly describe frequently updating facts, or facts that tend to be corrected. We find about half of these claims to describe changes in numerical values (e.g., number of COVID-19 cases, earnings or ratings of movies, number of awards etc.). In contrast, VITAMINC synthetic mostly covers general facts about specific entities, (e.g., place of birth, date of birth, occupation, etc.). This is a result of the synthetic claims being based on the FEVER dataset, where annotators were asked to come up with claims on popular Wikipedia pages. Combined, the VITAMINC dataset holds a diverse set of claims about various topics.

**A.1.2 Inter-annotator Agreement**

We ask three additional annotators to independently annotate a random set of two thousand claim-evidence pairs, evenly distributed between the development and test splits of the real and synthetic sets. The Fleiss $\kappa$ score [73] between the four annotations is 0.7065, which means substantial agreement. Similar agreement scores of 0.6841 and 0.7 were reported for fact verification [272] and NLI datasets [25], respectively.
Figure A.1.2: Probability density function of claim-evidence overlap for different labels in the dataset. The overlap is computed as the ratio of mutual bigrams in the two sentences.

A.1.3 Claim-only Classification

Annotation artifacts are common in crowd-sourced sentence-pair inference datasets such as fact verification and NLI. Models can leverage these idiosyncrasies to achieve unexpectedly high performance when given only one sentence of the pair. For example, [246] showed that a claim-only classifier can obtain 61.7% accuracy. The VITAMINC dataset avoids this bias by pairing each claim with two contrastive contexts.

All claims in the VITAMINC-synthetic are paired with one refuting and one supporting evidence, making it impossible for a claim-only to perform better than random. Each claim in the VITAMINC-real is paired with one refuting or neutral evidence, in addition to a supporting one. To evaluate whether models can utilize lexical cues in claims, we train a claim-only classifier on VITAMINC-real and find it to achieve 50% accuracy—the same as always predicting SUP.

A.1.4 Claim-evidence Word Overlap

Naturally, when pairing claims to evidence sentences, the overlapping words will be higher on average for claims with their supporting evidence. In VITAMINC dataset, we want to minimize this bias in order to create challenging examples that require sentence-pair inference and cannot be solved by simple word matching techniques. Therefore, we asked annotators, when possible, to avoid copying exact phrases from the evidence to the claim (see §4.3.2).

Figure A.1.2 shows the probability density function of bigram overlaps between the
claim and evidence for each relation. Similar to FEVER, the overlap ratio of supporting pairs in the VITAMINC dataset is only slightly higher than the one of refuting pairs. Also, the overlap ratio of the NEI pairs of the VITAMINC real dataset is on average higher than FEVER.

### A.2 Experimental Setting

We implement all our models with the HuggingFace Transformers library [305]. When comparing across training datasets of different sizes, we train the model for the same amount of update steps, upsampling the smaller datasets. We pick the checkpoint with the highest accuracy on the development set of the training task and report performance on the test set. More details are available at [https://github.com/TalSchuster/VitaminC](https://github.com/TalSchuster/VitaminC).

### A.3 GPT-3 Evaluation

The GPT-3 model has recently demonstrated impressive results in zero-shot and few-shot generation and classification tasks [29]. This 175B parameters language model was trained on billions of words from online sources, including the English Wikipedia. As result, it can be applied on many tasks without any further fine-tuning—instead, one need only provide a task-specific prefix (i.e., “prompt”) with a few examples that direct the language model towards the desired output format. For example, GPT-3 achieves better than random results on ANLI with only a single example in the prompt, and over 40% accuracy with 50 examples [29].

We used OpenAI’s beta API to query GPT-3. Due to our limited quota, we could not perform extensive experiments. Instead, we performed a qualitative evaluation using several examples from VITAMINC test set for the claim extraction (factually consistent generation) and the fact verification tasks. Therefore, these results should be viewed as exploratory only.

**GPT-3 for Claim Extraction.** We examine a two-shot setting for the claim extraction task. The model is asked to convert a revision into a short claim that expresses the fact that
is true after the edit. To guide the model for this task, we provide a prompt with two random examples from the VITAMINC training set (see Figure A.3.1). One of the main concerns regarding large language models is the limited control it allows for ensuring that the facts in the generated output align with the source [245]. The generation tasks of VITAMINC provide a useful test-bed for evaluating the factual consistency with the input. Importantly, our VITAMINC-trained fact verification classifiers ($f_{\text{verdict}}$) allow strong automatic evaluation for the factual agreement of the generation with the source.

We use GPT-3 to extract claims for four revisions with a sampling temperature value ($T$) set to either 0 or 0.7. The zero value is recommended for maximizing the factual consistency as the model follows its most certain predictions. Using low temperature, however, can result in less fluent generations [109]. Therefore, high values of $T$ are also commonly used.

The results are reported in Tables 4.7 and A.5.3. With only two guiding examples, GPT-3 is able to follow the desired format and create a short claim. Yet, some of its generations follow $s_{t-1}$ instead of $s_t$ or add new, unsupported facts. $f_{\text{verdict}}$ provides an indication for the factual correctness of the output. For example, it correctly classifies the output of the $T = 0.7$ setting for the top example in Table 4.7 as “Not Enough Information” since GPT-3 reported about 20 deaths even though the input doesn’t mention death numbers at all.

We expect GPT-3 to improve with longer prompts or fine-tuning and leave this to future research due to our limited quota.

**GPT-3 for Fact Verification.** We also experiment with using GPT-3 few-shot classification capabilities for the fact verification task. We follow the ANLI few-shot format of [29] and compose prompts with 6 examples (2 from each class) with random examples from VITAMINC training set. We use only numerical examples to evaluate numerical claims (Figure A.3.3), and mixed examples for other claims (Figure A.3.2). We set $T = 0$ as recommended for classification.

Table A.3.1 summarizes the results. Even with only six examples, GPT-3 seems to perform significantly better than random. Yet, its verdict is wrong in several cases that can be easily classified by humans. For example, we find it to refrain from predicting a True/False verdict even when the evidence is clear. We observe this both for a date-based
Life Is Peachy: Life Is Peachy is the first studio album by the American nu metal band Korn, released on October 15, 1996 through both Immortal Records and Epic Records.

Claim: Life Is Peachy is Korn’s second studio album.

2020 coronavirus pandemic in Kerala: As of 14 March 2020, there are 22 confirmed cases of the virus and more than 4000 people are under surveillance in Kerala.

Claim: As of 14 March, there have been more than 20 confirmed COVID-19 cases in Kerala.

Figure A.3.1: The prompt used for GPT-3 few-shot claim extraction.

To experiment with the sensitivity of the model to the provided context, we manually modified some of the examples to provide even stronger evidence. For example, while GPT-3’s prediction for line 5.2 is acceptable as actually, Turner Broadcasting System merged with WarnerMedia in 1996, changing the evidence to another disconnected entity (The Walt Disney Company) did not change the prediction (line 5.3) as expected. Even when explicitly stating that there is no other owner GPT-3 didn’t modify its verdict (line 5.4). Similarly, when evaluating the claim about the population of Beaverton being less than 90K, GPT-3 ignores the supporting evidence and outputs a false verdict (lines 1.4-1.5). Changing the claim to state “approximately 86K” instead of “less than 90,000” modified the prediction to “Neither” (line 1.6). Only repeating the exact same number as the evidence led to a true verdict (line 1.7).

A.4 Complementary Experiments

We report fact verification results with a fine-tuned BERT-base model in Table A.4.1. We find ALBERT-base to outperform BERT-base on most of the evaluated datasets. ALBERT-xlarge performed better than the two base models in all datasets except for Triggers. The Triggers dataset is very small (186 examples) and contains some unnaturally looking claims, which could explain the high variance across models.
Manchester is a major city and metropolitan borough in Greater Manchester, England, with a population of 545,500 as of 2017 (5th most populous English district).

Question: Manchester had a population of more than 540,000 in 2017 and was the 5th most populous English district. True, False, or Neither? True

As of March 2018, the apps have achieved more than 8 billion downloads.

Question: Talking Tom and Friends apps have less than 8 billion downloads. True, False, or Neither? False

He won the Premier League in 2018.

John Stones won both the Premier League and EFL Cup in 2018. True, False, or Neither? Neither

Neck Deep are a emo band.

Question: Neck deep is an emo band. True, False, or Neither? True

Critics generally gave The Final Frontier mixed to poor reviews.

Question: The film Star Trek V: The Final Frontier got negative reviews only. True, False, or Neither? False

The series was favorably compared to the HBO series The Jinx and the podcast Serial.

Question: The follow-up of the series Making a Murderer, was released in 2018. True, False, or Neither? Neither

Figure A.3.2: The prompt used for GPT-3 few-shot fact verification predictions on non-numerical claims (examples 2 and 4 in Table A.3.1). We follow the few-shot setting of [29] for ANLI.

### A.5 Example Outputs

We provide examples of predicted word-level rationales in Table A.5.1 and of outputs for the two generation tasks in Tables A.5.2 and A.5.3.
Manchester is a major city and metropolitan borough in Greater Manchester, England, with a population of 545,500 as of 2017 (5th most populous English district).
Question: Manchester had a population of more than 540,000 in 2017 and was the 5th most populous English district. True, False, or Neither? True

As of March 2018, the apps have achieved more than 8 billion downloads.
Question: Talking Tom and Friends apps have less than 8 billion downloads. True, False, or Neither? False

As of January 2015, JFC had a total of more than 3,000 stores worldwide, with system-wide retail sales totaling 82.1 billion pesos for the fiscal year 2011.
Question: Jollibee had a total of more than 20,000 stores worldwide after January 2016. True, False, or Neither? Neither

As of March 2018, the apps have achieved more than 8 billion downloads.
Question: Talking Tom and Friends apps have over 8 billion downloads. True, False, or Neither? True

Bet365 has more than 35 million customers globally.
Question: Bet365 has less than 30 million customers worldwide. True, False, or Neither? False

The series was favorably compared to the HBO series The Jinx and the podcast Serial.
Question: The follow-up of the series Making a Murderer, was released in 2018. True, False, or Neither? Neither

<Examined evidence>
Question: <Examined claim>. True, False, or Neither? <prediction>

Figure A.3.3: The prompt used for GPT-3 few-shot fact verification predictions on numerical claims (examples 1 and 3 in Table A.3.1)
<table>
<thead>
<tr>
<th>#</th>
<th>Claim</th>
<th>Evidence</th>
<th>GPT-3</th>
<th>Ours</th>
<th>Gold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td><strong>Less than 90,000</strong> people live in Beaverton, Oregon</td>
<td>its population is estimated to be 91,757, almost 14% more than the 2000 census figure of 76,129</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>1.2</td>
<td><strong>More than 90K</strong> people live in Beaverton</td>
<td>its population is estimated to be 91,757, almost 14% more than the 2000 census figure of 76,129</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>1.3</td>
<td><strong>More than 90K</strong> people live in Beaverton</td>
<td>its population is estimated to be 86,205, almost 14% more than the 2000 census figure of 76,129</td>
<td>Neither</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>1.4</td>
<td><strong>Less than 90,000</strong> people live in Beaverton, Oregon</td>
<td>its population is estimated to be 86,205, almost 14% more than the 2000 census figure of 76,129</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>1.5</td>
<td><strong>Less than 90,000</strong> people live in Beaverton, Oregon</td>
<td>Beaverton’s population is estimated to be 86,205</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>1.6</td>
<td><strong>Approximately 86k</strong> people live in Beaverton, Oregon</td>
<td>Beaverton’s population is estimated to be 86,205</td>
<td>Neither</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>1.7</td>
<td><strong>Approximately 86,205</strong> people live in Beaverton, Oregon</td>
<td>Beaverton’s population is estimated to be 86,205</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>2.1</td>
<td>Diego Corrales’ father was Puerto Rican and his mother Dominican</td>
<td>Corrales was born to a African American father and a Mexican mother</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>2.2</td>
<td>Diego Corrales’ father was Puerto Rican and his mother Dominican</td>
<td>Corrales was born to a Puerto Rican father and a Dominican mother</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>3.1</td>
<td><strong>COVID-19 outbreak was identified before December</strong></td>
<td>The outbreak was first identified in Wuhan, Hubei, China in December 2019 and recognized as a pandemic</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>3.2</td>
<td><strong>COVID-19 outbreak was identified before December</strong></td>
<td>The outbreak was first identified in Wuhan, Hubei, China in 17 November 2019 and recognized as a pandemic</td>
<td>Neither</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

Table A.3.1: See Table A.3.2 for lines 4-5. GPT-3 fact verification predictions on examples from the VITAMINC test dataset (examples 1.5-1.7 and 5.3-5.4 were manually modified to examine the model’s behavior). We follow the few-shot setting of [29] for ANLI (see Figures A.3.2 and A.3.3). The bold spans are for presentation and are not part of the input. Our VITAMINC-trained ALBERT classifiers predicted correctly on all these examples (though they weren’t picked this way). The GPT-3 few-shot succeeds on some examples and even expresses sensitivity to evidence in lines 2.1-2.2. In several cases, however, GPT-3 abstains from a True/False verdict, even when provided with strong evidence (see “Neither” predictions). Line 1.4 shows an example where GPT-3’s verdict is opposite of the provided evidence. Only when rephrasing the claim to exactly overlap with the evidence, it predicts an agreement.
<table>
<thead>
<tr>
<th>#</th>
<th>Claim</th>
<th>Evidence</th>
<th>GPT-3</th>
<th>Ours</th>
<th>Gold</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>There have been more than 400 confirmed coronavirus cases in Germany</td>
<td>There have been 444 confirmed cases and 16 recoveries of coronavirus in Germany</td>
<td>Neither</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>4.2</td>
<td>There have been more than 400 confirmed coronavirus cases in Germany</td>
<td>There have been less than 349 confirmed cases and 16 recoveries of coronavirus in Germany</td>
<td>Neither</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

| 5.1 | Cartoon Network is owned by Turner Broadcasting System               | Cartoon Network is an American pay television channel owned by Turner Broadcasting System, a subsidiary of AT&T’s WarnerMedia | True    | True | True |
| 5.2 | Cartoon Network is owned by Turner Broadcasting System               | Cartoon Network is an American pay television channel owned by Warner Bros. Entertainment, a subsidiary of AT&T’s WarnerMedia | Neither | False | False |
| 5.3 | Cartoon Network is owned by Turner Broadcasting System               | Cartoon Network is an American pay television channel owned by The Walt Disney Company  | Neither | False | False |
| 5.4 | Cartoon Network is owned by Turner Broadcasting System               | The Walt Disney Company is the only owner of Cartoon Network             | Neither | False | False |

Table A.3.2: Lines 4-5 of Table A.3.1

<table>
<thead>
<tr>
<th>Model</th>
<th>Train dataset</th>
<th>VitC real</th>
<th>VitC syn</th>
<th>FEVER</th>
<th>MNLI</th>
<th>Adversarial</th>
<th>Symmetric</th>
<th>Triggers</th>
<th>ANLI</th>
<th>Contrast</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT-base</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEVER</td>
<td>60.55</td>
<td>71.35</td>
<td>87.16</td>
<td>61.90</td>
<td>52.09</td>
<td>73.60</td>
<td>69.89</td>
<td>34.53</td>
<td>54.05</td>
<td></td>
</tr>
<tr>
<td>MNLI</td>
<td>46.31</td>
<td>69.01</td>
<td>70.06</td>
<td>83.80</td>
<td>50.13</td>
<td>73.88</td>
<td>65.05</td>
<td>26.88</td>
<td>51.92</td>
<td></td>
</tr>
<tr>
<td>FEVER + MNLI</td>
<td>56.24</td>
<td>81.80</td>
<td>95.59</td>
<td>85.06</td>
<td>63.05</td>
<td>85.11</td>
<td>37.63</td>
<td>29.63</td>
<td>60.63</td>
<td></td>
</tr>
<tr>
<td>VitC</td>
<td>85.80</td>
<td>90.63</td>
<td>74.21</td>
<td>66.66</td>
<td>76.24</td>
<td>90.17</td>
<td>63.98</td>
<td>33.19</td>
<td>72.49</td>
<td></td>
</tr>
<tr>
<td>VitC + MNLI</td>
<td>84.47</td>
<td>91.00</td>
<td>74.88</td>
<td>83.70</td>
<td>63.05</td>
<td>84.55</td>
<td>66.13</td>
<td>31.00</td>
<td>84.88</td>
<td></td>
</tr>
<tr>
<td>VitC + FEVER</td>
<td>84.72</td>
<td>89.16</td>
<td>87.55</td>
<td>69.28</td>
<td>64.75</td>
<td>90.73</td>
<td>72.58</td>
<td>34.06</td>
<td>84.01</td>
<td></td>
</tr>
</tbody>
</table>

Table A.4.1: Fact verification Complementary results for Table 4.4 with a BERT-base model.
Claim: The YouTube channel Chuchu TV is placed 42nd and has more than 25 million subscribers.
Evidence: Chuchu TV is the 43rd most subscribed YouTube channel in the world, with over 20 million subscribers.

Claim: The Rasmus has sold less than 4.5 million albums worldwide.
Evidence: The Rasmus has sold 5 million albums worldwide, 310,000 copies in their native Finland alone.

Claim: Darren Randolph is Spanish.

Claim: Astravyets is near Vilnius.
Evidence: His father may have migrated to the US in the 1860s from Astravyets near Vilnius.

Claim: The pace controlling stamina meter is a new feature in the game series.
Evidence: New to the series is a three-tier stamina meter which controls the pace of a match.

Claim: The movie will be released on 25 November 2015.
Evidence: [...] are producing the film which Columbia Pictures will release on November 25, 2015.

Table A.5.1: Example masks produced by the word-level rationale model for identifying anchoring words in the evidence that are responsible for the classifier's verdict regarding the claim (shown with green background). Masking these words leads the classifier to predict NEI instead of what would have been SUP or REF.
<table>
<thead>
<tr>
<th>Claim</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stephen Bruner was born on October 19, 1984.</td>
<td>Stephen Bruner (born October 19, 1984), better known by his stage name Thundercat, is an American multi-genre bass player, producer and singer from Los Angeles, California.</td>
</tr>
<tr>
<td>The House that Jack Built has an approval rate of over 45% and over 13 critics.</td>
<td>On Rotten Tomatoes, the film has an approval rating of 47%, based on 14 critics, and an average rating of 6.3/10.</td>
</tr>
<tr>
<td>Cartoon Network is owned by Warner Bros. Entertainment.</td>
<td>Cartoon Network is an American pay television channel owned by Warner Bros. Entertainment, a subsidiary of AT &amp; T’s WarnerMedia.</td>
</tr>
<tr>
<td>The New York Times reviewed the film Mad Max: Fury Road.</td>
<td>The New York Times wrote, “Miller has reminded us that blockbusters have the potential to not only be art, but radically visionary – even the fourth in a series.</td>
</tr>
</tbody>
</table>

Table A.5.2: Example outputs of the BART-base used for generating factually consistent revisions given the old version \( s_{t-1} \) and the updated claim we wish to support. The “ground-truth” \( s_t \) is provided for reference.
<table>
<thead>
<tr>
<th>$(s_{t-1}, s_t)$</th>
<th>WWE 2K15: As of August 2015, WWE 2K15 has shipped over[^1 six -&gt; 7.5] million units.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BART (VitC)</td>
<td>WWE 2K15 sold more than 7 million units. $</td>
</tr>
<tr>
<td>GPT-3 $\mathcal{T}=0$</td>
<td>As of August 2015, WWE 2K15 has shipped over seven million units $</td>
</tr>
<tr>
<td>GPT-3 $\mathcal{T}=0.7$</td>
<td>As of August 2015, WWE 2K15 has shipped over seven million units $</td>
</tr>
<tr>
<td>Reference</td>
<td>By August 2015, WWE 2K15 shipped over 7.5 million units. $</td>
</tr>
<tr>
<td>$(s_{t-1}, s_t)$</td>
<td>Pat Jennings: He has played for League of Ireland clubs UCD and is now at [^1 Shamrock Rovers -&gt; Dublin].</td>
</tr>
<tr>
<td>BART (VitC)</td>
<td>Pat Jennings is currently playing for Dublin club UCD and is now at Shamrock Rovers $</td>
</tr>
<tr>
<td>GPT-3 $\mathcal{T}=0$</td>
<td>He has played for League of Ireland clubs UCD and is now at Shamrock Rovers $</td>
</tr>
<tr>
<td>GPT-3 $\mathcal{T}=0.7$</td>
<td>He played for Shamrock Rovers and is now at Dublin $</td>
</tr>
<tr>
<td>Reference</td>
<td>Pat Jennings currently plays for the Dublin club. $</td>
</tr>
</tbody>
</table>

Table A.5.3: Additional examples for Table 4.7. Example outputs for extracting claims that express the factual change in a Wikipedia revision. The BART-base model is trained on VITAMINC data and GPT-3 is applied in a 2-shot setting with a temperature of 0 or 0.7. The revision $(s_{t-1}, s_t)$ is given to the model as a single sentence visualization where the edits are between curly brackets, preceded by the article’s title. The human-written claim is provided for reference. The prediction of our ALBERT-xl VITAMINC-trained model $f_{\text{verdict}}(c, s_t)$ on the generated claim against $s_t$ is also reported in the rightmost column.
Appendix B

Cascaded Conformal Prediction

B.1 Implementation Details

Multiple labeled answers. When multiple answers are given (i.e., \( y_{n+1} \) is a set) we take \( A_g(x_{n+1}, y_{n+1}) \) as the union of all the admissible answers, along with any additional answers expanded by \( g \). For the standard conformal prediction baseline, we calibrate on one of the answers at a time, chosen uniformly at random. Note that this is important to preserve equivalent sample sizes across min-calibrated CP and standard CP\(^1\).

Open-Domain QA. We use the open-domain setting of the Natural Questions (NQ) dataset \([144]\). In this task, the goal is to find a short span from any article in Wikipedia that answers the given question. Questions in the NQ dataset were sourced from real Google search queries, and human annotators identified answer spans to the queries in Wikipedia articles (we use the short answer span setting, and only consider answerable, non-boolean questions).

We use the open-source, pre-trained DPR model for retrieval (i.e., the document retriever) and the BERT model for question answering (i.e., the document reader) provided by \([130]\). To summarize briefly, the DPR model is trained to maximize the dot-product similarity between the dense representations (obtained via BERT embeddings) of the question and the passage that contains the answer. For candidate passages, we use the Wikipedia-based corpus processed by \([130]\), where each article is split into disjoint passages of 100 words, resulting

\(^1\)Another possibility would be to calibrate on all the given answers, but we found this generally does worse.
Table B.1.1: Nonconformity measures used in our experiments.

<table>
<thead>
<tr>
<th>Task</th>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA</td>
<td>retriever</td>
<td>−1· similarity score of the question and paragraph by the retrieval model.</td>
</tr>
<tr>
<td>QA</td>
<td>passage</td>
<td>−1· logit of the passage selection score by the BERT reader.</td>
</tr>
<tr>
<td>QA</td>
<td>span start</td>
<td>−1· logit of the answer’s first token by the BERT reader.</td>
</tr>
<tr>
<td>QA</td>
<td>span end</td>
<td>−1· logit of the answer’s last token by the BERT reader.</td>
</tr>
<tr>
<td>QA</td>
<td>span sum</td>
<td>span start + span end.</td>
</tr>
<tr>
<td>IR</td>
<td>BM25</td>
<td>−1· BM25 similarity score between query and candidate.</td>
</tr>
<tr>
<td>IR</td>
<td>CLS logit</td>
<td>−1· logit of the claim-evidence pair by the ALBERT classifier.</td>
</tr>
<tr>
<td>DR</td>
<td>RF</td>
<td>−1· score of the candidate by the Random Forest model.</td>
</tr>
<tr>
<td>DR</td>
<td>MPNN</td>
<td>−1· score of the candidate by the directed Message Passing Neural Network.</td>
</tr>
</tbody>
</table>

in a total of 21,015,324 passages. The DPR model pre-computes dense representations for all passages and indexes them with FAISS [124] for efficient retrieval. At test time, relevant documents for a given dense question encoding are retrieved via fast similarity search. The reader model is a standard BERT model with an independent span prediction head. This model encodes the question and passage jointly, and therefore, the representations cannot be pre-computed. For each token, the model outputs independent scores for being the start or end of the answer span. We also follow [130] by using the output of the “[CLS]” token to get a passage selection score from the reader model (to augment the score of the retriever). For ease of experimentation, we only consider the top 5000 answer spans per question—taken as the top 100 passages (ranked by the retriever) and the top 50 spans per passage (ranked by the reader). In order to be able to evaluate all $\epsilon \in (0, 1)$ we discard questions whose answers do not fall within this selection. We retain 6750/8757 questions from the validation set and 2895/3610 from the test set.

We compose the cascade for this task using four metrics: the retriever score, followed by the reader’s passage selection score, the “span start” score, and the “span end” score. For the non-cascaded models, we take the sum of the span start and span end scores as a single metric.

**IR.** We use the FEVER dataset for evidence retrieval and fact verification [272]. We focus on the retrieval part of this task. Note that the retrieved evidence can then be used to verify the correctness of the claim automatically [191, 246], or manually by a user.
follow the dataset splits of the Eraser benchmark [58] that contain 97,957 claims for training, 6,122 claims for validation, and 6,111 claims for test. The evidence needs to be retrieved from a set of 40,133 unique sentences collected from 4,099 total Wikipedia articles.

We compose the cascade using two metrics: an efficient non-neural BM25 model, and a neural sentence-pair classifier. Our BM25 retriever uses the default configuration available in the Gensim library [222]. We perform simple preprocessing to the text, including removing punctuation and adding word stems. We also add the article title to each sentence. Our neural classifier (CLS) is built on top of ALBERT-Base and is trained with BCE on (claim, evidence) pairs. We collect 10 negative pairs for each positive one by randomly selecting other sentences from the same article as the correct evidence. For shorter articles, we extend the negative sampling to also include the top (spurious) candidates identified by the BM25 retriever.

**DR.** We construct a molecular property screening task using the ChEMBL dataset [see 177]. Given a specified constraint such as “is active for property A and property B but not property C”, we want to retrieve at least one molecule from a given set of candidates that satisfies this constraint. The motivation of this task is to simulate in-silico screening for drug discovery, where it is often the case where chemists will searching for a new molecule that satisfies several constraints (such as toxicity and efficacy limits), out of a pool of many possible molecular candidates.

We split the ChEMBL dataset into a 60-20-20 split of molecules, where 60% of molecules are separated into a train set, 20% into a validation set, and 20% into a test set. Next, we take all properties that have more than 1000 labeled molecules (of which at least 100 are positive and 100 are negative, to avoid highly imbalanced properties). Of these \( \sim 200 \) properties, we take all \( N \choose K \) combinations that have at least 100 molecules with all \( K \) properties labelled (ChEMBL has many missing values). We set \( K \) to 3. For each combination, we randomly sample an assignment for each property (i.e., \( \{\text{active, inactive}\}^K \)). We keep 5000 combinations for each of validation and test sets. The molecules for each of the combinations are only sourced from their respective splits (i.e., molecular candidates for constraints in the property combination validation split only come from the molecule validation split). Therefore, at inference time, given a combination we have never seen
before, on a molecules we have never seen before, we must try to retrieve at least one molecule that has the desired combination assignment.

Both our random forest (RF) and the directed Message Passing Neural Network (MPNN) were implemented using the chemprop repository \[313\]. The RF model is based on the Scikit library \[202\] and uses 2048-bit binary Morgan fingerprints \[226\] of radius 2 to independently predict all target properties (active or inactive) for a given molecule. The RF model is fast to run during inference, even on a single CPU. The MPNN model uses graph convolutions to learn a deep molecular representation, that is shared across property predictions. Each property value (active/inactive) is predicted using an independent classifier head. The final prediction is based on an ensemble of 5 models, trained with different random seeds. Given a combination assignment \( (Z_1 = z_1, \ldots, Z_k = z_k) \), for both the RF and MPNN models, we take the nonconformity score as the model’s negative log-likelihood, where the likelihood is computed independently, i.e. \( p_\theta(Z_1 = z_1, \ldots, Z_k = z_k) = \prod p_\theta(Z_i = z_i) \).

### B.2 Additional Experimental Results

We provide supplemental experimental results to those shown in \S 5.6. In B.2.1 we show an example of a closed-domain QA task where cascading multiple measures actually boosts the predictive efficiency to the extent that it outweighs the conservative MHT correction factor. In B.2.2 we compare our method to heuristic methods at fixed efficiency levels, as described in \S 5.7.

#### B.2.1 Complementary Conformal Cascades for Closed-domain QA

The primary motivation in this work for conformalized cascades is to improve computational efficiency by allowing cheaper models to pre-filter the candidate space prior to running more expensive and more powerful models. Though not guaranteed, in some cases it is also possible that combining different nonconformity scores together has an overall synergistic effect that outweighs the generally conservative effects of the MHT corrections. This is similar in theory to ensembles or mixtures-of-experts \[115\], and similar results have been reported for combined conformal prediction \[34, 162, 278\].
While the tasks in §5.7 focus on cascades that are designed primarily for efficiency, here we also explore cascades for the smaller-scale task of closed-domain questions answering on the SQUAD 2.0 dataset [220]. This task isolates the document reader aspect of the open-domain QA pipeline, in which a relevant passage is already given. We cascade two primary models: (1) a span extractor (EXT) that gives independent scores for the start and end positions of the answer span, and (2) a more expensive answer classifier (CLS) that considers the entire candidate span (i.e., models the start and end jointly). We briefly outline the implementation details before giving the results below.

The EXT model uses the ALBERT-Base QA model [147]. It is trained to maximize the likelihood of the answer span \([i, j]\), where \(p(\text{start} = i, \text{end} = j)\) is modeled as \(p(\text{start} = i)p(\text{end} = j)\). During inference, the model computes all \(O(n^2)\) start and end position scores, and predicts the pair with the highest sum. We use the start and end position scores as two separate nonconformity measures. The CLS model is also built on top of ALBERT-Base, and is similar to [150]. Instead of scoring start and end positions independently, we concatenate the corresponding hidden representations at tokens \(i\) (start) and \(j\) (end), and score them jointly using an MLP. We then train with binary cross-entropy (BCE) over correct and incorrect answers. We limit the number of negative samples (incorrect answers) to the top 64 incorrect predictions of the EXT model.

The authors keep the original test set hidden; for ease of CP-specific evaluation we re-split the development set randomly by article. This results in 5,302 questions in our validation set, and 6,571 questions in our test set. The average length of a paragraph in the evaluation set is 127 words. Together with the “no answer” option, a question for a paragraph of that length will have 8,129 candidate answer spans. For the purposes of experimentation, we filter out questions for which the EXT model does not rank the correct answer within the top 150 predictions (so we can compute the full \((0, 1)\) range of \(\epsilon\) tractably). This discards less than 0.5% of questions.

Our results across several values of epsilon are given in Table B.2.1. As in the tasks in §5.7, the min-calibrated CPs greatly improve over the baseline CP. In this case, however, the cascaded CP also consistently outperforms the CP with only a single measure.
B.2.2 Heuristic Methods

In addition to evaluating our improvements over regular conformal prediction, we compare our conformal method to other common heuristics for making set-valued predictions. Specifically, we consider baseline methods that given some scoring function \(\text{score}(x, y)\) and threshold \(\tau\) to define the output set of predictions \(\mathcal{B}\) at \(x \in \mathcal{X}\) as

\[
\mathcal{B}(x, \tau) := \{y \in \mathcal{Y} : \text{score}(x, y) \geq \tau\},
\]

where \(\tau\) is then tuned on the calibration set to find the largest threshold for the desired accuracy:

\[
\tau^*_\epsilon := \sup \left\{ \tau : \frac{1}{n} \sum_{i=1}^{n} 1\{y_i \in \mathcal{B}(x, \tau)\} \geq 1 - \epsilon \right\}.
\]

The prediction for the test point \(x_{n+1}\) is then \(\mathcal{B}(x_{n+1}, \tau^*_\epsilon)\). We consider two variants: (1) fixed top-\(k\), where \(\text{score} := -\text{rank}(\text{metric}(x_{n+1}, y))\) according to some metric, and (2) raw thresholding, where \(\text{score} := \text{metric}(x_{n+1}, y)\), i.e., some raw, unnormalized metric. Top-\(k\) is simple and relatively robust to the variance of the metric used, but as it doesn’t depend on \(x\), it also means that both easy examples and hard examples are treated the same (giving prediction sets that are too large in the former, and too small in the latter). Raw metric thresholding, on the other hand, gives dynamically-sized prediction sets, but is more sensitive to metric variance. We emphasize that these baselines do not provide theoretical coverage guarantees for potentially non-i.i.d. finite samples.

When ignoring smoothing factors and restricting ourselves to a single, non-cascaded metric, it is straightforward to see that choosing the nonconformity measure to be \(S := \)

<table>
<thead>
<tr>
<th></th>
<th>CP Succ.</th>
<th>Succ.</th>
<th>(\mathcal{C}_n) Succ.</th>
<th>Succ.</th>
<th>(\mathcal{C}_n^{\min}) Succ.</th>
<th>Amortized cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 - \epsilon)</td>
<td>0.99</td>
<td>1.00</td>
<td>17.69</td>
<td>0.99</td>
<td>6.68</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>0.95</td>
<td>0.98</td>
<td>5.14</td>
<td>0.95</td>
<td>3.14</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.95</td>
<td>3.09</td>
<td>0.90</td>
<td>1.98</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>0.86</td>
<td>1.66</td>
<td>0.80</td>
<td>1.31</td>
<td>0.38</td>
</tr>
</tbody>
</table>

Table B.2.1: CP results on the SQUAD 2.0 dataset.
\[ \text{rank}(\text{metric}(x, y)) \] or simply \( S := -\text{metric}(x, y) \) makes the set \( C_n(x_{n+1}) \) equivalent to \( B(x_{n+1}, \bar{\tau}_\epsilon) \), where

\[ \bar{\tau}_\epsilon := \text{Quantile} \left( 1 - \epsilon, \hat{F} \left( \{ S(x_1, y_1), \ldots, S(x_n, y_n) \} \cup \{ \infty \} \right) \right) . \] (B.3)

We write \( \hat{F}(\mathcal{V}) \) to denote the empirical distribution of set \( \mathcal{V} \). For large enough \( n \), \( \bar{\tau}_\epsilon \) becomes nearly identical to \( \tau_\epsilon^* \). Note, however, that this comparison only applies to the split (i.e., inductive) conformal prediction setting. Metrics computed without relying on a held-out calibration set (as full CP is able to do) must treat the \( n + 1 \) data points symmetrically, a key feature of CP. Our methods extend to multiple cascaded metrics, finite \( n \), and the full CP setting.

We compare predictive efficiency results when using top-\( k \) and threshold heuristics versus our CP methods in Figure B.2.1 and Table B.2.2. As expected, with large enough \( n \), the baseline CP and Threshold methods are nearly equivalent. In some cases, top-\( k \) outperforms the threshold- and CP-based methods that use raw scores (this is likely due to variance in the densities of high scoring candidates across examples). When applying optimizing for admissible answers, our \( \text{min} \)-calibrated CP improves over all three methods. Note that optimizing for admissible answers is also applicable for the heuristic methods, in which case the trends will be similar to that of the baseline CP.

### B.3 Multiple Hypothesis Testing

As we discuss in §5.5.1, naively combining multiple hypothesis tests will lead to an increased family-wise error rate (FWER). For a visual example, see the uncorrected cascaded CP (blue dashed lined) in Figure B.3.1. It demonstrates that combining nonconformal measures without using a MHT correction can result in accuracies smaller than \( 1 - \epsilon \) (i.e., invalid coverage).

Several methods exist for correcting for MHT by bounding the FWER (with different assumptions on the dependencies between the hypothesis tests). We experiment with the Bonferroni and Simes procedures. We test the corrections on each task’s validation set and find Simes to work well for QA and IR, but not to hold for DR (See Figure B.3.1).
Figure B.2.1: How does our conformal method compare to non-conformal baselines? We show predictive efficiency and accuracy across various target accuracies (values of $1 - \epsilon$). As expected, with a large enough calibration set, the performance of the threshold heuristic is similar to CP. The top-$k$ threshold is harder to tune, resulting in lower than the desired accuracy for some values of IR, and sometimes in unnecessary large prediction sets. Specific values of $\epsilon$ are compared in Table B.2.2.

Therefore, we use the Bonferroni correction for DR and Simes for the other two tasks. For completeness, we briefly describe the two methods and the assumptions they rely on below (extensive additional details can be found in the literature).

### B.3.1 Bonferroni Procedure

The Bonferroni correction is easy to apply and does not require any assumptions on the dependencies between the hypothesis tests used (e.g., independent, positively dependent, etc). However, it is generally very conservative. The Bonferroni correction scales $\epsilon$ by a factor of $1/m$, and uses the union bound on the p-values ($P_1, \ldots, P_m$) to get:

$$
\mathbb{P}(Y_{n+1} \not\in C_n^m(X_{n+1})) = \mathbb{P}\left( \bigcup_{i=1}^m P_i \leq \frac{\epsilon}{m} \right) \leq \sum_{i=1}^m \mathbb{P}(P_i \leq \frac{\epsilon}{m}) = \epsilon.
$$
We achieve the same result by scaling each p-value by $m$, and take the final combined p-value to be the minimum of all the scaled p-values. It is easy to show that this correction is monotonic.

**B.3.2 Simes Procedure**

The Simes procedure [258] allows a stricter bound on the FWER when the measures are multivariate totally positive [239], which usually holds in practice [234]. To apply the correction, we first sort the p-values in ascending order, and then perform an order-dependent correction where the correction factor decreases as the index of the p-value increases. Specifically, if $(P_1, \ldots, P_m)$ are the sorted p-values $(P_1, \ldots, P_m)$ in an $m$-level cascade, we modify the p-values to be $P_{Sim}^{(i)} = m \cdot \frac{P^{(i)}}{i}$. We take the final combined p-value to be the minimum of the corrected p-values. It is easy to show that this correction is monotonic.
Figure B.3.1: Success rate against tolerance thresholds with different methods for MHT correction (validation set). Not applying any correction leads to an invalid classifier (the success rate is below the diagonal). The Bonferroni method is conservative, leading to a valid classifier, but sometimes with a higher accuracy rate than necessary (a result of having excessively large $C_n$). The Simes correction works for the QA and IR tasks and provides a tighter bound for them. The Simes correction does not work for the DR task, likely due to a violation of its MTP2 assumption, but the Bonferroni method provides a relatively tight correction there—especially for small $\epsilon$. 

(a) QA  
(b) IR  
(c) DR
Table C.1: Results (dev) using the naive development set calibration method (see §6.3.3).
This method tunes the early exit thresholds to get efficient \( \epsilon \)-consistent predictions on a
development set, but does not guarantee that prediction will be \( \epsilon \)-consistent on new data.
“Consist.” measures the empirical consistency on a test set, from which we compute a
guaranteed lower bound (“Bound”) to 99% confidence. The bound is significantly lower
than our target \( 1 - \epsilon \), and the measured consistency in our experiments also falls slightly
below \( 1 - \epsilon \) in some cases.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( 1 - \epsilon = 0.95 ):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>95.16</td>
<td>93.74</td>
<td>10.39</td>
<td>94.84</td>
<td>94.04</td>
<td>16.60</td>
<td>95.02</td>
<td>93.75</td>
<td>11.63</td>
</tr>
<tr>
<td>Meta</td>
<td>94.96</td>
<td>93.72</td>
<td>9.13</td>
<td>94.93</td>
<td>94.12</td>
<td>15.60</td>
<td>94.86</td>
<td>93.58</td>
<td>9.37</td>
</tr>
<tr>
<td>( 1 - \epsilon = 0.9 ):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>90.22</td>
<td>88.30</td>
<td>7.35</td>
<td>89.85</td>
<td>88.59</td>
<td>14.93</td>
<td>89.72</td>
<td>88.01</td>
<td>8.98</td>
</tr>
<tr>
<td>Meta</td>
<td>90.19</td>
<td>88.36</td>
<td>7.13</td>
<td>90.00</td>
<td>88.70</td>
<td>13.67</td>
<td>90.14</td>
<td>88.48</td>
<td>6.85</td>
</tr>
</tbody>
</table>

Appendix C

Confident Adaptive Transformers

C.1 Implementation Details

We implement our early exit Transformers (§6.3) on top of the Transformers library [305].
We set \( d_e \) to 32 in our experiments. For each task we fix a pre-trained \( F \) and train the early
and meta classifiers. We reuse the same training data that was used for \( F \) and divide it to
70/10/20% portions for \( D_{tune} \), \( D_{scale} \) and \( D_{meta} \), respectively. For classification tasks, we add
the temperature scaling step [91] after the early training to improve the calibration of the
softmax. We run the scaling for 100 steps on \( D_{scale} \) using an Adam optimizer [137] with a
learning rate of $10^{-3}$. For the early and meta training we use the same optimizer as for $\mathcal{F}$.

We fix $\mathcal{F}$ rather than train it jointly with the new components of $\mathcal{G}$ to avoid any reduction in $\mathcal{F}$’s performance [309]. This also makes our method simple to train over any existing Transformer without having to retrain the whole model which could be very costly. Training all parameters of $\mathcal{G}$ jointly can lead to more efficient inference as the early representations will be better suited for classification [248, 81], but potentially with the cost of reducing the accuracy of $\mathcal{F}_l$. In the case of joint training, our CATs will provide consistency guarantees with respect to the jointly-trained $\mathcal{F}_l$.

We implement the conformal calibration process in Python and perform retrospective analysis with different random splits of $\mathcal{D}_{cal}$ and $\mathcal{D}_{test}$. For Theorem 6.4.4 we simply use the uniform Bonferroni correction, setting $w_k = \frac{1}{l-1}$ $\forall k$. For the naive development set calibration, we use a shared threshold across all layers in order to reduce the examined solution space in Equation 6.3.

C.2 Additional Results

In this section, we provide complementary results for the experiments in Chapter 6. All results, except for sections C.2.4 and C.2.5, are with an Albert-xlarge model as $\mathcal{F}$, similar to the Chapter 6. However, we note that the results in these tables are based on the development sets, while the tables in Chapter 6 report the test set results.

C.2.1 Naive Development Set Calibration

For completeness, we evaluate the simple, but naive, calibration method described in §6.3.3. Recall that in this approach we first tune $\tau$ on a development set, and then bound the resulting $\mathcal{G}$’s accuracy using another heldout calibration split. The bound we get is static; we are not able to guarantee that it will satisfy our performance constraint in Eq. Equation 6.1.

Table C.1 gives results for our models when using either the Meta or SM confidence measures (which we threshold with $\tau$). We use half of $\mathcal{D}_{cal}$ to find the minimal threshold that provides $\epsilon$-consistency. Then, we evaluate the threshold on the second half of $\mathcal{D}_{cal}$ to get the empirical error. We compute the test set bound on this error with a confidence of
The test statistic used for a conformal prediction is typically called a nonconformity measure (i.e., in our work this is $\mathcal{M}_k(x)$). We experiment with different nonconformity measures as drop-in replacements for $\mathcal{M}_k(x)$, and report the results in Table C.2. The conformal calibration guarantees validity with any measure, even a random one, as long as they retain exchangeability. Good measures are ones that are statistically efficient, and will minimize the number of layers required for prediction at the required confidence level. This is a result of smaller $C_n$ sets, that tightly cover the inconsistent layers (and hence are more judicious estimates) of the number of layers required for prediction at the required confidence level. This is a result of small $C_n$ sets, that tightly cover the inconsistent layers (and hence are more judicious)

<table>
<thead>
<tr>
<th>Nonconformity measure</th>
<th>IMDB</th>
<th>VitaminC</th>
<th>AG News</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 - \epsilon = 0.95$:</td>
<td>(88.50)</td>
<td>(85.17)</td>
<td>(89.02)</td>
</tr>
<tr>
<td>Random</td>
<td>97.23</td>
<td>91.56</td>
<td>21.57</td>
</tr>
<tr>
<td>$D_{K L}(p_{k-1}</td>
<td></td>
<td>p_k)$</td>
<td>97.36</td>
</tr>
<tr>
<td>$\mathcal{H}(p_k)$</td>
<td>97.28</td>
<td>92.84</td>
<td>12.49</td>
</tr>
<tr>
<td>$p_k^{\text{diff}}$</td>
<td>97.28</td>
<td>92.84</td>
<td>12.49</td>
</tr>
<tr>
<td>$p_k^{\text{max}}$ (SM)</td>
<td>97.28</td>
<td>92.84</td>
<td>12.49</td>
</tr>
<tr>
<td>Meta</td>
<td>96.99</td>
<td>92.24</td>
<td>10.75</td>
</tr>
</tbody>
</table>

| $1 - \epsilon = 0.90$: | (83.84) | (80.69) | (84.33) |
| Random               | 94.52 | 89.68 | 19.21 | 93.94 | 85.44 | 21.47 | 94.27 | 89.28 | 19.01 |
| $D_{K L}(p_{k-1}||p_k)$ | 94.48 | 91.36 | 12.13 | 93.76 | 86.81 | 20.49 | 93.88 | 89.98 | 14.59 |
| $\mathcal{H}(p_k)$   | 94.49 | 91.31 | 9.91 | 93.67 | 86.41 | 16.29 | 94.54 | 90.80 | 13.08 |
| $p_k^{\text{diff}}$  | 94.49 | 91.31 | 9.91 | 93.67 | 86.53 | 16.11 | 94.60 | 90.56 | 10.69 |
| $p_k^{\text{max}}$ (SM) | 94.49 | 91.31 | 9.91 | 93.68 | 86.44 | 16.13 | 94.05 | 90.76 | 11.01 |
| Meta                 | 94.40 | 90.45 | 8.80 | 93.74 | 86.17 | 15.09 | 94.08 | 89.72 | 8.88 |

Table C.2: Results (dev) of our Shared model on the classification tasks using different nonconformity measures. $p_k^{\text{diff}}$ and $p_k^{\text{max}}$ are defined in Table 6.1. $D_{K L}(p_{k-1}||p_k)$ is the Kullback-Leibler Divergence between the previous layer’s softmax outputs and the current layer, and $\mathcal{H}(p_k)$ is the entropy of the softmax outputs. Our CP-based Shared method provides the guaranteed consistency with any measure, even random. The benefit, however, of using a better measure is in confidently exiting earlier. Our Meta measure allows the use of least Transformer layers meeting the consistency requirement with enough confidence.

As expected, the lower bound we compute is often significantly below $1 - \epsilon$, as it reflects the uncertainty that our measured consistency is accurate. Often the measured empirical consistency is also slightly below $1 - \epsilon$. At a high level, the overall consistency vs. efficiency trade-off is otherwise broadly similar to the one obtained by the Shared CP calibration.

### C.2.2 Nonconformity measure comparison

The test statistic used for a conformal prediction is typically called a nonconformity measure (i.e., in our work this is $\mathcal{M}_k(x)$). We experiment with different nonconformity measures as drop-in replacements for $\mathcal{M}_k(x)$, and report the results in Table C.2. The conformal calibration guarantees validity with any measure, even a random one, as long as they retain exchangeability. Good measures are ones that are statistically efficient, and will minimize the number of layers required for prediction at the required confidence level. This is a result of smaller $C_n$ sets, that tightly cover the inconsistent layers (and hence are more judicious estimations) of the number of layers required for prediction at the required confidence level. This is a result of small $C_n$ sets, that tightly cover the inconsistent layers (and hence are more judicious estimates) of the number of layers required for prediction at the required confidence level.
Figure C.2.1: Distribution of exit layers per tolerance level $\epsilon$ (dev sets) with our Shared/Meta Albert-xlarge model.

See Figure 6-4 for IMDB.

with the complement, $C_n^c$). To be consistent with previous work where softmax metrics are used [such as 248], we use $p_k^{\text{max}}$ as our non-Meta baseline in Chapter 6. In some settings, however, $p_k^{\text{diff}}$ performs slightly better.
C.2.3 Exit Layer Statistics

Figure C.2.1 depicts the distribution of exit layers for the different tasks with three reference tolerance levels. Reducing $\epsilon$ requires greater confidence before exiting, resulting in later exits on average. We provide example inputs with their respective exit layer in Appendix C.3.

C.2.4 Albert-base Results

Figure C.2.2 reports the classification and regression results with an Albert-base 12-layers model. The trends are similar to the larger 24-layers version. Again, we see the efficacy of our Shared conformal calibration and the Meta nonconformity scores. For example, the AG News CAT Shared/ Meta model can preserve 95% consistency while using less than 5 Transformer layers on average.

C.2.5 RoBERTa-large Results

Figure C.2.3 shows the results of our methods on top of the RoBERTa-large 24-layers Transformer. One main difference between RoBERTa and Albert, is that Albert shares the same parameters across all layers, essentially applying the same function recursively, whereas RoBERTa learns different parameters per layer. Yet, our method is agnostic to such differences and, as observed in the plots, results in similar trends. The value of our Meta classifier compared to the softmax response is even greater with the RoBERTa model.
Figure C.2.2: Development set results with an Albert-base 12-layers model as $\mathcal{F}$. 
Figure C.2.3: Development set results with an RoBERTa-large 24-layers model as $\mathcal{F}$. 

(a) IMDB
(b) VitaminC
(c) AG News
(d) STS-B
C.2.6 Conditional Conformal Calibration

Up until this point, we have been concerned with maintaining a marginal guarantee on 
$$\mathbb{P}(G(X_{n+1}) = \mathcal{F}(X_{n+1}))$$, where the randomness is over calibration points $$X_{1:n}$$ and test point $$X_{n+1}$$. In reality, however, we typically only care about consistency when $$\mathcal{F}(x)$$ is correct, as making an inconsistent prediction $$G(x) \neq \mathcal{F}(x)$$ in this case will necessarily result in an error. If exchangeable labeled calibration data is available, $$(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n$$, then we can consider a more refined version of Eq. Equation 6.1:

$$\mathbb{P}(G(X_{n+1}) = \mathcal{F}(X_{n+1}) \mid \mathcal{F}(X_{n+1}) = Y_{n+1}) \geq 1 - \epsilon. \quad (C.1)$$

Note that this criterion still guarantees the same lower bound on $$G$$’s performance—i.e., $$(1 - \epsilon)$$-fraction of $$\mathcal{F}$$’s original accuracy—and only requires minimal changes to our non-conditional algorithms. Empirically, we find that optimizing for Eq. Equation C.1 instead of Eq. Equation 6.1 leads to more efficient $$G$$.

**Corollary C.2.0.1.** Assume that examples $$(X_i, Y_i), i = 1, \ldots, n + 1$$ are exchangeable. Let $$\mathcal{A}$$ be the filtered set of correctly modeled examples, i.e.,

$$\mathcal{A} := \{X_i : \mathcal{F}(X_i) = Y_i\}. \quad (C.2)$$

For any $$\epsilon \in (0, 1)$$ and conformal procedure $$\mathcal{P}$$ satisfying Eq. Equation 6.6 apply $$\mathcal{P}$$ using only $$X_j \in \mathcal{A}$$. Then all derived $$G$$ (as defined prior) satisfy Eq. Equation C.1.

We evaluate this version on our three classification tasks and report the results in Table C.2.2. We also report the conditioned consistency: $$\mathbb{E}[G(X_{n+1}) = \mathcal{F}(X_{n+1}) \mid \mathcal{F}(X_{n+1}) = Y_{n+1}]$$.

The conditional calibration allows an additional gain in efficiency while meeting the accuracy goal. For example, the number of layers used for the IMDB task with $$\epsilon = 0.05$$ decreased from 10.75 to 10.02. Yet, the difference in performance is not substantial, demonstrating the strength of our non-conditional calibration with unlabeled data. The
Table C.2.2: Results (dev) with the Shared/ Meta model comparing the non-conditional (unlabeled) with the conditional calibration. Given the availability of labeled training data, conditional calibration allows a refined guarantee of consistency only over the inputs that are predicted correctly by $F$. In return, this leads to improved efficiency as the meta classifier performs better on these inputs. We report both conditional and non-conditional consistency results here and observe the validity of both calibration methods. In chapter 6 we use non-conditional calibration for all classification tasks as we assume an unsupervised setting.

difference between the two versions might become more significant in cases where $F$ is less accurate, resulting in a greater discrepancy between the conditional and non-conditional distributions.

### C.3 Example Predictions

Tables C.3.1-C.3.2 report examples of inputs for different tasks and the number of layers that our Albert-xlarge CAT with $\epsilon = 0.1$ required. These examples suggest that “easier” inputs (e.g., containing cue phrases or having large overlaps in sentence-pair tasks) might require less layers. In contrast, more complicated inputs (e.g., using less common language or requiring numerical analysis) can lead to additional computational effort until the desired confidence is obtained.
<table>
<thead>
<tr>
<th>Exit layer</th>
<th>Gold label</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>IMDB</strong> [172]</td>
</tr>
<tr>
<td>1</td>
<td>Pos</td>
<td>Without question, film is a powerful medium, more so now than ever before, due to the accessibility of DVD/video, which gives the filmmaker the added assurance that his story or message is going to be seen by possibly millions of people. [...]</td>
</tr>
<tr>
<td>4</td>
<td>Neg</td>
<td>This movie was obscenely obvious and predictable. The scenes were poorly written and acted even worse.</td>
</tr>
<tr>
<td>10</td>
<td>Pos</td>
<td>I think Gerard’s comments on the doc hit the nail on the head. Interesting film, but very long. [...]</td>
</tr>
<tr>
<td>15</td>
<td>Pos</td>
<td>here in Germany it was only shown on TV one time. today, as everything becomes mainstream, it’s absolute impossible, to watch a film like this again on the screen. maybe it’s the same in USA [...]</td>
</tr>
<tr>
<td>20</td>
<td>Neg</td>
<td>I tried to be patient and open-minded but found myself in a coma-like state. I wish I would have brought my duck and goose feather pillow... [...]</td>
</tr>
<tr>
<td>24</td>
<td>Neg</td>
<td>Hypothetical situations abound, one-time director Harry Ralston gives us the ultimate post-apocolptic glimpse with the world dead, left in the streets, in the stores, and throughout the landscape, sans in the middle of a forgotten desert. [...]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VitaminC [242]</th>
</tr>
</thead>
</table>
| 3              | Sup         | **Claim:** Another movie titled The SpongeBob Movie: Sponge on the Run is scheduled for release in 2020.  
**Evidence:** A second film titled The SpongeBob Movie: Sponge Out of Water was released in 2015, and another titled The SpongeBob Movie: Sponge on the Run is scheduled for release in 2020. |
| 5              | Sup         | **Claim:** Julie Bishop offered a defence of her nation’s intelligence cooperation with America.  
**Evidence:** The Australian Foreign Minister Julie Bishop stated that the acts of Edward Snowden were treachery and offered a staunch defence of her nation’s intelligence co-operation with America. |
| 10             | NEI         | **Claim:** The character Leslie hurts her head on the window in the film 10 Cloverfield Lane.  
**Evidence:** Michelle realizes Howard was right and returns his keys. |
| 15             | Sup         | **Claim:** Halakha laws are independent of being physically present in the Land of Israel.  
**Evidence:** The codification efforts that culminated in the Shulchan Aruch divide the law into four sections, including only laws that do not depend on being physically present in the Land of Israel. |
| 20             | Sup         | **Claim:** Germany has recorded less than 74,510 cases of coronavirus, including under 830 deaths.  
**Evidence:** 74,508 cases have been reported with 821 deaths and approximately 16,100 recoveries. |
| 24             | NEI         | **Claim:** For the 2015-16 school year, the undergraduate fee at USF is under $43,000.  
**Evidence:** Undergraduate tuition at USF is $44,040 for the 2016-17 school year. |

Table C.3.1: Number of Transformer layers used for example inputs from the task’s test sets with our Shared/Meta CAT with a tolerance level of $\varepsilon = 0.1$
<table>
<thead>
<tr>
<th>Exit layer</th>
<th>Gold label</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AG News [90, 318]</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Business</td>
<td>Crude Oil Rises on Speculation Cold Weather May Increase Demand Crude oil futures are headed for their biggest weekly gain in 21 months […]</td>
</tr>
<tr>
<td>5</td>
<td>Sports</td>
<td>NHL Owner Is Criticized for Talking of Replacement Players The day before the regular season was supposed to open […]</td>
</tr>
<tr>
<td>10</td>
<td>World</td>
<td>North Korea Says the Tyrant is Bush, not Kim North Korea says it sees no reason to join a working-level meeting with the United States […]</td>
</tr>
<tr>
<td>15</td>
<td>World</td>
<td>Scotch Whisky eyes Asian and Eastern European markets (AFP) AFP - A favourite tipple among connoisseurs the world over, whisky is treated with almost religious reverence on the Hebridean […]</td>
</tr>
<tr>
<td>20</td>
<td>Business</td>
<td>Arthritis drug withdrawn after trial A prescription painkiller used by more than 250,000 Australians to treat arthritis has been withdrawn from sale after a clinical trial found it doubled the risk […]</td>
</tr>
<tr>
<td>24</td>
<td>Sci/Tech</td>
<td>Airbus drops out of Microsoft appeal Aircraft builder withdraws its request to intervene in Microsoft’s antitrust appeal; Boeing also forgoes intervention.</td>
</tr>
</tbody>
</table>

**STS-B [37]**

| 10 | 0.6 | Sent. 1: A child wearing blue and white shorts is jumping in the surf.  
Sent. 2: A girl wearing green twists something in her hands. |
| 15 | 2.8 | Sent. 1: Saudi Arabia gets a seat at the UN Security Council  
Sent. 2: Saudi Arabia rejects seat on UN Security Council |
| 20 | 4.2 | Sent. 1: a small bird sitting on a branch in winter.  
Sent. 2: A small bird perched on an icy branch. |
| 24 | 3.0 | Sent. 1: It depends entirely on your company and your contract.  
Sent. 2: It depends on your company. |

Table C.3.2: Number of Transformer layers used for example inputs from the task’s test sets with our Shared/Meta CAT with a tolerance level of $\epsilon = 0.1$
Bibliography


[241] Kaylyn Jackson Schiff, Daniel Schiff, and Natália S Bueno. The liar’s dividend: How deepfakes and fake news affect politician support and trust in media.


220


