Evidence-based Factual Error Correction

James Thorne
Department of Computer Science
University of Cambridge
jt719@cam.ac.uk

Andreas Vlachos
Department of Computer Science
University of Cambridge
av308@cam.ac.uk

Abstract

This paper introduces the task of factual error correction: performing edits to a claim so that the generated rewrite is better supported by evidence. This extends the well-studied task of fact verification by providing a mechanism to correct written texts that are refuted or only partially supported by evidence. We demonstrate that it is feasible to train factual error correction systems from existing fact checking datasets which only contain labeled claims accompanied by evidence, but not the correction. We achieve this by employing a two-stage distant supervision approach that incorporates evidence into masked claims when generating corrections. Our approach, based on the T5 transformer and using retrieved evidence, achieved better results than existing work which used a pointer copy network and gold evidence, producing accurate factual error corrections for 5x more instances in human evaluation and a .125 increase in SARI score. The evaluation is conducted on a dataset of 65,000 instances based on a recent fact verification shared task and we release it to enable further work on the task.[1]

1 Introduction

Fact verification is the task of predicting whether claims are true or false using evidence. With the availability of a number of resources (Wang, 2017; Karadzhov et al., 2017; Thorne et al., 2018; Augenstein et al., 2019; Wadden et al., 2020), the task has attracted significant attention and spawned the development of new models, architectures and approaches. With potentially sensitive applications, recent works have focused on building explainable variants of fact checking (Atanasova et al., 2020; Stammbach and Ash, 2020; Kotonya and Toni, 2020). Exposing the evidence source and

---

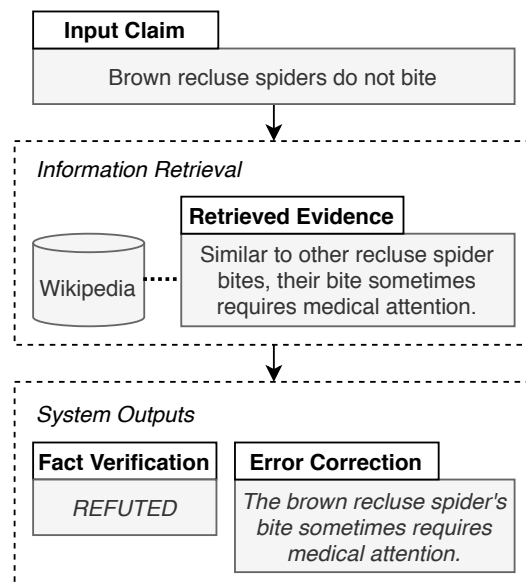[1] https://github.com/j6mes/2021-acl-factual-error-correction

Figure 1: Factual Error Correction uses evidence to make corrections to claims, in contrast to fact verification, which instead classifies the veracity of the claim.

decision making process may help the reader uncover subtle issues that cause automated systems to fail. Additionally, using such evidence to continuously update news articles as facts change forms part of the vision outlined by Cohen et al. (2011) for automated newsrooms.

In this paper, we propose *Factual Error Correction*, as an explainable alternative for fact verification. Rather than merely assigning a truth label, possibly accompanied by evidence, our goal is to rewrite claims so that they are better supported by the retrieved evidence. For example, in Figure 1, a claim that would be REFUTED by the evidence using a fact verification system is rewritten so that it becomes supported by evidence retrieved from Wikipedia. This work extends fact guided sentence modification (Shah et al., 2020), which uses short factoid claims to introduce changes to Wikipedia passages. However, they assume that the claim and

Wikipedia text are always incongruous and require a meaning-altering change, our proposal makes no assumptions over the veracity, and is applicable to claims both supported and refuted by evidence. Additionally, we incorporate a retrieval component to select evidence for a given claim from a corpus (in our case, Wikipedia) rather than requiring gold standard evidence to be explicitly provided.

A challenge for factual error correction is the lack of datasets consisting of claims paired with their corrections. However, with recent developments in fact checking, there is an abundance of new datasets consisting of claims paired with evidence. To address this data scarcity, we make use of distant supervision to incorporate retrieved evidence into generating the corrections.

We release a dataset of 65,000 claims, containing the intermediate annotations from FEVER (Thorne et al., 2018). These consist of factoid sentences that were used to construct the supported and refuted claims in the dataset, and use these as reference targets for automated evaluation. We further verify the findings through a final round of annotation using human raters. Our evaluation finds high correlation between manual scores and the SARI metric (Xu et al., 2016) and our best performing distantly-supervised system generated corrected claims for 24% of instances when using retrieved evidence, with a SARI Final score of .419. A fully-supervised system with gold evidence generated corrections for 69% of instances, indicating plenty of opportunities for future work to extend our contributions.

2 Related Work

A number of related works offer methods to make corrections to sentences. However, their use of external information differs. This can be placed on a continuum from only using the knowledge captured during language model pre-training, to conditioning generation based on a context sentence. We briefly outline key methods and approaches below.

Grammatical Error Correction (GEC) (Knight and Chander, 1994; Han et al., 2010; Ng et al., 2014) is the task of making *meaning-preserving* changes to sentences such that grammatical errors made by language learners are removed. No external information is required as the sentence is undergoing a surface-level transformation where the (intended) semantic content of the sentence should remain unchanged.

In contrast, the semantic content of sentences

undergoing *factual* error correction will be altered, if needed, to better align the meaning with ground truth evidence. Shah et al. (2020) make meaning-altering updates to sentences in Wikipedia in a two step process that does not require reference corrections in training: salient tokens are masked and a corrector conditionally replaces the masks with ground truth evidence. In this approach, token salience is predicted by querying a model that is trained to perform fact verification for a claim against evidence. Cao et al. (2020) generate corrections as a post-editing step for outputs from abstractive summarization so that they are consistent with the source text. Their approach uses a sequence-to-sequence model trained to restore artificially generated corruptions of a reference summary.

One potential way to introduce knowledge is to use information stored in the parameters of large-scale pre-trained language models (Petroni et al., 2019). The language model can be used recover tokens responsible for causing factual errors that are masked out as a variant of cloze-style evaluation (Taylor, 1953). While such approaches have been employed for fact verification (Lee et al., 2020), these approaches share the following limitations. Without explicit control (Nie et al., 2019), the most likely token when decoded may not be factually accurate, or supported by the retrieved evidence, commonly referred to as a hallucination (Rohrbach et al., 2018; Zhou et al., 2020). Furthermore, *even if* the information stored within language model parameters could be reliably retrieved for factual error correction, facts change over time and the need to obtain information from up-to-date sources becomes greater as the state of the world diverges from the information captured within the model parameters. Recent language models augmented with a retrieval component such as REALM (Guu et al., 2020) and RAG (Lewis et al., 2020) could be applied, however, task-specific fine-tuning would still be required to condition the generation based on the factual error to mitigate hallucination.

3 Task Definition

Training Let a claim c be the input sentence undergoing correction to yield c'. The correction requires incorporating knowledge from retrieved evidence $E(c)$ such that c' is supported by this evidence, $E(c) \vDash c'$. Factual error correction is subject to the following 3 requirements:
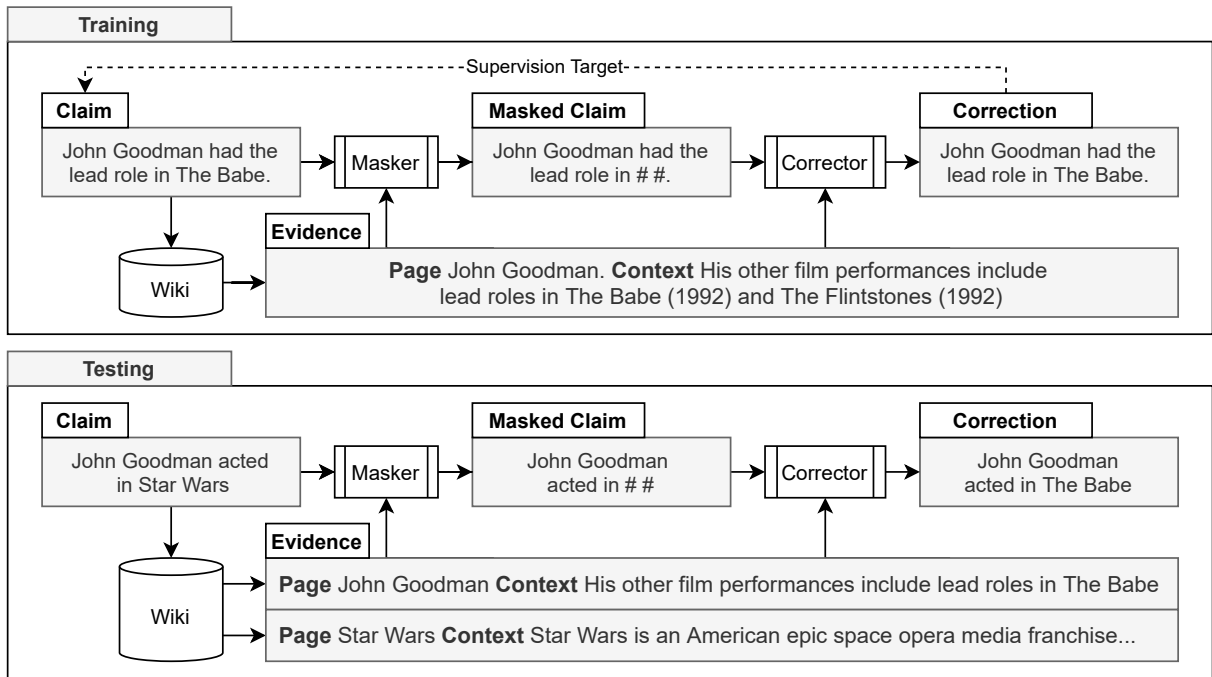
Figure 2: The corrector is trained to reconstruct masked claims, conditioned on retrieved evidence, indicated by the dashed arrow. At test time, the corrector is able to incorporate new facts from the evidence to generate corrections.

R1 - Intelligible Similar to other language generation tasks, our first requirement is that generated outputs are fluent and intelligible. They must be free of grammatical mistakes and the meaning must be understandable without the aid of additional context or evidence so that their factual correctness can be assessed.

R2 - Supported by Evidence The generated correction must be supported by the retrieved evidence. This property follows from previous work (Thorne et al., 2018) and also requires models to condition generation on the retrieved evidence – penalizing models that hallucinate (Holtzman et al., 2020).

R3 - Error correction Specific to our task, the corrections should be targeted to the errors present in the inputted claim. While this, in part, can be assessed by R2 we need to compare the correction to the inputted claim to ensure the output is not introducing new unrelated information. For example, an erroneous claim: *France is in South America* could be supported by evidence if it were rewritten as *France is a republic*. However, the desired correction should instead state *France is in Europe*.

4 Task Decomposition

The choice of supervision for the error correction system influences the task decomposition. For example, with full supervision, the system can be constructed with an information retrieval module and a sequence-to-sequence module that conditionally generates a correction given the claim and evidence. However, large datasets of claims paired with corrections are not available. The absence of full supervision requires that we distantly-supervise our systems using *fact verification* datasets, which are an abundant resource. Fact verification datasets contain claims labeled with evidence but do not contain corrections. With this resource, we propose a task decomposition that generated corrections by training models to reconstruct claims with masked tokens using retrieved evidence.

4.1 Distantly-supervised corrections

Test time Corrections are generated by a two-stage process, illustrated in Figure 2. Tokens from the claim, c, are first masked, yielding $\tilde{c}$, and then input to the corrector $c' = Corr(\tilde{c}, E(c))$. The masker, $\tilde{c} = Mask(c, E(c))$, replaces a subset of tokens in the claim with a blank placeholder, conditioned on $E(c)$. Its purpose is to remove tokens that are salient to the claim being supported or refuted by the evidence. Using the masked claim, $\tilde{c}$, the corrector replaces the blank placeholders with tokens conditionally generated using retrieved evidence. To correct errors, evidence refuting a claim ($E(c) \nvDash c$) conditions generation of a correction supported by it $E(c) \vDash c'$. This extends the pro-

tocol Shah et al. (2020) by conditioning both the masker and corrector with multiple retrieved evidence sentences, rather than a single gold factoid.

Training the corrector   Similar to masked language modeling, the training objective is to generate the input claim $c' = c$ conditioned on the masked claim $\tilde{c}$ and evidence $E(c)$. By training the model to generate the input claim, we expect the model to generate the input claim only if it was in complete agreement with the evidence (assuming the masking and the evidence are correct). Otherwise, the generated correction will contain evidence pertinent to the correcting the masked claim, which enables us to generate corrections satisfying requirements R2 and R3.

Masker   When applied to factual error correction, masking the tokens from the claim acts as a proxy to which tokens need to be removed to correct an error. Parallels can be drawn between masking and generating token-level explanations. We briefly summarize common approaches to generating explanations in Section 5.2.

5 Model

5.1 Evidence retrieval

We use GENRE (Cao et al., 2021) and Dense Passage Retrieval (Karpukhin et al., 2020) together to retrieve evidence for claims $E(c)$. Both have shown success for a number of language understanding tasks over Wikipedia (Petroni et al., 2020). GENRE is a pre-trained seq2seq model, trained to predict a Wikipedia page name for a claim. DPR encodes fixed length passages from Wikipedia into vectors using a BERT encoder to build a static index. At test-time, the claim is encoded and the most-similar passages are returned using an inner-product search. We return the top-k passages returned by DPR from pages predicted by GENRE.

5.2 Token-level explanations as masks

At test time, the purpose of the masker is to selectively remove tokens that contribute to the factual errors within a claim. We study how the choice of masker influences the quality of corrections. This considers varying levels of access to model information and different run-time complexity. Both the black- and white-box methods, outlined below, require querying a model trained to classify the veracity of claims given evidence whereas the the language model masker and baselines do not.

Black-box masker   We evaluate perturbing the input to a classifier that is trained to predict the veracity of a claim given evidence. We use LIME (Ribeiro et al., 2016), a diagnostic that trains a locally linear model to score the importance of input features (in our case, tokens in the claim) with respect to the predicted labels. The model under test is a BERT classifier where evidence and the claim are concatenated in the input. This is referred to as *black-box* because the model does not undergo modification and no information about internal values or states is exposed.

White-box masker   In contrast, to obtain *white-box* model explanations, the model has undergone modification to expose internal information. We use the Neutrality Masker from (Shah et al., 2020) to predict which tokens, when masked, are likely to cause a label flip from supports or refuted to not enough information. This masker exposes encoded input of an ESIM classifier (Chen et al., 2017), and adds a linear classifier over the hidden states to predict per-token masking probability. At test time, masks can be generated through a single query to the model (unlike LIME in the black-box masker which requires multiple queries to the model), however this requires an additional step to train, using predictions from the classifier as signal.

Language model masker   We evaluate whether it is possible to generate masks without the need for a fact verification model. We use a BERT pre-trained language model (Devlin et al., 2019) to measure the surprisal of tokens in the claim. Our intuition is to identify tokens which introduce misinformation under the hypothesis that the world knowledge (Petroni et al., 2019) captured in retraining would assign lower probabilities to tokens contradictory to the world state. This language model has no additional task-specific fine-tuning. We independently predict the cross-entropy for each token under a masked language modelling objective using BERT and return the top-k tokens.

Baselines   We additionally consider two simple baseline maskers: **random** masking of a subset of tokens and also a **heuristic** method of masking tokens which are not in common between the claim and the retrieved evidence.

5.3 Corrections

We train an encoder-decoder transformer model to generate corrections from masked claims and

evidence. Our model uses a pre-trained T5 transformer (Raffel et al., 2020) which we fine-tune with the distant supervision protocol described in Section 4.1. This model jointly encodes the masked claim and evidence by concatenating these two inputs in the input.

We also compare against a baseline model from a related task of fact guided sentence modification (Shah et al., 2020) which uses a pointer generator network (See et al., 2017). Unlike our model, which captures long-range dependencies between claim and evidence through the transformer self-attention (Vaswani et al., 2017), the baseline independently encodes the evidence and masked claim using LSTMs (Hochreiter and Schmidhuber, 1997) before decoding using a pointer-copy mechanism.

In order to evaluate the impact of conditioning on evidence, we decode tokens from masked claims using a language model without fine-tuning or conditioning, similar to the Language Models as Knowledge Bases hypothesis introduced by Petroni et al. (2019). This would consider correcting claims using the implicit knowledge stored within the model parameters rather than using external evidence.

6 Data

We make use of FEVER (Thorne et al., 2018), a commonly used fact verification dataset, as the basis for our experiments. FEVER is one of the largest resources consisting of claims paired with evidence from Wikipedia. There are 185k instances with corresponding evidence sentences and a label as to whether the claim is SUPPORTED or REFUTED by it. Claims where no information could be found are labeled as NOTENOUGHINFO.

To comprehensively evaluate the corrections generated manual evaluation is required. However, this is expensive and not suitable for system development and hyper-parameter optimization. To automate system evaluation or to train a seq2seq model with full supervision, a reference "gold standard" correction is also required. For this, we release annotations from the FEVER shared task as follows. The claims in FEVER were generated in a two-stage process: annotators extracted facts from Wikipedia and then performed meaning altering perturbations called *mutations* over these extracted facts. Each claim was independently labeled using retrieved evidence. Our reference corrections are the *unmodified* facts extracted from Wikipedia.

The class balance and size of the dataset is re-

ported in Table 1. The training and test splits are disjoint by entity. The additional hidden shared task test set was not used. The claims labelled as NOTENOUGHINFO. are used for training fact verification classifiers, but they will not be used for training the error correction systems in this paper as there is no labeled evidence to make corrections from. For completeness, we also release these unused NOTENOUGHINFO instances, as they have claims paired unmodified extracted facts (21934 training, 1870 development and 2037 test).

Label	Instance Count		
	Train	Validation	Test
Supports	37961	1477	1593
Refutes	20075	2091	2289
Total Training	58036	3568	3891

Table 1: Instance counts by class and dataset partitions

7 Evaluation

While it's convenient to use an automatic metric during development, these metrics compute token overlap against a single reference sentence and cannot capture the nuances required to assess the veracity of the generated corrections against evidence. Thus, our primary evaluation will use human raters to label whether the model predictions meet the task requirements stated in Section 3.

Human raters are asked three questions about system outputs to assess whether the corrections meet the requirements of intelligibility, supported by evidence, and error correction introduced in Section 3. For the first 2 requirements, the question has a binary answer. For the third requirement of error correction, the question has 3 answer choices: (1) the information content w.r.t. the evidence improved, (2) information unrelated to the claim was added (i.e. the claim was ignored), (3) no correction was needed (i.e. the claim was already supported by evidence). The raters were shown each question in this sequence without knowledge of which system generated the correction. Negative answers to a question automatically assigned negative answers to subsequent ones (prescribing that an unintelligible sentence could not contain a fact supported by evidence or introduce a correction). 20% of the tasks are assigned to two raters to measure inter-annotator agreement. We used 4 expert participants from our lab (none of them co-authors of the paper) who were familiar with fact verifica-

tion, but not with error correction. Responses were calibrated using a pilot study on the validation set.

For automated evaluation, we use SARI (Xu et al., 2016) which is a metric used for sentence simplification. SARI considers ngrams retained from the source as well added or deleted ngrams through comparison against a reference sentence. We additionally report BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) to indicate precision and recall of the correction. In Section 9, we report correlation of automated metrics against our manual evaluation.

8 Implementation

T5 Masker-Corrector We fine-tuned the T5-base pre-trained models released by HuggingFace (Wolf et al., 2020). The number of training epochs and learning rate was selected through optimizing the overall SARI score. The search space for learning rate was $\{10^{-5}, 5 \cdot 10^{-5}, 10^4, 5 \cdot 10^{-4}\}$. We used $5 \cdot 10^{-5}$ for all experiments. We found diminishing returns in SARI after 4 epochs and stopped training.

Fully Supervised Ceiling We use this model to estimate the ceiling performance of a factual error correction system (assuming a reasonable amount of training data is available) that other methods can be compared against. We fine-tune a T5-base model with supervision of the correction (see Section 6), using the same hyper-parameter choices as the T5 Masker-Corrector.

Automated Scoring A single reference sentence from the FEVER dataset is used for automated scoring. We consider BLEU, ROUGE, and SARI. SARI considers the F1 of added tokens, F1 of kept tokens, precision of deletions, and the mean of these 3 scores (denoted *final*). We use code made available by Xu et al. (2016).

Evidence Retrieval We use the Facebook implementation of DPR (Karpukhin et al., 2020) without fine-tuning and constructed an index over the Wikipedia version released with FEVER (Thorne et al., 2018), chunked into passages of 50 tokens. For GENRE, the original authors' implementation was used. We selected the top matching 2 passages. This resulted in the highest scores on the downstream corrections; SARI was lower when using 1 or 3 passages.

Maskers For the white-box masker, we use the implementation provided by Shah et al. (2020) applied to our dataset retaining original hyper-parameters trained on FEVER. For the black-box masker, we use the LIME implementation from (Ribeiro et al., 2016) to probe a BERT classifier (Devlin et al., 2019) fine-tuned on FEVER. For the LM and random baseline maskers, where the number of masks was tunable, we masked 50% of the tokens, which was similar to the number of tokens masked by the black- and white-box maskers.

Language Model as Correctors? We greedily decode masked tokens using a BERT-base-cased language model using the HuggingFace implementation (Wolf et al., 2020) without fine-tuning.

Comparison to Previous Work For comparison to previous work, we use the dual-encoder pointer network implementation from (Shah et al., 2020), retaining the original hyper-parameter choices.

9 Results

We first report results from a manual evaluation, assessing the requirements that corrections are intelligible, supported by evidence, and improve the factuality of the claim, as listed in Section 3. Our evaluation considers a sample of 200 instances per system. We report the results in Table 2. For inter-annotator agreement control, 20% of instances were annotated by two annotators: the Cohen's κ scores for the 3 questions are 0.92 for intelligible, 0.92 for supported, and 0.86 for corrected. When using retrieved evidence, the white-box masker generated no masks for 41% of instances. Without masked tokens, the T5 corrector copied the input claim to the output. This fits the assumption that, if the claim is already supported well by evidence, no correction is required.

The fully supervised models had the highest rate of satisfactory corrections that improved the factuality of the claim (requirement 3), indicating a performance ceiling for the distantly-supervised models. Incorporating retrieved evidence in these supervised models (rather than gold) reduced the number of corrections supported by evidence from 88.9% to 64.7% and the number of satisfactory corrections from 68.9% to 48.9% showing the challenges of incorporating (possibly noisy) retrieved evidence when generating the corrections.

When using the masker and corrector distant supervision strategy, different maskers could be used

System	Evidence	Training Masks	Test Masks	Aggregated Score (%)		
				Intelligible	Supported	Corrected
T5 Fully Supervised	Gold	-	-	98.9	88.9	68.9
T5 Fully Supervised	Retrieved	-	-	97.7	64.7	48.9
T5 Masker + Corrector	Retrieved	Random	Heuristic	89.3	**57.9**	**40.0**
T5 Masker + Corrector	Retrieved	Heuristic	Heuristic	90.0	38.0	20.0
T5 Masker + Corrector	Retrieved	Random	Black-box	**93.1**	42.2	24.0
T5 Masker + Corrector	Retrieved	Black-box	Black-box	91.4	37.0	19.8
T5 Masker + Corrector	Retrieved	White-box	White-box	90.6	41.7	23.9
BERT Language Model	-	-	Heuristic	48.0	20.7	15.0
BERT Language Model	-	-	Black-box	30.1	4.9	3.4
Shah et al. (2020) M+C	Gold	White-box	White-box	32.2	10.7	5.0

Table 2: Aggregated scores from human evaluation considering intelligibility, whether generated instances were supported by evidence and errors corrected.

to train the corrector to the masker used at test time. We observed that training the corrector with random masks yielded both a higher rate of satisfactory corrections and corrections supported by evidence when using either the black-box or heuristic masker at test time. We further evaluate other maskers with automated metrics in Section 9.2.

Using a heuristic masker at test time, which removed tokens from the claim not present in the evidence, generated more claims meeting the supported and corrected requirements than masks generated by querying a fact verification model (both black-box and white-box). An analysis of the masker's influence on the corrections is provided in Section 9.1. The two baseline systems, Dual Encoder M+C, based on Shah et al. (2020), and a pre-trained BERT language model, generated corrections that were intelligible or supported by evidence at a lower rate than the aforementioned models, further discussed in Sections 9.3 and 9.4.

We report the correlation between automated scoring metrics and our manual evaluation in Table 3. The KEEP component of SARI, which measures the F1 of n-grams from the claim retained in the output, had the highest correlation with all three requirements. Overly aggressive maskers which remove too much content from the claim can result in unintelligible outputs, or corrections unrelated to the claim. ROUGE2, which measures the recall of bigrams in the correction w.r.t. the reference, exhibited reasonable correlation to the manual evaluation against the supported and corrected requirements, however does not correlate as well with intelligibil-

ity. The ADD and DELETE components of SARI provide further information but do not correlate as strongly with the human judgements. Having only one reference correction reduces the utility of precision-oriented metrics, like BLEU, as valid corrections can differ from the reference.

Metric	Correlation (Pearson r)		
	Intelligible	Supported	Corrected
SARI Keep	**.87**	**.95**	**.93**
SARI Final	.78	.92	.91
SARI Delete	.72	.82	.91
SARI Add	.52	.84	.79
ROUGE2	.75	.90	.91
ROUGE1	.71	.87	.88
BLEU2	−.05	.32	.45
BLEU1	−.46	−.10	.05

Table 3: Both SARI and ROUGE automated scoring metrics have high correlation to manual evaluation.

9.1 Choice of masker

When training the corrector with the same masker that is used at test time, both the heuristic and black-box maskers yielded comparable scores under human evaluation. Inspection of SARI breakdown in Table 4 indicates that more tokens were kept when using the heuristic masker (Keep=.651) whereas the black box model was more aggressive in masking, resulting in less information from the claim being retained (Keep=.594). This correlated well with human judgements as more information retained gives a richer context for generating the correction and prevents erasure of claims already (partially) supported by the evidence.

Both the black-box (LIME) and white-box (the masker from Shah et al. (2020)) methods require querying a veracity classifier to generate the masks. Using retrieved evidence for the veracity classifier, which was used to generate the masks in conjunction with these two methods, had a negative impact on most components of the SARI score. For the black-box masker, using retrieved evidence reduced the number of masked tokens from an average of 4.7 per claim to 3.9. Whereas the number of masked tokens by the white-box masker remained unchanged at 4.7 (approximately 50% of number of tokens in the claim). Most notably, the white-box method of mask generation (row 4 in Table 4) did not to generate masks for 41% of instances when using retrieved evidence, whereas all instances had at least one mask when using gold evidence – an artefact of the noise introduced by retrieval.

Masker	SARI Score			
	Keep	Delete	Add	Final
Black-box (Gold)	.630	**.582**	.088	.433
White-box (Gold)	**.652**	.559	**.128**	**.447**
Black-box (IR)	.594	.526	.090	.412
White-box (IR)	.628	.535	.107	.426
Heuristic (IR)	.651	.574	.041	.422
Masked LM	.538	.509	.062	.370
Random	.619	.475	.087	.390

Table 4: Extrinsic evaluation of maskers, varying the use of evidence when generating the masks, evaluated using the T5 Masker+Corrector model.

9.2 Corrector trained with random masks

Generating large quantities of masked training data through querying a model, such as with the black-box model explanation techniques, can be computationally expensive. In contrast, random masks can be generated without querying a model. Using a corrector trained on random masks resulted in higher quality outputs at test time when paired the black-box and heuristic maskers. Training with random masks promotes good exploration of the task. In contrast, while the black-box and heuristic approaches worked well during testing, correctors *trained* on these maskers generated worse outputs due to the limited exploration of the task space. Additionally, generating training data using the black- and white-box methods requires making predictions using the model's training data which may result in different outcomes to making predictions on unseen test data.

Masker	SARI Score			
	Keep	Delete	Add	Final
Black-box (Gold)	.618	.622	.102	.447
White-box (Gold)	.640	.570	.114	.441
Black-box (IR)	.611	.543	**.194**	.419
White-box (IR)	.618	.590	.144	.452
Heuristic (IR)	**.652**	**.627**	.155	**.478**
Masked LM	.561	.529	.078	.389

Table 5: Using random masks at training resulted in higher scores when testing with different maskers

9.3 Comparison to previous work

Previous work uses a dual encoder pointer network (Shah et al., 2020) to make corrections, reported in Table 6. The corrector tended to copy portions of claim rather than correct it, resulting in a SARI KEEP score of .452 which is lower than the T5 model using the same white-box masker (Table 4). Human evaluation considered these corrections mostly unintelligible, even when using gold evidence (Table 2). This was especially the case for rarer entities. Hyper-parameter tuning of the corrector's coverage ratio, as suggested by the authors, did not yield improvements.

System	SARI Score			
	Keep	Delete	Add	Final
Dual Enc Ptr (Gold)	**.452**	**.569**	**.039**	**.353**
Dual Enc Ptr (IR)	.345	.481	.017	.281

Table 6: Results using a dual encoder pointer network (Shah et al., 2020) were low, despite the strong masker.

9.4 Language Models as Correctors?

With the exception of the heuristic masker, using a pre-trained language model, without fine-tuning, to correct claims resulted in low SARI scores (Table 7). Without conditioning on the evidence, the correction is not related to the claim or supported by evidence to verify the claim, which is indicated by the low SARI Add scores which consider the precision of the added tokens. As these maskers deleted most tokens, retaining only stop-words, decoding most likely tokens without a prompt or context tokens resulted in unintelligible outputs. For the heuristic masker, more content words were retained yielding more intelligible outputs. However, these were not always supported by evidence, indicated in the human evaluation in Table 2.

Masker	SARI Score			
	Keep	Delete	Add	Final
Masked LM	.360	.472	.019	.289
Heuristic (IR)	**.629**	**.651**	**.034**	**.438**
White-box (IR)	.232	.446	.005	.228
Black-box (IR)	.364	.003	.001	.122

Table 7: Correcting claims using a language model does not condition the generation on evidence.

10   Qualitative Error Analysis

In this section we discuss the following issues which were present in all master-corrector systems:

Over-erasure   In some instances, the masker removed most or all of the non-stopword tokens from the claim. This resulted in the original meaning of the claim being erased. Without this information the corrector could not reconstruct the claim, resulting in corrections that were unrelated to the input claim. This issue was most prevalent for the black-box masker, where 15% of instances had more than 5 consecutive tokens masked and 32% of instances had 4 consecutive tokens masked. In contrast, the heuristic masker, which identifies the tokens not present in the retrieved evidence had 5 consecutive tokens masked for 3% of instances and 4 consecutive tokens masked for 9% of instances. While, in some cases, appropriate corrections could be made despite the aggressive masking (e.g. the claim "Exit the King is by man[sic]." was fully masked, but corrected to include the author's name), others were re-written focusing on a different fact, e.g. a claim about the length of reign of Maria Theresa was rewritten to be about her date of birth.

Incorrect masking   When the erroneous tokens in a claim were not masked, the corrector would generate outputs not supported by evidence. For example the following claim, which has an incorrect year, was masked but retaining the error: "Ghost, the film was released in 1994" as "[MASK] , [MASK] [MASK] [MASK] [MASK] [MASK] in 1994". Even with suitable retrieved evidence, indicating the release year is 1990, no appropriate correction could be made.

Inadequate evidence retrieval   Where the evidence retrieved was related, but not specifically supporting or refuting the claim, the generated corrections were vague: the claim "Poldark aired on HBO" was corrected to "Poldark premiered on TV" as the evidence lacked the name of the correct TV station. Similarly, where incorrect masks were made, additional retrieval retrieval may be required to prevent the corrector from hallucinating information to cover the knowledge missing from the evidence. For example, the name of the TV show was masked in the claim "Two and a half men starred Jamie Fox[sic]", but as no mention of Jamie Fox was present in the evidence, the model hallucinated a different TV show name.

11   Conclusions and Future Work

Going beyond simply identifying errors, factual error correction presents a number of challenges for information retrieval, fact verification and abstractive summarization communities alike. In this paper, we demonstrated that the task can be performed with distant supervision in the form of claims labeled by evidence supporting or refuting them. However, there are a number of outstanding challenges that must be addressed. The data we used from the FEVER task was re-purposed to evaluate whether systems can undo mutations introduced by human annotators and may not be representative of the range of factual errors that would be present in real-world documents. While some automated metrics correlated well with human judgements, future work should consider how automated scoring can be better used to discriminate the adequacy of the generated corrections going beyond similarity to the reference sentence. From a modelling perspective, the masks strongly influenced the corrector and further work is required to generate masks that result in better corrections. We observed where masks mismatched the evidence, the correction was vague, hallucinated or did not correct the factual errors in the claim. This could be addressed through joint training of both components to enable them to avoid error propagation from masking to correction.

Acknowledgements

Broader Impact Statement

Our experiments were performed on publicly available data about common facts from Wikipedia. These data are released under a creative-commons license. The expert raters from our lab who manually reviewed the generated instances were volunteers and were compensated through quid-pro-quo help on their own projects.

The intended use of this project is to help explain reasoning using evidence, going beyond single-label classification. This adds an additional safeguard, making the decision process more transparent as poor predictions by our model expose limitations that would be hidden by classification. Our data is synthetic in nature and is biased towards synthetic facts from popular entities. Application to political or scientific domains would require additional work. Misinformation about populations that are under-represented in our data may not be accurately identified or corrected without further mitigation. One positive finding in our paper was that some of biases perpetuated in the hallucinations of language models were mitigated when conditioning the generation on retrieved evidence.

Model fine-tuning took approximately 2 hours per experiment on a single P100 GPU. Generating LIME explanations of the training dataset took approximately one day – motivating our experiments that used models trained on random or heuristic maskers which required fewer resources by several orders of magnitude.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364. Association for Computational Linguistics.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4685–4697, Hong Kong, China. Association for Computational Linguistics.

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual Error Correction for Abstractive Summarization Models. In *Empirical Methods in Natural Language Processing*, pages 6251–6258.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Sarah Cohen, Chengkai Li, Jun Yang, and Cong Yu. 2011. Computational Journalism: a call to arms to database researchers. *Proceedings of the 5th Biennial Conference on Innovative Data Systems Research (CIDR 2011) Asilomar, California, USA.*, (January):148–151.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training.

Na-Rae Han, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using an error-annotated learner corpus to develop an ESL/EFL error correction system. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. Fully automated fact checking using external sources. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 344–353. INCOMA Ltd.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of the National Conference on Artificial Intelligence*, volume 1, pages 779–784.

Neema Kotonya and Francesca Toni. 2020. Explainable Automated Fact-Checking for Public Health Claims. In *The 2020 Conference on Empirical Methods in Natural Language Processing*.

Nayeon Lee, Belinda Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language models as fact checkers? In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, volume 2, pages 36–41. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, July, pages 1–14. Association for Computational Linguistics.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. 2019. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, July, pages 311–318. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2020. KILT: a Benchmark for Knowledge Intensive Language Tasks.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21:1–67.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, volume 39, pages 97–101. Association for Computational Linguistics.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083. Association for Computational Linguistics.

Darsh J Shah, Tal Schuster, and Regina Barzilay. 2020. Automatic Fact-guided Sentence Modification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dominik Stammbach and Elliott Ash. 2020. e-FEVER: Explanations and Summaries for Automated Fact Checking. In *Proceedings of the 2020 Truth and Trust Online Conference (TTO 2020)*, page 32. Hacks Hackers.

Wilson L. Taylor. 1953. "Cloze Procedure": A New Tool for Measuring Readability. *Journalism Quarterly*, 30(4):415–433.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Lilon Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *31st Conference on Neural Information*

Processing Systems (NIPS 2017), Long Beach, CA, USA.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Chunting Zhou, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting Hallucinated Content in Conditional Neural Sequence Generation. pages 1–21.