Factual Error Correction of Claims

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 supported by evidence. This serves two purposes: firstly this provides a mechanism to correct written texts that contain misinformation, and secondly, this acts as an inherent explanation for claims already partially supported by evidence. We demonstrate that factual error correction is possible without the need for any additional training data using distant-supervision and retrieved evidence. We release a dataset of 65,000 instances, based on a recent fact verification dataset, to compare our distantly-supervised method to a fully supervised ceiling system. Our manual evaluation indicates which automated evaluation metrics best correlate with human judgements of factuality and whether errors were actually corrected.

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., 2018a; 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 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 vision outlined by Cohen et al. (2011) for automated newsrooms. Understanding how to make changes to documents based on updated evidence would help fulfill this vision.

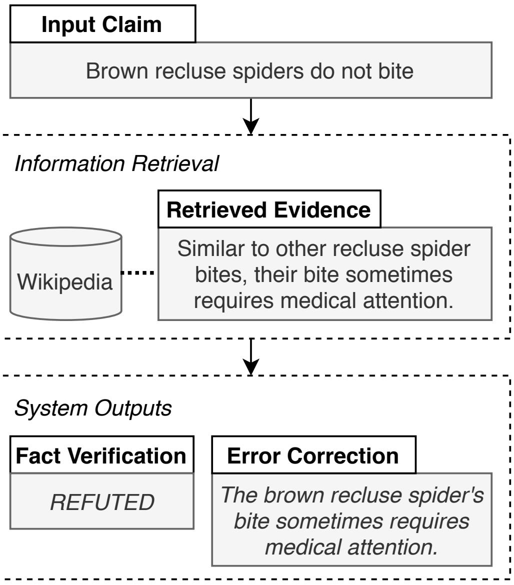
Input Claim
Brown recluse spiders do not bite
Information Retrieval
Wikipedia
Retrieved Evidence
Similar to other recluse spider bites, their bite sometimes requires medical attention.
System Outputs
Fact Verification
REFUTED
Error Correction
The brown recluse spider's bite sometimes requires medical attention.

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.

In this paper, we propose *Factual Error Correction*, as an explainable alternative for fact verification. Rather than merely finding evidence or assigning a label, 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) in two ways. Firstly, we incorporate a retrieval component to select evidence for a given claim from a corpus of texts (in our case, Wikipedia). This both demonstrates viability for open-book applications, and represents an advance over previous work in which gold standard evidence is explicitly provided. Secondly, our proposal is to offer a correction regardless of veracity (i.e. for claims supported by evidence, not only refuted). This summarizes the evidence with respect

Preprint: work in progress

to the claim, helping to resolve half truths, and better informs the end-user of the decision making process, regardless of label.

Our primary contribution is the new task. For this, we evaluate a distantly supervised baseline system for factual error correction using retrieved evidence from DPR (Karpukhin et al., 2020) and a T5 seq2seq transformer (Raffel et al., 2020). We compare to full supervision and a system from a related task, finding that the addition of evidence retrieval adds a significant challenge – especially when using distant supervision. Our evaluation finds a high correlation between human raters and the SARI metric, introduced by Xu et al. (2016), to evaluate document simplification, which we adopt for automated scoring for system development.

Finally, we release a dataset of 65,000 claims, extending an existing fact verification dataset (Thorne et al., 2018a) to train and factual evaluate error correction systems. All claims are paired with evidence and a true fact used to generate the claim through a meaning altering or meaning preserving mutation, further discussed in Section 6.

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 no external information at all, to knowledge stored within model parameters, to incorporating external information. 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 knowledge-source is required as the meaning of the sentence is undergoing a surface-level transformation where the (intended) semantic content of the sentence should remain unchanged. Recent works model GEC as a sequence-to-sequence task (Yuan and Briscoe, 2016), similar to machine translation. In contrast, the semantic content of sentences undergoing *factual* error correction will be altered, if needed, to better align the meaning with is presented in the evidence. It is a requirement that the generated corrections both remove the error from the original claim and be supported by the evidence in the generated correction.

One potential way to introduce external knowledge when performing error corrections would be to use information stored in the parameters of large-scale pre-trained language models (Petroni et al., 2019). In this case, making corrections can be modeled as a variant of cloze-style evaluation (Taylor, 1953) – using the language model to fill in masked tokens. While such approaches have been employed for fact verification (for example, Lee et al. (2020) used a BERT language model (Devlin et al., 2019)), these approaches share the following limitations. When masked language models are used to correct claims, the underlying task is to predict the most likely token to replace a mask. 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 expressed as text when making corrections, facts change over time and the need to read information from external evidence 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 RAG (Lewis et al., 2020) and REALM (Guu et al., 2020) could be applied. Fine-tuning may still be required as a language modelling objective alone would be insufficient to condition the generation based on error when replacing the masked tokens.

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. In contrast, Shah et al. (2020) make meaning-altering updates to sentences in Wikipedia, by masking out salient tokens and training a corrector to replace these from 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. Parallels can be drawn between the masker and generating token-level explanations for natural language inference. Natural language inference is a related classification task predicting whether a premise is entailed or contradicted by a hypothesis. When generating explanations, salient tokens are highlighted. These could be generated through perturbing the input, (Ross et al., 2017; Li et al., 2016; Ribeiro et al., 2016, 2018, *inter alia*), by interpreting model weights, gradients and attention (Linzen et al., 2016; Li et al., 2016; Jain and Wallace, 2019; Thorne et al., 2019b, *inter alia*) or attention. While these explanations describe model behaviour, their usage alone can lead to a mismatch between which features are important to the model compared to what features humans would judge as important (Ribeiro et al., 2016). Generating corrections should provide a mechanism that helps explain the decision making process beyond than simply assigning a label or highlighting salient tokens.

3 Task Definition

We define factual error correction as the task of making meaning-altering changes to a claim given evidence retrieved from a corpus of documents. For a given claim, a factual error correction system must first retrieve evidence that SUPPORTS or REFUTES it from a corpus of ground truth documents. While in *fact verification* the task is to assign a label to the claim using this retrieved evidence, *error correction* differs as the evidence must be used to remove the error from the claim and rewrite it so that the correction is supported by the evidence. This is outlined by the following 3 requirements:

R1 - Fluency Similar to query focused summarization (Dang, 2005), retrieved evidence passages are combined with respect to the input claim to generate a short text output. While the tasks serve different purposes, there are common requirements and criteria for the generated outputs: notably with respect to grammaticality and focus (focus is also discussed in R3). Our task output is corrected facts. Similar to other language generation tasks, surveyed by Celikyilmaz et al. (2020), our first requirement is that generated outputs are fluent, without grammatical mistakes and that the meaning can be understood without the aid of additional context or evidence.

R2 - Supported by Evidence The generated correction must be supported by the retrieved evidence. We mandate that the outputs must be factually accurate (Holtzman et al., 2019) – requiring the models to condition generation on the retrieved evidence rather than hallucinate new information.

R3 - Error correction Our final requirement is that the corrections remove 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 directly to ensure that the newly generated correction is focused around the errors present, rather than introducing new unrelated information.

3.1 Contrast to fact verification

The related task of fact verification requires systems to predict whether a claim is supported or refuted given retrieved evidence. The input to the task is a claim c and corpus of evidence $\mathcal{E}$. Fact verification systems first retrieve evidence $E \subset \mathcal{E}$, perhaps approximately 1-3 sentences from one or more documents, and use this as input to a classifier predicting whether c is SUPPORTED or REFUTED by this evidence, or that there is NOTENOUGHINFO to say either way.

We can take advantage of recent developments for fact verification as the need to find supporting evidence to generate corrections is a common component between both tasks. The tasks differ in how this evidence is used. Instead of assigning a single label once appropriate evidence has been identified, a fact correction system would generate a corrected version of the claim, c'.

4 Task Decomposition

The choice of supervision for the error correction system influences the task decomposition. With full supervision, the system can be constructed with an information retrieval module, such as TF-IDF (Spärck Jones, 1972) or DPR (Karpukhin et al., 2020) and a sequence-to-sequence module that uses the retrieved evidence and claim as input to conditionally generate the correction. In the absence of full supervision, which would consist of claims paired with their corrections, it is possible to incorporate the retrieved evidence through a distant-supervision strategy. One method we evaluate, extending work from Shah et al. (2020), is to conditionally mask tokens from the claim and replace these – conditioning their replacements using evidence. In this instance, the system can be supervised with pre-existing fact verification datasets (where each instance is a claim, labeled with evidence) without the need for supervision over the target correction in the training data. This approach is outlined in Section 4.1

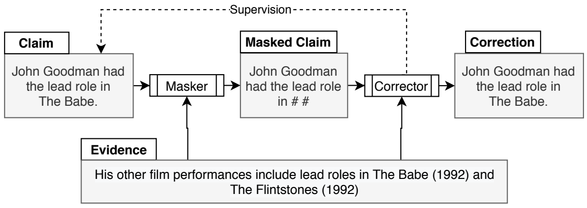
Supervision
Claim
John Goodman had the lead role in The Babe.
Masker
Masked Claim
John Goodman had the lead role in # #
Corrector
Correction
John Goodman had the lead role in The Babe.
Evidence
His other film performances include lead roles in The Babe (1992) and The Flintstones (1992)

Figure 2: Training a corrector to reconstruct masked claims conditioned on the evidence. The target for supervision is the original claim input (dashed line).

4.1 Distantly-supervised corrections

Given the retrieved evidence, a system must remove errors from the claim and generate its correction. It is possible to train a seq2seq model without the need for claims paired with their correction using distant-supervision from a fact verification system. The goal is to incorporate new ground truth facts from the retrieved evidence into a masked version of the claim, assuming that tokens that contribute to factual inaccuracy can be masked out from the claim. We train our factual error correction system to recover masked tokens by conditioning generation from supporting evidence. We first summarize the differences between training and test time for this corrector module, before discussing different masking strategies.

Training the corrector Given a claim, c, the input to the corrector is a masked version of the claim $\tilde{c}$ and retrieved evidence E. Similar to masked language modeling, the training objective is to recover the input claim c prior to masking as illustrated by the dashed arrow in Figure 2. While the knowledge captured by the parameters of masked language models can be used to predict suitable tokens (as is the training objective of the BERT language model Devlin et al. (2019)) our application differs as the generated tokens must be conditioned on the retrieved evidence. At test time, this would then allow a different correction to be generated based on different evidence.

Test time Using the approach outlined above, we have a system that is trained to recover masked tokens in a claim conditioned on retrieved evidence. During training, the retrieved evidence would be supporting the claim. However, at test time, we apply the system to claims refuted by the evidence. The system works under the assumption that tokens in the claim that cause it to be contradicted by the evidence can be appropriately masked out and replaced so that the correction is consistent with the retrieved evidence.

Masker The aforementioned correction system requires masked claims as input. For a claim, which is a sequence of tokens $c = (c_1, \ldots, c_n)$, and set of evidence sentences $e \in E$. The masker replaces a version of the claim, $\tilde{c}$, with a subset of tokens replaced with a blank placeholder. The objective is to generate a mask that forms an explanation of which tokens are salient to a claim being supported or refuted.

To generate masks, previous works make use of a pre-trained fact verification system to highlight which tokens are salient to predicting the veracity of the claim. Such a classifier would trained to predict whether a claim is supported or refuted by evidence. Previous works (Shah et al., 2020) extend the classifier to predict which tokens, when masked, would cause the label to flip to a neutral label (i.e. neither SUPPORTED nor REFUTED).

Generating token level explanations of classification tasks has been widely studied in other NLP disciplines. Each of these approaches expresses a trade-off between run-time complexity (either fine-tuning the model or perturbing the inputs), access to model information (such as model weights and attention), and the quality of the explanations that would become the token masks. We briefly summarize common approaches and their properties in Section 5.3.

5 Model

5.1 Fully supervised ceiling

With full supervision of the claim, evidence and target corrections, factual error correction may be modelled as sequence-to-sequence language task, without the need for masking. For this model, the input is the source claim and evidence and the output is the target correction. To encode both the evidence and claim, we concatenate these into a single string. In this paper, we use an auto-regressive encoder-decoder transformer model, T5 (Raffel et al., 2020), which we fine-tune on the data we introduce in Section 6. We use this model to estimate the ceiling performance of a factual error correction system (assuming sufficient training data is available) that other methods can be compared against.

5.2 Evidence retrieval

We use Dense Passage Retrieval (Karpukhin et al., 2020) to retrieve a set of evidence to condition error correction. DPR encodes passages from Wikipedia into a fixed-size vector using BERT to build a static index. At run-time, the claim is encoded through a separate BERT encoder and the most-similar passages are returned using an inner-product search. This method has shown success for a number of language understanding tasks over Wikipedia (Petroni et al., 2020) when combined with a classifier, even demonstrating a higher accuracy some purpose built systems for the FEVER fact verification task.

5.3 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. This broadly follows recent work on generating token-level explanations for NLI (Thorne et al., 2019a) where salient tokens can be masked out, allowing new evidence to be introduced from the retriever during correction.

We study how the choice of masker at training and test time influences the quality of corrections generated by the factual error correction system. There is a trade-off between the time taken to generate the masks, and how the masked tokens agree with human judgements of token importance (Ribeiro et al., 2016). On one extreme, tokens could be randomly masked, following a similar training strategy to BERT (Devlin et al., 2019) or the unsupervised objective in T5 (Raffel et al., 2020) pre-trained language models. Generating large quantities of training data is possible without having to query an oracle model. On the other extreme, one could perturb the input to an oracle fact verification model that is trained to predict the veracity of a claim given evidence. Querying this *unmodified* model with perturbed inputs using a *black-box* model explanation technique such as LIME (Ribeiro et al., 2016) agrees more strongly with human judgements of important tokens but requires a number of queries to the oracle model to generate the token-level explanation.

In previous work, Shah et al. (2020) predict which tokens, when masked, are likely to cause a label flip to the neutral NOTENOUGHINFO class and use these predicted tokens to mask the input to a corrector. This is analogous to *white-box* model explanations where the model has undergone modification to expose new information from a bi-text classification task in fact verification. Their implementation adds a linear classifier over the encoded input of an ESIM model (Chen et al., 2016), to predict per-token masking probability. At run time, masks can be generated through a single query to the model, however requires an additional training step to supervise. Other methods for "white-box" model explanations expose internal information from the models such as attention. However attention alone may be insufficient to generate quality explanations (Jain and Wallace, 2019).

We experiment with 3 variants of the masker (and 2 baselines) used to generate the training and test data for the downstream correction module as follows:

Black-box masker Tokens are masked from the claim using predictions from a sentence pair classification model trained on instances from the FEVER dataset. This model is a fine-tuned BERT classifier that is representative of the state of the art for the FEVER shared task (Thorne et al., 2018b). We use LIME (Ribeiro et al., 2016) to generate explanations of which tokens from the claim are salient to a sentence pair classifier trained on claims and evidence. This method perturbs instances by masking tokens and uses the change in labels to train a linear classifier that weights the importance of each token.

White-box masker We use the masker from Shah et al. (2020). Tokens are masked using a linear classifier over the encoded inputs. This classifier is supervised to predict which tokens, when masked would cause a label flip from the SUPPORTS or REFUTES labels to NOTENOUGHINFO with additional regularization to control mask size.

Language model masker We evaluate whether it's possible to generate meaningful 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 objective is to identify tokens which introduce misinformation in the claim, Our hypothesis is the world knowledge (Petroni et al., 2019) captured would assign lower probabilities to tokens contradictory to the world state. This language model has no additional task-specific fine-tuning.

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

5.4 Corrections

We train an encoder-decoder transformer model to generate corrections from masked sentences. We experiment with using T5 transformer (Raffel et al., 2020), pretrained with a sequence to sequence masked language modelling objective. While this pre-training objective is very similar to the task of correction, although there is no conditioning on evidence. To condition on evidence, we additionally fine-tuning the T5 transformer to recover masked tokens from the evidence, as illustrated in Figure 2. We jointly encode the claim and evidence by concatenating these two inputs to the model as this this would allow the self-attention layers within the transformer to better capture long-range dependencies (Vaswani et al., 2017).

We compare this to pointer generator network (See et al., 2017) which was adapted by Shah et al. (2020) to independently encode a single evidence sentence and the masked claim using LSTMs (Hochreiter and Schmidhuber, 1997). This pointer network uses that trained to copy masked tokens from the evidence and generate a correction. Rather than independently encode the inputs, the transformer model we use (outlined above) jointly encodes the claim and evidence to condition generation of the generation of the correction. We evaluate the quality of corrections trained using different maskers as this is likely to impact the effectiveness of the generated correction. To also evaluate the impact of conditioning on evidence, we consider an unconditioned language model, replicating 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

A factual error correction requires different training data depending on whether the system is trained with full-supervision or using the masker-corrector approach discussed in Section 5.3. In this paper, we make use of FEVER (Thorne et al., 2018a), a common fact verification dataset as the basis for experiments for both training strategies. FEVER contains 185k instances which are annotated with corresponding evidence from Wikipedia and a label as to whether the claim is SUPPORTED or REFUTED by it. This is the largest available resource for fact verification and can be used to train a masker-corrector system without modification.

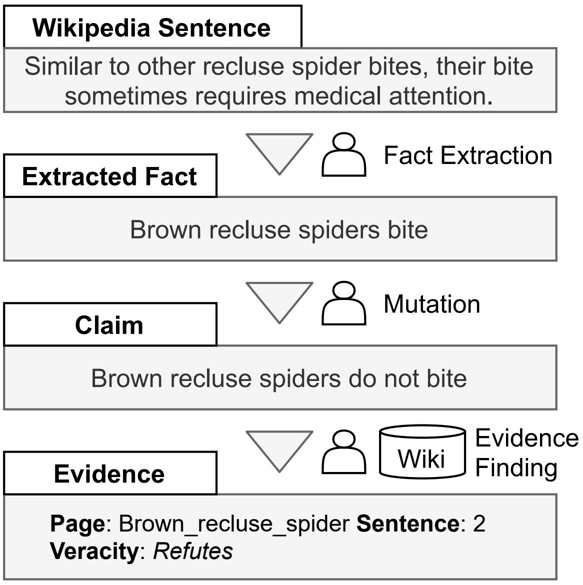
Wikipedia Sentence
Similar to other recluse spider bites, their bite sometimes requires medical attention.
Extracted Fact
Fact Extraction
Brown recluse spiders bite
Claim
Mutation
Brown recluse spiders do not bite
Evidence
Wiki
Evidence Finding
Page: Brown_recluse_spider Sentence: 2
Veracity: Refutes

Figure 3: Claims in FEVER were written by human annotators by mutating facts extracted from Wikipedia. Undoing the mutation provides a reference sentence to evaluate error correction systems.

To comprehensively evaluate the corrections generated by error correction systems, manual evaluation is required. However, this comes with a high cost and is not suitable for system debugging and development. To support automated evaluation, we make use of intermediate data from FEVER to generate reference sentences for corrections. The claims in FEVER were generated by annotators in a two-stage process. Annotators both extracted facts from Wikipedia and then performed meaning altering perturbations called *mutations* over these extracted facts, illustrated Figure 3. Each claim is independently labeled using retrieved evidence Our reference corrections are *undoing* the mutations introduced by the annotators, restoring the extracted facts.

The error correction dataset we release in this paper is constructed from adapting FEVER. For all claims, we release the extracted facts used along with which type of mutation the annotator used to generate the claims. These extracted facts serve as the reference sentences as targets for correction. While this reference sentence isn't needed for training a masker-corrector system, we release the annotations to both support automated system evaluation, and to train a baseline system with full supervision for comparison.

The class balance and size of the dataset is reported in Table 1. A portion of the claims in FEVER cannot be verified by evidence and are labelled as NOTENOUGHINFO. While these are used as negative examples for training fact verification classifiers, they will not be used for training the error correction systems in this paper as there is no evidence to make corrections. For completeness, we release these additional 21934 training, 1870 development and 2037 test instances that remained unused.

Label	Instance Count		
	Train	Dev	Test
Supports	37961	1477	1593
Refutes	20075	2091	2289
Total Training	58036	3568	4380

Table 1: Instance counts by class for training, development and test dataset partitions

7 Evaluation

Due to the difficulties in automated evaluation of generated language, the primary evaluation for our study will use human raters to label whether the generated outputs meet the task requirements. While, during development, it's convenient to use an automatic metric for speed and to reduce expense, the automated metrics that compute token overlap against a single reference sentence cannot capture the nuances required to assess the veracity of the generated corrections against evidence.

Our human evaluation is a double-blind discrete assessment of whether the corrections meet the requirements of fluency, supported by evidence, and error correction introduced in Section 3. Annotators are asked three questions which had a binary answer for whether the correction is fluent and supported by evidence, and a three way answer for whether the error is corrected. This additional neutral choice for the final question accounts for cases where an error is removed, but replaced with an alternative fact rather than a correction. Annotators were shown each question in sequence. Negative answers to a question automatically assigned negative answers to subsequent ones (describing that a disfluent sentence could not contain a fact supported by evidence or introduce a correction). In Section 9, we compare the correlation of automated metrics against this manual evaluation.

8 Implementation + Protocol

The T5 sequence-to-sequence models were fine-tuned using the T5-base pre-trained models released by HuggingFace (Wolf et al., 2020). Each model was trained for 4 epochs, optimizing the negative log likelihood for the corrected sentence (for masker+corrector variant) or the target correction (for the fully supervised ceiling). The claim and evidence were concatenated at input, and the page title for evidence was included in this string. We limited the input size to 256 tokens, truncating anything longer.

Evidence was retrieved using the Facebook implementation of DPR (Karpukhin et al., 2020), we used the Wikipedia dataset released with FEVER (Thorne et al., 2018a) and chunked this into passages of 50 tokens. We selected the top k passages through selecting the model with the highest SARI score. We had highest results when selecting $k = 2$ passages of evidence.

For maskers that queried a fact verification model, these models were trained using a variant of the FEVER dataset which sub-sampled SUPPORTED instances during training to ensure class balance and generated NOTENOUGHINFO instances by randomly sampling negative evidence for equal number of SUPPORTED and REFUTED claims using evidence retrieved from DPR. For the black-box masker, we use the LIME implementation from (Ribeiro et al., 2016) against a BERT classifier (Devlin et al., 2019) fine-tuned for 3 epochs, optimizing label accuracy. For the white-box masker, we use the ESIM model implementation released by Shah et al. (2020) with our updated dataset. To compare against the corrector released by Shah et al. (2020), we also use their implementation with our data retaining the authors' hyper-parameter choices and training regimen. For the masked LM and random baselines, we controlled the mask rate to 50% of the tokens to remain consistent with the model-based maskers.

9 Results

We characterize the task and dataset, comparing full supervision to the masker and corrector approach. We first report scores using automated evaluations before a final human evaluation in Section 9.6.

9.1 Fully supervised system

Firstly, we report the performance ceiling using a fully supervised sequence to sequence architecture and evaluate how noisy evidence retrieved from DPR has an impact on this. Adding retrieval using DPR has a small negative impact on the SARI scores, despite having a lower evidence recall.

Using the top 2 passages retrieved by DPR, at least one passage from the annotated evidence in the FEVER dataset was retrieved for 63% of instances. This, however, cannot be directly compared to retrieval scores for the FEVER task as we are using a more challenging subset of the dataset containing only mutated claims, rather than facts as they appear in Wikipedia. A further point for consideration is that, in FEVER, evidence annotation is incomplete and focused around the entity used to generate the claim. We did not account for this bias in our retrieval. Despite this, an average of 62% of tokens in the labeled evidence ($\sigma = 30\%$) were retrieved by DPR.

Automated scoring using SARI was more sensitive to the retrieved evidence whereas ROUGE2 and BLEU were not sensitive to this change. While SARI acts as an indicator for performance, human evaluation is required to assess to what extent the changes in this metric change the properties of the generated output with respect to the task requirements. Human evaluation is reported on in Section 9.6.

9.2 Choice of masker

We compare three strategies for masking out salient tokens from claims and evaluate these extrinsically by scoring the generated corrections against references. For the corrector, we use a T5 model with evidence retrieved from DPR in all cases. We use the masks generated by each of the techniques outlined in Section 5.3.

Both the black-box (LIME) and white-box (surrogate classifier from Shah et al. (2020)) methods require querying a fact veracity classifier to generate the masks and vary the use of evidence (gold vs retrieved with DPR) when generating the masks. Using retrieved evidence when generating the masks had different effects for the black-box and white-box maskers. 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). When using retrieved evidence to generate white-box explanations for masking (row 4 in Table 3), the system failed to generate explanations for approximately quarter of instances.

For corrections, the SARI Add scores are markedly lower than with full supervision. This is due to the different supervision signal's influence. Notably, that errors can be corrected without fully rewriting the claims or undoing mutations by simply deleting contradictory tokens or by substituting a single token in the claim. This will be discussed further in Section 9.6.

9.3 Training with a random masker

Generating large quantities of masked training data through querying a model, such as with the black-box model explanation techniques, can be costly. In contrast, random masks can be generated without querying a model. We evaluate the combined masker and corrector, using a corrector trained on random masks but with informed maskers (such as white-box and black-box model explanation) at test time. The results are reported in Table 4.

For all maskers, final SARI scores remained consistent with, or were improved, when using a random masker during training. We observed that in every case, the SARI Add score increased indicating that more novel information is incorporated into the correction. The tokens masked with the random masker were selected uniformly at random, whereas the distribution of tokens using the other maskers is informed by the information content in the claim and evidence. This indicates that using training data with a wider coverage of token masking positions may have a positive impact on the downstream correction task.

9.4 Alternative correctors

In Section 2, we identified that knowledge captured in language models can be used to make corrections in claims (Petroni et al., 2019) and that pointer networks can be used to generate corrections using an independently encoded claim and evidence (Shah et al., 2020). For completeness, we include the outcomes of these systems in Table 6 and Table 5 respectively.

Like the T5 model, the corrections generated by a dual encoder pointer network using the system released by Shah et al. (2020) did not introduce many new, reflected in the low SARI Add score. However, the final SARI score is lower than the

System	SARI Score				ROUGE2	BLEU
	Keep	Delete	Add	Final		
T5 supervised + Gold Evidence	.7659	.8736	.5415	.7270	.7752	.7047
T5 supervised + Retrieved Evidence (DPR)	.7567	.8495	.5060	.7041	.7740	.7044

Table 2: Fully supervised T5 sequence to sequence model with gold evidence (collected by annotators) and retrieved evidence (using DPR) to correct claims.

Masker	Success Rate	SARI Score			
		Keep	Delete	Add	Final
Black-box (gold)	99.8	.5751	.5473	.0633	.3952
White-box (gold)	100	.6112	.5192	.0831	.4045
Black-box (DPR)	99.7	.5294	.5305	.0589	.3730
White-box (DPR)	73.9	.4500	.4980	.0375	.3285
Heuristic (DPR)	91.7	.5846	.5187	.0278	.3770
Random	100	.5695	.4498	.0624	.3606

Table 3: Extrinsic evaluation of maskers using a corrector built using a T5 seq2seq model with evidence retrieved from DPR.

Masker	Success Rate	SARI Score			
		Keep	Delete	Add	Final
Black-box (gold)	99.8	.5640	.5861	.0751	.4084
White-box (gold)	100	.5970	.5332	.0765	.4022
Black-box (DPR)	99.7	.5539	.5549	.0793	.3960
White-box (DPR)	73.9	.5272	.4918	.0793	.3661
Masked LM	100	.4841	.5053	.0560	.3485
Heuristic (DPR)	91.7	.6065	.6800	.1130	.4665

Table 4: Using random masks at training resulted in higher SARI scores when testing with different maskers

System	SARI Score			
	Keep	Delete	Add	Final
Dual Enc Pointer - gold	.4521	.5689	.0390	.3533
Dual Enc Pointer - DPR	.3451	.4807	.0167	.2808

Table 5: Using a dual encoder pointer network with white-box model explanations.

systems using the T5 corrector, as fewer tokens from the claim were kept, indicated by the lower SARI Keep score. While the protocol for white-box masking was useful when combined with the T5 transformer corrector, independently encoding the claim and evidence did not result in satisfactory corrections, especially when combining retrieved evidence in the corrector (Table 5, row 2). Hyper-parameter tuning did not yield improvements.

Using a pre-trained language model, without fine-tuning to correct claims also resulted in low SARI scores, regardless of choice of masker. This model has two limitations: firstly it does not condition the correction on the original claim or evidence, and secondly, multiple tokens are masked and the baseline model independently decodes the missing tokens sometimes resulting in ungrammatical outputs.

System	SARI Score			
	Keep	Delete	Add	Final
Masked LM	.3474	.4793	.0158	.2808
White-box (gold)	.4191	.5408	.0154	.3251
White-box (DPR)	.3586	.3065	.0045	.2232
Black-box (gold)	.4516	.5919	.0260	.3565
Black-box (DPR)	.3632	.5080	.0163	.2958

Table 6: Using a dual encoder pointer network with white-box model explanations, did not yield satisfactory corrections.

9.5 Informing mutations

We evaluate whether informing the model with the mutation type used by the annotator to generate the claim increases the quality of corrections. We experiment with two methods for informing the model: one is to include the mutation type in the source, prior to correction; the second is to train the model to predict the mutation type as part of the target correction. We experiment with this using a T5 model with full supervision for the correction and evidence for each claim.

System	SARI Score			
	Keep	Delete	Add	Final
Uninformed	.7659	.8736	.5415	.7270
Mutation (source)	.7859	.8726	.5774	.7453
Mutation (target)	.7471	.8367	.5423	.7087

Table 7: Fully supervised T5 sequence to sequence model with gold evidence (collected by annotators) and retrieved evidence (using DPR) to correct claims.

The results, listed in Table 7, indicate that adding the mutation type information generated corrections with a moderately higher SARI score with

Masker	Corrector	Success Rate (%)	Aggregated Score (%)		
			Fluency	Supported	Corrected
Fully supervised T5 (Gold)		100	95.4	90.1	85.1
Fully supervised T5 (DPR)		100	99.3	88.2	81.3
Black-box (DPR)	T5 random (DPR)	99.7	94.5	45.6	30.6
Black-box (DPR)	T5 black-box (DPR)	99.7	92.4	35.8	20.6
White-box (DPR)	T5 white-box (DPR)	73.9	98.0	22.1	18.1
White-box (Gold)	Dual Enc Pointer (Gold)	100	66.2	21.8	6.6
Masked LM	Greedy BERT LM decode	100	32.9	7.4	4.0

Table 8: Aggregated scores from human evaluation considering fluency, whether generated instances were supported by evidence and errors corrected.

increases to the Keep and Add components. While some mutation types provide information about which entities to substitute (for example, the generalization of pianist is could be a musician), there is still ambiguity about which entities may have been substituted by the annotators when generating the mutations for extracted facts.

9.6 Human Evaluation

We finally report a double-blind human evaluation of the outputs generated by the systems against the three requirements listed in Section 3: fluency, supported by evidence, and error correction. We sample 1000 instances and report the averaged scores from the 3 evaluation questions in Table 8.

The fully supervised models had the highest rate of satisfactory corrections (requirement 3). The systems generated outputs for all inputs and most corrections were fluent (requirement 1), supported by evidence (requirement 2), and addressed the error in the source claim (requirement 3). Using retrieved evidence from DPR had a negative impact on the number of corrections supported by evidence.

For systems built using the masker-corrector architecture, the correction rate was lower than the fully supervised models. Using random masks when training the corrector resulted in a higher number of corrected claims than using a specific masker for training. For the DPR black-box model, this resulted in an improvement of the correction rate by 10%.

Using the alternative maskers, such as the dual encoder pointer network and greedy LM decoding, did not result satisfactory corrections. The Masked LM outputs were often disfluent, and the dual encoder pointer network tended to copy information from the original claim without making corrections.

In Table 9, we report the correlation between the human evaluation and automated metrics. Most automated evaluation metrics exhibited a high correlation against whether the correction is supported by evidence and correcting the error in the claim. However, BLEU did not correlate as strongly with either requirement. ROUGE and all SARI components correlated with whether the correction was supported by evidence and correcting any errors. However, SARI is the only scoring metric to consider the source claim for error correction. Correlation between these metrics and the fluency was lower in all cases.

Even though the SARI Add scores were low for all systems (with the exception of full supervision), they still correlated well with the requirement 2 and 3 in the manual evaluation. In this manual evaluation, it was observed that claims can be corrected without the need to add new tokens. For example “Donald Trump was president of France” can be corrected by deleting “of France” without copying new tokens from the evidence. This is in contrast to the reference correction which may have used “of the USA” to undo a entity substitution introduced by the annotator when generating the original mutated claim.

10 Conclusions

Going beyond simply identifying errors, factual error correction presents a number of interesting research challenges for information retrieval, misinformation detection and abstractive summarization communities alike. In this paper, we demonstrated that the task can be performed with both full supervision and distant supervision with reasonable suc-

Metric	Correlation (Pearson r)		
	Fluency	Suport.	Correct.
SARI Final	.57	.98	.98
SARI Keep	.73	.97	.96
SARI Delete	.45	.96	.94
SARI Add	.46	.95	.98
ROUGE2	.70	.97	.96
ROUGE1	.71	.94	.92
BLEU2	.29	.78	.79
BLEU1	-.02	.56	.57

Table 9: Correlation between automated scoring metrics and human evaluation scores for the corrected claims.

cess using existing technologies. However, there are a number of outstanding challenges that must be addressed including how evidence is incorporated into the correction and how the generated corrections are scored. 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 written documents. Furthermore, the use of the extracted facts as references for scoring assesses whether systems can undo mutations introduced by the annotators rather than correct errors, warranting further manual evaluation. While these automated metrics correlated well with human judgements (with the exception of fluency), future work should consider how sensitivity in the scoring can be used to better discriminate the quality of corrections generated by the systems.

Acknowledgements

This research was supported by donation of resources from Google Cloud. Thank you to Tal Schuster for his helpful comments and feedback.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. Generating Fact Checking Explanations. In *ACL 2020 - 58th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*.

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 EMNLP-IJCNLP*, 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.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, page 1657–1668, Vancouver, Canada.

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.

Hoa Trang Dang. 2005. Overview of DUC 2005. In *Proceedings of the document understanding conference*, page 1Ą12.

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)*, Minneapolis, Minnesota.

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 conference on International Language Resources and Evaluation ({LREC}{'}10)*, Valletta, Malta. European Languages 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. 2019. The curious case of neural text degeneration.

Sarthak Jain and Byron C Wallace. 2019. Attention is not not Explanation. In *Proceedings of NAACL-HLT*, Minneapolis, Minnesota. ACL.

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

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. *Proceedings of the National Conference on Artificial Intelligence*, 1:779–784.

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?

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.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding Neural Networks through Representation Erasure.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. (1990).

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. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, (July):1–14.

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.

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, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language Models as Knowledge Bases? In *Proceedings of EMNLP-IJCNLP*, 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 Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. 39(2011):117831.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. *Proc. of 32nd Conference on Artificial Intelligence (AAAI)*, pages 1527–1535.

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.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *IJCAI International Joint Conference on Artificial Intelligence*, 0:2662–2670.

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

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

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval.

Dominik Stammbach and Elliott Ash. 2020. e-FEVER: Explanations and Summaries for Automated Fact Checking. *Truth and Trust Online*.

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. 2018a. 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.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019a. Evaluating adversarial attacks against multiple fact verification systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019b. Generating Token-Level Explanations for Natural Language Inference. 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 963–969, Minneapolis, Minnesota. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The Fact Extraction and VERification (FEVER) Shared Task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. 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.

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.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. *Naacl-2016*, pages 380–386.

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.