Automatic Fact-Guided Sentence Modification

Darsh J Shah,[*] Tal Schuster,[*] Regina Barzilay
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
{darsh, tals, regina}@csail.mit.edu

Abstract

Online encyclopediae like Wikipedia contain large amounts of text that need frequent corrections and updates. The new information may contradict existing content in encyclopediae. In this paper, we focus on rewriting such dynamically changing articles. This is a challenging constrained generation task, as the output must be consistent with the new information and fit into the rest of the existing document. To this end, we propose a two-step solution: (1) We identify and remove the contradicting components in a target text for a given claim, using a neutralizing stance model; (2) We expand the remaining text to be consistent with the given claim, using a novel two-encoder sequence-to-sequence model with copy attention. Applied to a Wikipedia fact update dataset, our method successfully generates updated sentences for new claims, achieving the highest SARI score. Furthermore, we demonstrate that generating synthetic data through such rewritten sentences can successfully augment the FEVER fact-checking training dataset, leading to a relative error reduction of 13%.[1]

1 Introduction

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 Specifically, given a claim and an outdated sentence from an article, we rewrite the sentence to be consistent with the given claim while preserving non-contradicting content.

Consider the Wikipedia update scenario depicted in Figure 1. The claim, informing that *23 of 43* minority stakeholdings are significant, contradicts the old information in the Wikipedia sentence, requiring modification. Directly learning a model for this task would demand supervision, i.e. demonstrated updates with the corresponding claims. For Wikipedia, however, the underlying claims which drive the changes are not easily accessible. Therefore, we need to utilize other available sources of supervision.

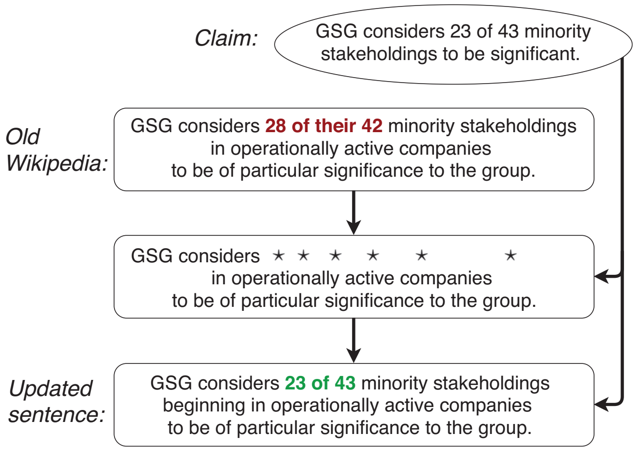
Claim:
GSG considers 23 of 43 minority stakeholdings to be significant.
Old Wikipedia:
GSG considers 28 of their 42 minority stakeholdings in operationally active companies to be of particular significance to the group.
GSG considers ⋆ ⋆ ⋆ ⋆ ⋆ ⋆ in operationally active companies to be of particular significance to the group.
Updated sentence:
GSG considers 23 of 43 minority stakeholdings beginning in operationally active companies to be of particular significance to the group.

Figure 1: Our fact-guided update pipeline. Given a claim which refutes incorrect information, a masker is applied to remove the contradicting parts from the original text while preserving the rest of the context. Then, the residual neutral text and claim are fused to create an updated text that is consistent with the claim.

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

For the first step, we utilize a neutrality stance classifier as indirect supervision to identify the polarizing spans in the target sentence. We consider a sentence span as polarizing if its absence increases the neutrality of the claim-sentence pair. To identify and mask such sentence spans,

[*] Order decided by a coin toss.

[1] Code: (1) https://github.com/TalSchuster/TokenMasker (2) https://github.com/darsh10/split_encoder_pointer_summarizer

we introduce an interpretability-inspired (Lei, Barzilay, and Jaakkola 2016) neural architecture to effectively explore the space of possible spans. We formulate our objective in a way that the masking is minimal, thus preserving the context of the sentence.

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

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

2 Related Work

Text Rewriting There have been several recent advancements in the field of text rewriting, including style transfer (Shen et al. 2017; Zhang et al. 2018; Chen et al. 2018) and sentence fusion (Barzilay and McKeown 2005; Narayan et al. 2017; Geva et al. 2019). Unlike previous approaches, our sentence modification task addresses potential contradictions between two sources of information.

Our work is fairly related to the approach of (Li et al. 2018), which separates the task of sentiment transfer into deleting strong markers of sentiment in a sentence and retrieving markers of the target label to generate a sentence with the opposite sentiment. In contrast to such work, where the requisite modification is along a fixed aspect (e.g. sentiment), in our setting, an arbitrary input sentence (the claim) dictates the space of desired modifications. Therefore, in order to succeed at our task, a system should understand the varying degree of polarization in the spans of the outdated sentence against the claim before modifying the sentence to be consistent with the claim.

Wikipedia Edits Wikipedia edit history has been analyzed for insights into the kinds of modifications made (Daxenberger and Gurevych 2013; Yang et al. 2017; Faruqui et al. 2018). The edit history has also been used for text generation tasks such as sentence compression and simplification (Yatskar et al. 2010), paraphrasing (Max and Wisniewski 2010) and writing assistance (Cahill et al. 2013). In this work, we are interested in the novel task of automating the editing process with the guidance of a textual claim.

Fact Verification Datasets The growing interest in automatic fake news detection led to the development of several fact verification datasets (Vlachos and Riedel 2014; Wang 2017; Rashkin et al. 2017; Thorne et al. 2018). FEVER, the largest fact-checking dataset, contains 185K human written fake and real claims, generated by crowd-workers, in context of sentences from Wikipedia articles. This dataset contains biases that allow a model to identify many of the false claims without any evidence (Schuster et al. 2019). This bias affects the generalization capabilities of models trained on such data. In this work, we show that our automatic modification method can be used to augment a fact-checking dataset and to improve the inference of models trained on it.

Data Augmentation Methods for data augmentation are commonly used in computer vision (Perez and Wang 2017). There have been recent successes in NLP where augmentation techniques such as paraphrasing and word replacement were applied to text classification (Kobayashi 2018; Wu et al. 2018). Adversarial examples in NLI with syntactic modifications can also be considered as methods of data augmentation (Iyyer et al. 2018; Zhang, Baldridge, and He 2019). In this work, we create constrained modifications, based on a reference claim, to augment data for our task at hand. Our additions are specifically aimed towards reducing the bias in the training data, by having a false claim appear in both "Agrees" and "Disagrees" classes.

3 Model

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

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

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

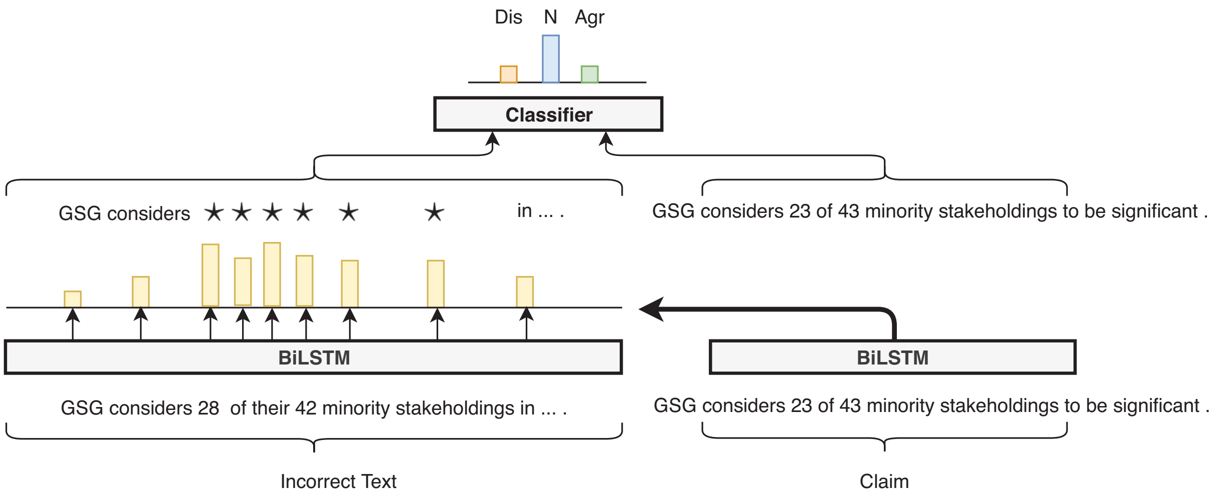
Dis
N
Agr
Classifier
GSG considers ★ ★ ★ ★ ★ ★ in ... .
GSG considers 23 of 43 minority stakeholdings to be significant .
BiLSTM
BiLSTM
GSG considers 28 of their 42 minority stakeholdings in ... .
GSG considers 23 of 43 minority stakeholdings to be significant .
Incorrect Text
Claim

Figure 2: Illustrating the flow of the masker module.

3.1 Masker: Eliminate Polarizing Spans

In this section we describe the module to identify the polarizing spans within a Wikipedia sentence. Masking these spans ensures that the residual sentence-claim pairs attain a neutral relation. Here, neutrality is determined by a classifier trained on claim and Wikipedia sentence pairs as described below. Using this classifier, the masking module is trained to identify the polarizing spans by maximizing the neutrality of the residual-sentence and claim pairs. In order to preserve the context of the original sentence, we include optimization constraints to ensure minimal deletions. This approach is similar to neural rationale-based models (Lei, Barzilay, and Jaakkola 2016), where a module tries to identify the spans of the input that justify the model's prediction.

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

$$S_i^{\emptyset} = \begin{cases} x_i, & \text{if } m_i = 0 \\ \star, & \text{otherwise} \end{cases} \tag{1}$$

where $\star$ is a special token.[2] The details of the masker architecture are stated below and depicted in Figure 2.

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

$$z_i = e_i + \sum_{j=1}^{n} a_{i,j} \cdot c_j, \tag{2}$$

where c_j are the encoded representations of the claim and $a_{i,j}$ are the parameterized bilinear attention (Kim, Jun, and Zhang 2018) weights computed by:

$$a_{i,j} = \text{softmax}_j(\text{atten}(e_i, c_j)), \tag{3}$$

$$\text{atten}(e_i, c_j) = e_i W c_j^T + b. \tag{4}$$

Finally, the aggregated representations are used as input to a sequence encoder $g(\cdot; \boldsymbol{w}_g)$.

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

$$p(m_i = 1) = \sigma(g(z; \boldsymbol{w}_g)_i). \tag{5}$$

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

$$S^{\emptyset} = S \circ (1 - m), \tag{6}$$

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

Training A pretrained fact-checking neutrality classifier's prediction $\boldsymbol{rel}(S, C)$ is used to guide the training of the masker. In order to encourage maximal retention of the context, we utilize a regularization term to minimize the fraction of the masked words. The joint objective is to minimize:

$$\mathcal{L}(S, C, m) = -\log\left(p(\boldsymbol{rel}(S^{\emptyset}, C) = \boldsymbol{N})\right) + \frac{\lambda}{l}\sum_{i=1}^{l} m_i. \tag{7}$$

Fact-checking Neutrality Classifier Our fact-checking classifier is pretrained on agreeing and disagreeing (S, C) pairs from $\mathcal{D}$, in addition to neutral examples constructed through negative sampling. For each claim we construct a

[2] The special token is treated as an out-of-vocabulary token for the following models.

neutral pair by sampling a random sentence from the same paragraph of the polarizing sentence, making it contextually close to the claim, but unlikely to polarize it. We pretrain the classifier on these examples and fix its parameters during the training of the masker.

Optional Syntactic Regularization Currently the model is trained with distant supervision, so, we pre-compute a valid neutrality mask as additional signal, when possible. To this end, we parse the original sentences using a constituency parser and iterate over continuous syntactic phrases by increasing length. For each sentence, the shortest successful neutrality mask (if any) is selected as a target mask.[3] In the event of successfully finding such a mask, the masking module is regularized to emulate the target mask by adding the following term to Eq. 7:

$$\frac{1}{l}||m - m'||^2, \tag{8}$$

where m' is the target mask.

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

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

In this section we describe our method to generate an output which agrees with the claim. If the earlier masking step is done perfectly, the merging boils down to a simple fusion task. However, in certain cases, especially ones with a strong contradiction, our minimal deletion constraint might leave us with some residual contradictions in $S^{\emptyset}$. Thus, we develop a model which can control the amount of information to consider from either input.

We extend the pointer-generator model of (See, Liu, and Manning 2017) to enable multiple encoders. While sequence-to-sequence models support the encoding of multiple sentences by simply concatenating them, our use of a per input encoder allows the decoder to better control the use of each source. This is especially of interest to our task, where the context of the claim must be translated to the output while ignoring contradicting spans from the outdated Wikipedia sentence.

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

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

$$h^t = \mathit{RNN}([r^t, emb(y^{t-1})], h^{t-1}). \tag{9}$$

As the decoder should decide at each time step which encoder to attend more, we introduce an encoder weight α. The shared encoder context representation r^t is based on their individual representations r_1^t and r_2^t:

$$\begin{aligned} \alpha &= \sigma(u_{enc}^T[r_1^t, r_2^t]), \\ r^t &= \alpha \cdot r_1^t + (1-\alpha) r_2^t. \end{aligned} \tag{10}$$

The context representation r_i^t ($i \in \{1, 2\}$) is the attention score over the encoder representation r_i for a particular decoder state h^{t-1}:

$$\begin{aligned} z_j^t &= u^T \tanh(r_{i,j} + h^{t-1}), \\ a_i^t &= \text{softmax}(z^t), \\ r_i^t &= \sum_j a_{i,j}^t r_{i,j}. \end{aligned} \tag{11}$$

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

$$p_{gen} = \sigma(v_x^T x^t + v_h^T h^t + v_r^T r^t). \tag{12}$$

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

$$p_{enc1} = \sigma(u_x^T x^t + u_h^T h^t + u_r^T r^t). \tag{13}$$

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

$$P_{vocab} = \text{softmax}(V^T[h^t, r^t]). \tag{14}$$

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

$$\begin{aligned} P(w) = &\, p_{gen} P_{vocab}(w) + \\ &(1-p_{gen})(p_{enc1}) \sum_{j:w_j=w} a_{1,j}^t + \\ &(1-p_{gen})(1-p_{enc1}) \sum_{j:w_j=w} a_{2,j}^t, \\ y^t = &\, \text{argmax}_w \, P(w). \end{aligned} \tag{15}$$

where a^t are the input sequence attention scores from Eq. 11.

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

[3] If there are several successful masks of the same length, we use the one with the highest neutrality score.

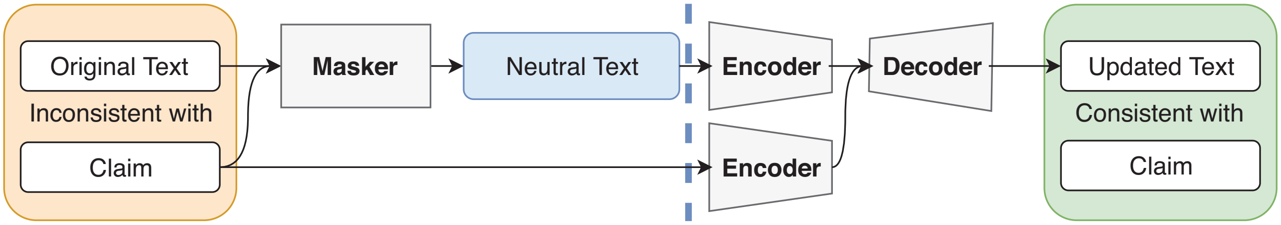

Figure 3: A summary of our pipeline. Given a sentence that is inconsistent with a claim, a masker is applied to mask out the contradicting parts from the original text while preserving the rest of the content. Then, the residual neutral text and claim are fused to create an updated text that is consistent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

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

4 Experimental Setup

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

4.1 Datasets

Training Data from FEVER We use FEVER (Thorne et al. 2018), the largest available Wikipedia based fact-checking dataset to train our models for both of our tasks. This dataset contains claim-evidence pairs where the claim is a short factual sentence and the evidence is a relevant sentence retrieved from Wikipedia. We use these pairs as our claim-setnence samples and use the "refutes", "not enough information", "supports" labels of that dataset as our $\boldsymbol{Dis}$, $\boldsymbol{N}$, $\boldsymbol{Agr}$ relations, respectively.

Evaluation Data for Automatic Fact Updates We evaluate the automatic fact updates task on an evaluation set based on part of the symmetric dataset from (Schuster et al. 2019) and the fact-based cases from a Wikipedia updates dataset (Yang et al. 2017). For the symmetric dataset, we use the modified Wikipedia sentences with their guiding claims to generate the true Wikipedia sentence. For the cases from the updates dataset, we have human annotators write a guiding claim for each update and use it, together with the outdated sentence, to generate the updated Wikipedia sentence. Overall we have a total of 201 tuples of fact update claims, outdated sentences and updated sentences.

Evaluation Data for Augmentation To measure the proficiency of our generated outputs for data augmentation, we use the unbiased FEVER-based evaluation set of (Schuster et al. 2019). As shown by (Schuster et al. 2019), the claims in the FEVER dataset contain give-away phrases that can make FEVER-trained models overly rely on them, resulting in decreased performance when evaluated on unbiased datasets.

The classifiers trained on our augmented dataset are evaluated on the unbiased symmetric dataset of (Schuster et al. 2019). This dataset (version 0.2) contains 531 claim-evidence pairs for validation and 534 claim-evidence pairs for testing.

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

4.2 Implementation Details

Masker We implemented the masker using the AllenNLP framework (Gardner et al. 2018). For a neutrality classifier, we train an ESIM model (Chen et al. 2017) to classify a relation of $\boldsymbol{Agr}$, $\boldsymbol{Dis}$ or $\boldsymbol{N}$. To train this classifier, we use the $\boldsymbol{Agr}$ and $\boldsymbol{Dis}$ pairs from the FEVER dataset and for each claim we add a neutral sentence which is sampled from the sentences in the same document as the polarizing one. The classifier and masker are trained with GloVe (Pennington, Socher, and Manning 2014) word embeddings. We use BiLSTM (Sak, Senior, and Beaufays 2014) encoders with hidden dimensions of 100 and share the parameters of the claim and original sentence encoders. The model is trained for up to 100 epochs with a patience value of 10, where the stopping condition is defined as the highest delta between accuracy and deletion size on the development set (Δ in Table 3).

For syntactic guidance, we use the constituency parser of

	Automatic Evaluation				Human's Scores	
model	SARI	Keep	Add	Del	Grammar	Agreement
***Fact updates*:**						
Split-no-Copy	15.1	36.9	1.9	49.5	-	-
Paraphrase	15.9	18.7	4.2	50.7	3.75	3.65
Claim Ext.	12.9	22.6	1.9	50.4	1.75	2.65
M. Concat	26.5	**61.7**	6.7	44.9	3.28	2.75
Ours	**31.5**	45.4	**13.2**	**52.1**	**3.85**	**4.00**
Human					4.80	4.70
***Data augmentation*:**						
Paraphrase	18.2	12.5	10.6	45.7	4.12	3.92
Claim Ext.	12.2	9.8	4.0	46.4	1.58	2.84
M. Concat	22.1	**71.6**	6.8	22.3	**4.45**	2.05
Ours	**34.4**	33.0	**26.0**	**47.5**	4.14	**3.98**
Human					4.69	4.15

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

(Stern, Andreas, and Klein 2017) and consider continuous spans of length 2 to 10 as masking candidates (without combinations). By doing so, we obtain valid neutrality masks for 38% of the $\boldsymbol{Agr}$ and $\boldsymbol{Dis}$ pairs from the FEVER training dataset. These masks are used for Eq. 8.

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

BERT Fact-Checking Classifier We use a BERT (Devlin et al. 2018) classifier, which takes in as input a (claim-evidence) pair separated by a special token, to predict out of 3 labels ($\boldsymbol{Agr}$, $\boldsymbol{Dis}$ or $\boldsymbol{N}$). The model is fine-tuned for 3 epochs, which is sufficient to perform well on the task.

Evidence Regeneration Since we are interested in using the generated supporting pairs for data augmentation, we add machine generated cases to the $\boldsymbol{Agr}$ set of the dataset. Adding machine generated sentences to only one of the labels in the data can be ineffective. Therefore, we balance this by regenerating paraphrased refuting evidence for the false claims. This is then added along with all models' outputs for a balanced augmentation.

4.3 Baselines

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

- **Copy Claim** The sentence of the claim is copied and used as the updated sentence for itself (used only for data augmentation).
- **Paraphrase** The claim is paraphrased using the back-translation method of (Wieting and Gimpel 2018)[4], and the output is used as the updated sentence.
- **Claim Extension [Claim Ext.]** A pointer-generator network is trained to generate the updated sentence from an input claim alone. The model is trained on FEVER's agreeing pairs and applied on the to-be-updated claims during inference.
- **Masked Concatenation [M. Concat]** Instead of our Two-Encoder Generator, we use a pointer-generator network. The residual sentence (output from the masker module) and the claim are concatenated and used as input.
- **Split Encoder without Copy [Split-no-Copy]** Our Two-Encoder Generator, without the copy mechanism. The original text and contradicting claim are passed through each of the encoders.

5 Results

We report the performance of the model outputs for automatic fact-updates by comparing them to the corresponding correct wikipedia sentences. We also have crowd workers score the outputs on grammar and for agreeing with the claim. Additionally, we report the results on a fact-checking classifier using model outputs from the FEVER training set as data augmentation.

Fact Updates Following recent text simplification work, we use the SARI (Xu et al. 2016) method. The SARI method takes 3 inputs: (i) original sentence, (ii) human written updated sentence and (iii) model output. It measures the similarity of the machine generated and human reference sen-

[4] https://github.com/vsuthichai/paraphraser

MODEL	DEV	TEST	+TURK
No Augmentation	62.7	66.1	77.0
Paraphrase	60.8	64.6	77.4
Copy Claim	62.1	63.6	77.4
Claim Ext.	62.5	65.0	76.8
M. Concat	60.1	63.7	78.5
Ours	**63.8**	**67.8**	**80.0**

Table 2: Classifiers' accuracy on the symmetric DEV and TEST splits. The right column (+TURK) shows the accuracy on the TEST set extended to include the 500 responses of turkers for the simulated process and the refuted pairs that they originated from. The BERT classifiers were trained on the FEVER training dataset augmented by outputs of the different methods.

tences based on the deletions, additions and kept n-grams[5] with respect to the original sentence.[6] For human evaluation of the model's outputs, 20% of the evaluation dataset was used. Crowd-workers were provided with the model outputs and the corresponding supposably consistent claims. They were instructed to score the model outputs from 1 to 5 (1 being the poorest and 5 the highest), on grammaticality and agreement with the claim.

Table 1 reports the automatic and human evaluation results. Our model gets the highest SARI score, showing that it is the closest to humans in modifying the text for the corresponding tasks. Humans also score our outputs the highest for consistency with the claim, an essential criterion of our task. In addition, the outputs are more grammaticality sound compared to those from other methods.

Examining the gold answers, we notice that many of them include very minimal and local modifications, keeping much of the original sentence. The M. Concat model keeps most of the original sentence as is, even at the cost of being inconsistent with the claim. This corresponds to a high KEEP score but a lower SARI score overall, and a low human score on supporting the claim. Claim Ext. and Paraphrase do not maintain the structure of the original sentence, and perform poorly on KEEP, leading to a low SARI score. The Split-no-Copy model has the same low ADD score as Claim Ext. since instead of copying the accurate information from the claim, it generates other tokens.

Data Augmentation For 41850 $\boldsymbol{Dis}$ pairs in the FEVER training data, our method generates synthetic evidence sentences leading to 41850 $\boldsymbol{Agr}$ pairs. We train the BERT fact-checking classifier with this augmented data and report the performance on the symmetric dataset in Table 2. In addition, we repeat the human evaluation process on the generated augmentation pairs and report it in Table 1.

λ	ACC	SIZE	Δ	PREC	REC	F1
.5	5.1	0.0	5	0.0	0.0	0.0
.4	80.0	26.3	**54**	27.2	75.1	**39.9**
.3	77.0	27.5	50	25.9	71.6	38.0
.2	81.6	31.1	51	23.1	74.8	35.3

Table 3: Results of different values of λ for the masker with syntactic regularization. The left three columns describe the accuracy and average mask size (% of the sentence) over the FEVER development set with the masked evidence and a neutral target label. Δ is ACC $-$ SIZE. The right three columns contain the precision, recall and F1 of the masks that we have human annotations for. For results without syntactic regularization see the appendix.

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

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

The results for different values of the regularization coefficient are reported in Table 3. Increasing the regularization coefficient helps to minimize the mask size and to improve the precision while maintaining the classifier accuracy and the mask recall. However, setting λ too large, can collapse the solution to no masking at all. The generation experiments use the outputs of the $\lambda = 0.4$ model.

6 Conclusion

In this paper, we introduce the task of automatic fact-guided sentence modification. Given a claim and an old sentence, we learn to rewrite it to produce the updated sentence. Our method overcomes the challenges of this conditional generation task by breaking it into two steps. First, we identify the polarizing components in the original sentence and mask them. Then, using the residual sentence and the claim, we generate a new sentence which is consistent with the claim. Applied to a Wikipedia fact update evaluation set, our method successfully generates correct Wikipedia sentences using the guiding claims. Our method can also be used for data augmentation, to alleviate the bias in fact verification

[5] We use the default up to 4-grams setting.

[6] Following (Geva et al. 2019) we use the F1 measure for all three sets, including deletions. The final SARI score is the geometric mean of the ADD, DEL and KEEP score.

datasets without any external data, reducing the relative error by 13%.

7 Acknowledgments

We thank the anonymous reviewers and the MIT NLP group for their helpful discussion and comments. This work is supported by DSO grant DSOCL18002.

References

Barzilay, R., and McKeown, K. R. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics* 31(3):297–328.

Cahill, A.; Madnani, N.; Tetreault, J.; and Napolitano, D. 2013. Robust systems for preposition error correction using Wikipedia revisions. *NAACL HLT* 507–517.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced lstm for natural language inference. *ACL* 1657–1668.

Chen, W.-F.; Wachsmuth, H.; Al Khatib, K.; and Stein, B. 2018. Learning to flip the bias of news headlines. *ICNLG* 79–88.

Daxenberger, J., and Gurevych, I. 2013. Automatically classifying edit categories in Wikipedia revisions. *EMNLP* 578–589.

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

Faruqui, M.; Pavlick, E.; Tenney, I.; and Das, D. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 305–315. Brussels, Belgium: Association for Computational Linguistics.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. AllenNLP: A deep semantic natural language processing platform. *Workshop for NLP-OSS* 1–6.

Geva, M.; Malmi, E.; Szpektor, I.; and Berant, J. 2019. DiscoFuse: A large-scale dataset for discourse-based sentence fusion. *NAACL HLT* 3443–3455.

Iyyer, M.; Wieting, J.; Gimpel, K.; and Zettlemoyer, L. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *NAACL HLT* 1875–1885.

Kim, J.-H.; Jun, J.; and Zhang, B.-T. 2018. Bilinear attention networks. *NIPS* 1564–1574.

Kobayashi, S. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *NAACL HLT* 452–457.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing neural predictions. *EMNLP* 107–117.

Li, J.; Jia, R.; He, H.; and Liang, P. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. *NAACL HLT* 1865–1874.

Max, A., and Wisniewski, G. 2010. Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Narayan, S.; Gardent, C.; Cohen, S. B.; and Shimorina, A. 2017. Split and rephrase. *EMNLP* 606–616.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. *EMNLP* 1532–1543.

Perez, L., and Wang, J. 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Rashkin, H.; Choi, E.; Jang, J. Y.; Volkova, S.; and Choi, Y. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. *EMNLP* 2931–2937.

Sak, H.; Senior, A.; and Beaufays, F. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *ISCA*.

Schuster, T.; Shah, D. J.; Yeo, Y. J. S.; Filizzola, D.; Santus, E.; and Barzilay, R. 2019. Towards debiasing fact verification models. *EMNLP-IJCNLP*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get to the point: Summarization with pointer-generator networks. *ACL* 1073–1083.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. *NIPS* 6830–6841.

Stern, M.; Andreas, J.; and Klein, D. 2017. A minimal span-based neural constituency parser. *ACL* 818–827.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a large-scale dataset for fact extraction and VERification. *NAACL* 809–819.

Vlachos, A., and Riedel, S. 2014. Fact checking: Task definition and dataset construction. *ACL Workshop on Language Technologies and Computational Social Science* 18–22.

Wang, W. Y. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *ACL* 422–426.

Wieting, J., and Gimpel, K. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *ACL* 451–462.

Wu, X.; Lv, S.; Zang, L.; Han, J.; and Hu, S. 2018. Conditional bert contextual augmentation. *arXiv preprint arXiv:1812.06705*.

Xu, W.; Napoles, C.; Pavlick, E.; Chen, Q.; and Callison-Burch, C. 2016. Optimizing statistical machine translation for text simplification. *TACL* 4:401–415.

Yang, D.; Halfaker, A.; Kraut, R.; and Hovy, E. 2017. Identifying semantic edit intentions from revisions in Wikipedia. *EMNLP* 2000–2010.

Yatskar, M.; Pang, B.; Danescu-Niculescu-Mizil, C.; and Lee, L. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. *NAACL* 365–368.

Zhang, Y.; Baldridge, J.; and He, L. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

Zhang, Z.; Ren, S.; Liu, S.; Wang, J.; Chen, P.; Li, M.; Zhou, M.; and Chen, E. 2018. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*.