Fixed That for You: Generating Contrastive Claims with Semantic Edits

Christopher Hidey
Department of Computer Science
Columbia University
New York, NY 10027
chidey@cs.columbia.edu

Kathleen McKeown
Department of Computer Science
Columbia University
New York, NY 10027
kathy@cs.columbia.edu

Abstract

Understanding contrastive opinions is a key component of argument generation. Central to an argument is the claim, a statement that is in dispute. Generating a counter-argument then requires generating a response in contrast to the main claim of the original argument. To generate contrastive claims, we create a corpus of Reddit comment pairs self-labeled by posters using the acronym FTFY (fixed that for you). We then train neural models on these pairs to edit the original claim and produce a new claim with a different view. We demonstrate significant improvement over a sequence-to-sequence baseline in BLEU score and a human evaluation for fluency, coherence, and contrast.

1 Introduction

In the Toulmin model (1958), often used in computational argumentation research, the center of the argument is the claim, a statement that is in dispute (Govier, 2010). In recent years, there has been increased interest in argument generation (Bilu and Slonim, 2016; Hua and Wang, 2018; Le et al., 2018). Given an argument, a system that generates counter-arguments would need to 1) identify the claims to refute, 2) generate a new claim with a different view, and 3) find supporting evidence for the new claim. We focus on this second task, which requires an understanding of contrast. A system that can generate claims with different views is a step closer to understanding and generating arguments (Apothloz et al., 1993). We build on previous work in automated claim generation (Bilu et al., 2015) which examined generating opposing claims via explicit negation. However, researchers also noted that not every claim has an exact opposite. Consider a claim from Reddit:

> Get employers out of the business, **pass universal single-payer healthcare**. (1)

This is an example of a policy claim - a view on what should be done (Schiappa and Nordin, 2013).

While negation of this claim is a plausible response (e.g. asserting there should be no change by stating *Do not get employers out of the business, do not pass universal healthcare*), negation limits the diversity of responses that can lead to a productive dialogue. Instead, consider a response that provides an alternative suggestion:

> Get employers out of the business, **deregulate and allow cross-state competition**. (2)

In Example 1, the speaker believes in an increased role for government while in Example 2, the speaker believes in a decreased one. As these views are on different sides of the political spectrum, it is unlikely that a single speaker would utter both claims. In related work, de Marneffe et al. (2008) define two sentences as contradictory when they are extremely unlikely to be true simultaneously. We thus define a contrastive claim as one that is *likely to be contradictory if made by the speaker of the original claim*. Our goal, then, is to develop a method for generating contrastive claims when explicit negation is not the best option. Generating claims in this way also has the benefit of providing new content that can be used for retrieving or generating supporting evidence.

In order to make progress towards generating contrastive responses, we need large, high-quality datasets that illustrate this phenomenon. We construct a dataset of 1,083,520 contrastive comment pairs drawn from Reddit using a predictive model to filter out non-contrastive claims. Each pair contains very similar, partially aligned text but the responder has significantly modified the original post. We use this dataset to model differences in views and generate a new claim given an original comment. The similarity within these pairs

Proceedings of NAACL-HLT 2019, pages 1756–1767
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

allows us to use them as distantly labeled contrastive word alignments. The word alignments provide semantic information about which words and phrases can be substituted *in context* in a coherent, meaningful way.

Our contributions[1] are as follows:

1. Methods and data for *contrastive claim identification* to mine comment pairs from Reddit, resulting in a large, continuously growing dataset of 1,083,520 distant-labeled examples.

2. A crowd-labeled set of 2,625 comments each paired with 5 new contrastive responses generated by additional annotators.

3. Models for *generating contrastive claims* using neural sequence models and constrained decoding.

In the following sections, we describe the task methodology and data collection and processing. Next, we present neural models for contrastive claim generation and evaluate our work and present an error analysis. We then discuss related work in contrast/contradiction, argumentation, and generation before concluding.

2 Task Definition and Motivation

Previous work in claim generation (Bilu et al., 2015) focused on explicit negation to provide opposing claims. While negation plays an important role in argumentation (Apothloz et al., 1993), researchers found that explicit negation may result in incoherent responses (Bilu et al., 2015). Furthermore, recent empirical studies have shown that arguments that provide new content (Wachsmuth et al., 2018) tend to be more effective. While new concepts can be introduced in other ways by finding semantically relevant content, we may find it desirable to explicitly model contrast in order to control the output of the model as part of a rhetorical strategy, e.g. concessions (Musi, 2018). We thus develop a model that generates a contrastive claim given an input claim.

Contrastive claims may differ in more than just viewpoint; they may also contain stylistic differences and paraphrases, among other aspects. We thus propose to model contrastive claims by controlling for context and maintaining the same text between pairs of contrastive claims except for the contrastive word or phrase. Much of the previous work in contrast and contradiction has examined the relationship between words or sentences. In order to understand when words and phrases are contrastive in argumentation, we need to examine them *in context*. For example, consider the claim *Hillary Clinton should be president.* A reasonable contrastive claim might be *Bernie Sanders should be president.* (rather than the explicit negation *Hillary Clinton should not be president.*) In this context, Hillary Clinton and Bernie Sanders are contrastive entities as they were both running for president. However, for the claim *Hillary Clinton was the most accomplished Secretary of State in recent memory.* they would be unrelated. Consider also that we could generate the claim *Hillary Clinton should be senator.* This contrastive claim is not coherent given the context. Generating a contrastive claim then requires 1) identifying the correct substitution span and 2) generating a response with semantically relevant replacements.

While some contrastive claims are not coherent, there are often multiple plausible responses, similar to tasks such as dialogue generation. For example, *Donald Trump should be president* is just as appropriate as *Bernie Sanders should be president*. We thus treat this as a dialogue generation task where the goal is to generate a plausible response given an input context.

3 Data

In order to model contrastive claims, we need datasets that reflect this phenomenon.

3.1 Collection

We obtain training data by scraping the social media site Reddit for comments containing the acronym *FTFY*.[2] FTFY is a common acronym meaning “fixed that for you.”[3] FTFY responses (hereafter FTFY) are used to respond to another comment by editing part of the “parent comment” (hereafter parent). Most commonly, FTFY is used for three categories of responses: 1) expressing a contrastive claim (e.g. the parent is ***Bernie Sanders*** *for president* and the FTFY is ***Hillary Clinton*** *should be president*) which may be sarcastic (e.g. ***Ted Cruz*** *for president* becomes ***Zodiac***

[1] Data and code available at github.com/chridey/fixedthat

[2] https://en.wiktionary.org/wiki/FTFY

[3] https://reddit.zendesk.com/hc/en-us/articles/205173295-What-do-all-these-acronyms-mean

killer** for president*) 2) making a joke (e.g. *This Python library really **piques** my interest* vs. *This really ***py*ques** my interest*), and 3) correcting a typo (e.g. *This **peaks** my interest* vs. ***piques). In Section 3.2, we describe how we identify category 1 (contrastive claims) for modeling.

To obtain historical Reddit data, we mined comments from the site pushshift.io for December 2008 through October 2017. This results in 2,200,258 pairs from Reddit, where a pair consists of a parent and an FTFY. We find that many of the top occurring subreddits are ones where we would expect strong opinions (/r/politics, /r/worldnews, and /r/gaming).

3.2 Classification

To filter the data to only the type of response that we are interested in, we annotated comment pairs for contrastive claims and other types. We use our definition of contrastive claims based on contradiction, where both the parent and FTFY are a claim and they are unlikely to be beliefs held by the same speaker. A joke is a response that does not meaningfully contrast with the parent and commonly takes the form of a pun, rhyme, or oronym. A correction is a response to a typo, which may be a spelling or grammatical error. Any other pair is labeled as "other," including pairs where the parent is not a claim.

In order to identify contrastive claims, we selected a random subset of the Reddit data from prior to September 2017 and annotated 1993 comments. Annotators were native speakers of English and the Inter-Annotator Agreement using Kripendorff's alpha was 0.72. Contrast occurs in slightly more than half of the sampled cases (51.4%), with jokes (23.0%) and corrections (21.2%) comprising about one quarter each. We then train a binary classifier to predict contrastive claims, thus enabling better quality data for the generation task.

To identify the sentence in the parent that the FTFY responds to and derive features for classification, we use an edit distance metric to obtain sentence and word alignments between the parent comment and response. As the words in the parent and response are mostly in the same order and most FTFYs contain significant overlap with the parent response, it is possible to find alignments by moving a sliding window over the parent. A sample of 100 comments verifies that this approach yields exact word alignments in 75 comments and exact sentence alignments in 93.

Given these pairs of comments, we derive linguistic and structural features for training a binary classifier. For each pair of comments, we compute features for the words in the *entire comment span* and features from the *aligned phrases span* only (as identified by edit distance). From the *aligned phrases* we compute the *character edit distance* and *character Jaccard similarity* (both normalized by the number of characters) to attempt to capture jokes and typos (the similarity should be high if the FTFY is inventing an oronym or correcting a spelling error). From the *entire comment*, we use the *percentage of characters copied* as a low percentage may indicate a poor alignment and the *percentage of non-ASCII characters* as many of the jokes use emojis or upside-down text. In addition, we use features from GloVe (Pennington et al., 2014) word embeddings[4] for both the *entire comment* and *aligned phrases*. We include the *percentage of words in the embedding vocabulary* for both spans for both the parent and FTFY. The reason for this feature is to identify infrequent words which may be typos or jokes. We compute the *cosine similarity* of the average word embeddings between the parent and FTFY for both spans. Finally, we use *average word embeddings* for both spans for both parent and FTFY.

As we want to model the generation of new content, not explicit negation, we removed any pairs where the difference was only "stop words." The set of stop words includes all the default stop words in Spacy[5] combined with expletives and special tokens (we replaced all URLs and usernames). We trained a logistic regression classifier and evaluated using 4-fold cross-validation. We compare to a *character* overlap baseline where any examples with Jaccard similarity > 0.9 and edit distance < 0.15 were classified as non-contrastive. The goal of this baseline is to illustrate how much of the non-contrastive data involves simple or nonexistent substitutions. Results are shown in Table 1. Our model obtains an F-score of 80.25 for an 8 point absolute improvement over the baseline.

3.3 Selection

After using the trained model to classify the remaining data, we have 1,083,797 Reddit pairs. We set aside 10,307 pairs from October 1-20, 2017 for

[4] We found the 50-dimensional Wikipedia+Gigaword embeddings to be sufficient

[5] spacy.io

Model	Precision	Recall	F-score
Majority	51.4	100	67.5
Baseline	67.75	77.19	72.16
LR	74.22	87.60	80.25

Table 1: Results of Identifying Contastive Claims

development and October 21-30 for test (6,773), with the remainder used for training. As we are primarily working with sentences, the mean parent length was 16.3 and FTFY length was 14.3.

The resulting test FTFYs are naturally occurring and so do not suffer from annotation artifacts. At the same time, they are noisy and may not reflect the desired phenomenon. Thus, we also conducted an experiment on Amazon Mechanical Turk[6] (AMT) to obtain additional gold references, which are further required by metrics such as BLEU (Papineni et al., 2002). We selected 2,625 pairs from the 10 most frequent categories[7] (see Table 2). These categories form a three-level hierarchy for each subreddit and we use the second-level, e.g. for /r/pokemongo the categories are "Pokémon", "Video Games", and "Gaming" so we use "Video Games." Before participating, each annotator was required to pass a qualification test - five questions to gauge their knowledge of that topic. For the movies category, one question we asked was whether for the sentence *Steven Spielberg is the greatest director of all time*, we could instead use *Stanley Kubrick* or *Paul McCartney*. If they passed this test, the annotators were then given the parent comment and "keywords" (the subreddit and three category levels) to provide additional context. We obtained *five* new FTFYs for each parent and validated them manually to remove obvious spam or trivial negation (e.g. "not" or "can't").

Category	Count	Category	Count
Video Games	1062	Basketball	116
Politics	529	Soccer	99
Football	304	Movies	88
Television	194	Hockey	60
World News	130	Baseball	55

Table 2: Comments for Mechanical Turk

4 Methods

Our goal of *generating* contrastive claims can be broken down into two primary tasks: 1) identifying the words in the original comment that should be removed or replaced and 2) generating the appropriate substitutions and any necessary context. Initially, we thus experimented with a modular approach by tagging each word in the parent and then using the model predictions to determine if we should copy, delete, or replace a segment with a new word or phrase. We tried the bi-directional LSTM-CNN-CRF model of Ma and Hovy (2016) and used our edit distance word alignments to obtain labels for copying, deleting, or replacing. However, we found this model performed slightly above random predictions, and with error propagation, the model is unlikely to produce fluent and accurate output. Instead, we use an end-to-end approach using techniques from machine translation.

4.1 Model

We use neural sequence-to-sequence encoder-decoder models (Sutskever et al., 2014) with attention for our experiments. The tokens from the parent are passed as input to a bi-directional GRU (Cho et al., 2014) to obtain a sequence of encoder hidden states h_i. Our decoder is also a GRU, which at time t generates a hidden state s_t from the previous hidden state s_{t-1} along with the input. When training, the input x_t is computed from the previous word in the gold training data if we are in "teacher forcing" mode (Williams and Zipser, 1989) and otherwise is the prediction made by the model at the previous time step. When testing, we also use the model predictions. The input word w_t may be augmented by additional features, as discussed in Section 4.2. In the baseline scenario $x_t = e(w_t)$ where e is an embedding. The hidden state s_t is then combined with a context vector h_t^*, which is a weighted combination of the encoder hidden states using an attention mechanism:

$$h_t^* = \sum_i \alpha_t^i h_i$$

To calculate α_i^t, we use the attention of Luong et al. (2015) as this encourages the model to select features in the encoder hidden state which correlate with the decoder hidden state, which we want because our input and output are similar. Our experiments on the development data verified this,

[6] We paid annotators the U.S. federal minimum wage and the study was approved by an IRB.

[7] Obtained from the snoopsnoo.com API

as Bahdanau attention (Bahdanau et al., 2015) performed worse. Attention is then calculated as:

$$\alpha_t^i = \frac{\exp(h_i^T s_t)}{\sum_{s'} \exp(h_{s'}^T s_t)}$$

Finally, we make a prediction of a vocabulary word w by using features from the context and decoder hidden state with a projection matrix W and output vocabulary matrix V:

$$P(w) = \text{softmax}(V \tanh(W[s_t; h_t^*] + b_w) + b_v)$$

We explored using a copy mechanism (See et al., 2017) for word prediction but found it difficult to prevent the model from copying the entire input.

4.2 Decoder Representation

Decoder Input: We evaluate two representations of the target input: as a sequence of **words** and as a sequence of **edits**. The sequence of words approach is the standard encoder-decoder setup. For the example parent *Hillary Clinton for president 2020* and FTFY *Bernie Sanders for president* we would use the FTFY without modification. Schmaltz et al. (2017) found success modeling error correction using sequence-to-sequence models by representing the target input as a sequence of edits. We apply a similar approach to our problem, generating a target sequence by following the best path in the matrix created by the edit distance algorithm. The new target sequence is the original parent interleaved with “DELETE-N tokens” that specify how many previous words to delete, followed by the newly generated content. For the same example, *Hillary Clinton for president 2020*, the modified target sequence would be *Hillary Clinton DELETE-2 Bernie Sanders for president 2020 DELETE-1*.

Counter: Kikuchi et al. (2016) found that by using an embedding for a length variable they were able to control output length via a learned mechanism. In our work, we compute a counter variable which is initially set to the number of new content words the model should generate. During decoding, the counter is decremented if a word is generated that is not in the source input (I) or in the set of stop words (S) defined in Section 3.2. The model uses an embedding $e(c_t)$ for each count, which is parameterized by a count embedding matrix. The input to the decoder state in this scenario is $x_t = e(w_t, c_t)$. At each time step, the count is computed by:

$$c_0 = |O \setminus (S \cup I)| \text{ or desired count}$$

$$c_{t+1} = \begin{cases} c_t - 1, & w_t \notin S \cup I \text{ and } c_t > 0 \\ c_t, & \text{otherwise} \end{cases}$$

where O is the set of gold output words in training.

For the parent comment *Hillary Clinton for president 2020* and FTFY *Bernie Sanders for president*, the decoder input is presented, with the time t in the first row of Table 3 and the inputs w_t and c_t in the second and third rows, respectively. At the start of decoding, the model expects to generate two new content words, which in this example it generates immediately and decrements the counter. When the counter reaches 0, it only generates stop or input words.

t	0	1	2	3	4
w_t	-	Bernie	Sanders	for	president
c_t	2	1	0	0	0

Table 3: Example of Counter

Unlike the controlled-length scenario, at test time we do not know the number of new content words to generate. However, the count for most FTFYs is between 1 and 5, inclusive, so we can exhaustively search this range during decoding. We experimented with predicting the count but found it to be inaccurate so we leave this for future work.

Subreddit Information: As the model often needs to disambiguate polysemous words, additional context can be useful. Consider the parent comment *this is a strange bug*. In a programming subreddit, a sarcastic FTFY might be *this is a strange feature*. However, in a Pokémon subreddit, an FTFY might be *this is a strange dinosaur* in an argument over whether Armaldo is a bug or a dinosaur. We thus include additional features to be passed to the encoder at each time step, in the form of an embedding g for each the three category levels obtained in Section 3.3. These embeddings are concatenated to the input word w_t at each timestep, i.e. $x_t = e(w_t, g_t^1, g_t^2, g_t^3)$.

4.3 Objective Function

We use a negative log likelihood objective function $\mathcal{L}_{NLL} = -\log \sum_{t \in 1:T} P(w_t^*)$, where w_t^* is the gold token at time t, normalized by each batch. We also include an additional loss term that uses the encoder hidden states to make a binary prediction over the input for whether a token will be copied or inserted/deleted. For the example from Section 4.2, the target for *Hillary Clinton for president 2020* would be *0 0 1 1 0*. This encourages the model to select features that indicate whether the encoder input will be copied to the output. We use a 2-layer multi-layer perceptron and a binary cross-entropy loss $\mathcal{L}_{BCE}$. The joint loss is then:

$$\mathcal{L} = \mathcal{L}_{NLL} + \lambda \mathcal{L}_{BCE}$$

4.4 Decoding

We use beam search for generation, as this method has proven effective for many neural language generation tasks. For the settings of the model that require a counter, we expand the beam by count m so that for a beam size k we calculate $k * m$ states.

Filtering: We optionally include a constrained decoding mode where we filter the output based on the counter; when $c_t > 1$ the end-of-sentence (EOS) score is set to $-\infty$ and when $c_t = 0$ the score of any word $w \in V \setminus (S \cup I)$ is set to $-\infty$. The counter c_t is decremented at every time step as in Section 4.2. In other words, when the counter is zero, we only allow the model to copy or generate stop words. When the counter is positive, we prevent the model from ending the sentence before it generates new content words and decrements the counter. The constrained decoding is possible with any combination of settings, with or without the counter embedding.

4.5 Hyper-parameters and Optimization

We used Pytorch (Paszke et al., 2017) for all experiments. We used 300-dimensional vectors for the word embedding and GRU layers. The count embedding dimension was set to 5 with $m = 5$ and $k = 10$ for decoding. The category embedding dimensions were set to 5, 10, and 25 for each of the non-subreddit categories. We also set $\lambda = 1$ for multi-task learning. We used the Adam optimizer (Kingma and Ba, 2015) with settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and a learning rate of 10^{-3} decaying by $\gamma = 0.1$ every epoch. We used dropout (Srivastava et al., 2014) on the embeddings with a probability of 0.2 and teacher forcing with 0.5. We used a batch size of 100 with 10 epochs, selecting the best model on the development set based on perplexity. We set the minimum frequency of a word in the vocabulary to 4.

5 Results

For training, development, and testing we use the data described in Section 3.3. The test reference data consists of the Reddit FTFYs and the FTFYs generated from AMT. We evaluate our models using automated metrics and human judgments.

Automated metrics should reflect our joint goals of 1) copying necessary context and 2) making appropriate substitutions. To address point 1, we use BLEU-4 as a measure of similarity between the gold FTFY and the model output. As the FTFY may contain significant overlap with the parent, BLEU indicates how well the model copies the appropriate context. As BLEU reflects mostly span selection rather than the insertion of new content, we need alternative metrics to address point 2. However, addressing point 2 is more difficult due to the variety of possible substitutions, including named entities. For example, if the parent comment is *jaguars for the win!* and the gold FTFY is *chiefs for the win!* but the model produces *cowboys for the win!* (or any of 29 other NFL teams), most metrics would judge this response incorrectly even though it would be an acceptable response. Thus we present results using both automated metrics and human evaluation. As an approximation to address point 2, we attempt to measure when the model is making changes rather than just copying the input. To this end, we present two additional metrics - *novelty*, a measure of whether novel content (non-stop word) tokens are generated relative to the parent comment, and *partial match*, a measure of whether the novel tokens in the gold FTFY match any of the novel tokens in the generated FTFY. To provide a reference point, we find that the partial match between two different gold FTFYs (Reddit and AMT) was 11.4% and BLEU was 47.28, which shows the difficulty of automatic evaluation. The scores are lower than expected because the Reddit FTFYs are noisy due to the process in Section 3.2. This also justifies obtaining the AMT FTFYs.

Results are presented in Table 4. The baseline is a sequence-to-sequence model with attention. For other components, the counter embedding is referred to as “COUNT,” the category/subreddit embeddings as “SUB,” the sequence of edits as “EDIT,” and the multi-task copy loss as “COPY.” The models in the top half of the table use constrained decoding and those in the bottom half are unconstrained, to show the learning capabil-

			Reddit		AMT	
	Model	Novelty	BLEU-4	% Match	BLEU-4	% Match
Constrained	Baseline	79.88	18.81	4.67	40.14	10.06
	COUNT	89.69	22.61	4.72	47.55	12.55
	COUNT + SUB + COPY	**90.45**	**23.13**	**4.83**	**50.05**	**14.92**
	EDIT	64.64	16.12	3.37	35.48	7.33
	EDIT + COUNT + SUB + COPY	82.96	19.37	4.23	42.69	11.62
Unconstrained	Baseline	3.34	7.31	0.73	25.83	0.68
	COUNT	16.19	8.51	1.95	27.68	2.36
	COUNT + SUB + COPY	16.26	9.62	1.93	31.23	3.81
	EDIT	7.97	**35.41**	1.57	**74.24**	1.56
	EDIT + COUNT + SUB + COPY	**39.99**	32.59	**3.25**	67.56	**6.25**

Table 4: Automatic Evaluation

ities of the models. For each model we compute statistical significance with bootstrap resampling (Koehn, 2004) for the constrained or unconstrained baseline as appropriate and we find the COUNT and EDIT models to be significantly better for constrained and unconstrained decoding, respectively ($p < 0.005$).

Under constrained decoding, we see that the "COUNT + SUB + COPY" model performs the best in all metrics, although most of the performance can be attributed to the count embedding. When we allow the model to determine its own output, we find that "EDIT + COUNT" performs the best. In particular, this model does well at understanding which part of the context to select, and even does better than other unconstrained models at selecting appropriate substitutions. However, when we combine this model with constrained decoding, the improvement is smaller than for the other settings. We suspect that because the EDIT model often needs to generate a DELETE-N token before a new response, these longer-term dependencies are hard to capture with constrained decoding but easier if included in training.

We also conducted a human evaluation of the model output on the same subset of 2,625 examples described in Section 3.3. We performed an additional experiment on AMT where we asked annotators to rate responses on fluency, coherence, and contrast. **Fluency** is a measure of the quality of the grammar and syntax and the likelihood that a native English speaker would utter that statement. **Coherence** is a measure of whether the response makes sense, is semantically meaningful, and would be usable as a response to a claim. **Contrast** is a measure of how much the response contradicts the original comment. We specified that if the response is different but does not provide a contrasting view it should receive a low rating. Previous work (Bilu et al., 2015) used fluency, clarity/usability (which we combine into coherence), and opposition (where we use contrast).

Model	Fluency	Coherence	Contrast
Gold	4.34	4.26	3.01
Baseline	3.49	3.19	1.94
Constrained	3.46	3.32	2.53
Best	**3.52**	**3.46**	**2.87**

Table 5: Human Evaluation

We used a Likert scale where 5 is strongly agree and 1 is strongly disagree. We used the same data and qualification test from Section 3.3 for each category and used three annotators per example. We asked annotators to judge 4 different pairs: 3 model outputs and the gold Reddit[8] FTFYs for comparison. We include the baseline, the baseline with constrained decoding, and the best constrained model ("COUNT + SUB + COPY") according to BLEU and partial match. We verified that the annotators understood how to rate contrast by examining the distribution of responses: the annotators selected option 3 (neither) 15% of the time and preferred to select either extreme, 5 (21%) or 1 (27%). Results are presented in Table 5, showing a clear preference for the best model. Note the degradation in fluency for the constrained baseline, as the model is prevented from generating the EOS token and may repeat tokens up to the

[8] We did not evaluate the AMT FTFYs as these were generated by the same pool of annotators.

Parent: ah yes the wonders of the free market
Model: ah yes the wonders of government intervention
Parent: i know that this is an unofficial mod , but xp is the best os for this machine
Model: linux is the best os for this machine
Parent: that 's why it 's important to get all your propaganda from infowars and brietbart
Model: propaganda from fox news outlets

Table 6: Model Output

maximum length.

6 Qualitative Analysis

We provide three examples of the model output in Table 6 with the first and third from the News and Politics category, demonstrating how the model handles different types of input. In the first example, the contrast is between allowing markets to regulate themselves versus an increased role of government. In the second example, the contradiction is due to the choice of operating system. In the third (invalid) example, the model responds to a sarcastic claim with another right-wing news organization; this response is not a contradiction since it is plausible the original speaker would also utter this statement.

6.1 Error Analysis

We conduct an error analysis by selecting 100 responses where the model did not partially match any of the 6 gold responses and we found 6 main types of errors. One error is the model identifying an **incorrect substitution span** while the human responses all selected a different span to replace. We noticed that this occurred 5 times and may require world knowledge to understand which tokens to select. For example, in response to the claim *Hillary Clinton could have been president if not for robots*, the model generates *Donald Trump* in place of *Hillary Clinton*, whereas the gold responses generate *humans / votes / Trump's tweets* in place of *robots*. Another type of error is when the responses are not coherent with the parent and the language model instead determines the token selection based on the **most recent context** (11 cases). For example, given the claim *bb-8 gets a girlfriend and poe still does n't have a girlf :')* the Reddit FTFY has *boyf* instead of *girlf* whereas the model generates ... *and poe still does n't have a clue what i 'm talking about* . We also found examples where the model chose poorly due to unfiltered jokes or **errors in the training data** (12 in total). In 15 cases, due to the constrained decoding the model **repeated** a word until the maximum length or appended an incoherent phrase. For the most common error, the model made a substitution that **was not contrasting** as in Table 6 (19 examples). Finally, we found 38 of the samples were **valid** responses, but did not match the gold, indicating the difficulty of automatic evaluation. For example, in response to the claim *Nintendo is the only company that puts customers over profits*, the model replaces *Nintendo* with *Rockstar* (both video game companies) while the gold FTFYs had other video game companies.

7 Related Work

Understanding contrast and contradiction is key to argumentation as it requires an understanding of differing points-of-view. Recent work examined the negation of claims via explicit negation (Bilu et al., 2015). Other work investigated the detection of different points-of-view in opinionated text (Al Khatib et al., 2012; Paul et al., 2010). Wachsmuth et al. (2017; 2018) retrieved arguments for and against a particular stance using online debate forums. In non-argumentative text, researchers predicted contradictions for types such as negation, antonyms, phrasal, or structural (de Marneffe et al., 2008) or those that can be expressed with functional relations (Ritter et al., 2008). Other researchers have incorporated entailment models (Kloetzer et al., 2013) or crowdsourcing methods (Takabatake et al., 2015). Contradiction has also become a part of the natural language inference (NLI) paradigm, with datasets labeling contradiction, entailment, or neutral (Bowman et al., 2015a; Williams et al., 2018). The increase in resources with contrast and contradiction has resulted in new representations with contrastive meaning (Chen et al., 2015; Nguyen et al., 2016; Vulić, 2018; Conneau et al., 2017). Most of this work has focused on identifying contrast or contradiction while we aim to generate contrast. Furthermore, while contradiction and contrast are present in these corpora, we obtain distant-labeled alignments for contrast at the word and phrase level. Our dataset also includes contrastive concepts and entities while other corpora primarily contain antonyms and explicit negation.

Contrast also appears in the study of stance, where the opinion towards a target may vary. The SemEval 2016 Stance Detection for Twitter task (Mohammad et al., 2016) involved predicting if a tweet favors a target entity. The Interpretable Semantic Similarity task (Agirre et al., 2016) called to identify semantic relation types (including opposition) between headlines or captions. Target-specific stance prediction in debates is addressed by Hasan and Ng (2014) and Walker et al. (2012). Fact checking can be viewed as stance toward an event, resulting in research on social media (Lendvai and Reichel, 2016; Mihaylova et al., 2018), politician statements (Vlachos and Riedel, 2014), news articles (Pomerleau and Rao, 2017), and Wikipedia (Thorne et al., 2018).

In computational argumentation mining, identifying claims and other argumentative components is a well-studied task (Stab and Gurevych, 2014). Daxenberger et al. (2017) and Schulz et al. (2018) developed approaches to detect claims across across diverse claim detection datasets. Recently, a shared task was developed for argument reasoning comprehension (Habernal et al., 2018). The best system (Choi and Lee, 2018) used models pre-trained on NLI data (Bowman et al., 2015b), which contains contradictions. While this work is concerned with identification of argumentative components, we propose to generate new claims.

In the field of argument generation, Wang and Ling (2016) train neural abstractive summarizers for opinions and arguments. Additional work involved generating opinions given a product rating (Wang and Zhang, 2017). Bilu and Slonim (2016) combine topics and predicates via a template-based classifier. This work involves the generation of claims but in relation to a topic. Other researchers generated political counter-arguments supported by external evidence (Hua and Wang, 2018) and generating argumentative dialogue by maximizing mutual information (Le et al., 2018). This research considers end-to-end argument generation, which may not be coherent, whereas we focus specifically on contrastive claims.

8 Conclusion

We presented a new source of over 1 million contrastive claim pairs that can be mined from social media sites such as Reddit. We provided an analysis and models to filter noisy training data from 49% down to 25%. We created neural models for generating contrastive claims and obtained significant improvement in automated metrics and human evaluations for Reddit and AMT test data.

Our goal is to incorporate this model into an argumentative dialogue system. In addition to generating claims with a contrasting view, we can also retrieve supporting evidence for the newly-generated claims. Additionally, we plan to experiment with using our model to improve claim detection (Daxenberger et al., 2017) and stance prediction (Bar-Haim et al., 2017). Our model could be used to generate artificial data to enhance classification performance on these tasks.

To improve our model, we plan to experiment with retrieval-based approaches to handle low-frequency terms and named entities, as sequence-to-sequence models are likely to have trouble in this environment. One possibility is to incorporate external knowledge with entity linking over Wikipedia articles to find semantically-relevant substitutions.

Another way to improve the model is by introducing controllable generation. One aspect of controllability is intention; our model produces contrastive claims without understanding the view of the original claim. Category embeddings partially address this issue (some labels are “Liberal” or “Conservative”), but labels are not available for all views. Going forward, we hope to classify the viewpoint of the original claim and then generate a claim with a desired orientation. Furthermore, we hope to improve on the generation task by identifying the types of claims we encounter. For example, we may want to change the target of the claims in some claims but in others change the polarity.

We also plan to improve the dataset by improving our models for contrastive pair prediction to reduce noise. Finally, we hope that this dataset proves useful for related tasks such as textual entailment (providing examples of contradiction) and argument comprehension (providing counter-examples of arguments) or even unrelated tasks like humor or error correction.

Acknowledgments

We would like to thank the AMT annotators for their work and the anonymous reviewers for their valuable feedback. The authors also thank Alyssa Hwang for providing annotations and Smaranda Muresan, Chris Kedzie, Jessica Ouyang, and Elsbeth Turcan for their helpful comments.

References

Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. 2016. Semeval-2016 task 2: Interpretable semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 512–524. Association for Computational Linguistics.

Khalid Al Khatib, Hinrich Schütze, and Cathleen Kantner. 2012. Automatic detection of point of view differences in wikipedia. In *Proceedings of COLING 2012*, pages 33–50. The COLING 2012 Organizing Committee.

Denis Apothloz, Pierre-Yves Brandt, and Gustavo Quiroz. 1993. The function of negation in argumentation. *Journal of Pragmatics*, 19:23–38.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261, Valencia, Spain. Association for Computational Linguistics.

Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. 2015. Automatic claim negation: Why, how and when. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 84–93. Association for Computational Linguistics.

Yonatan Bilu and Noam Slonim. 2016. Claim synthesis via predicate recycling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 525–530, Berlin, Germany. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015a. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015b. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Hui Jiang, and Xiaodan Zhu. 2015. Revisiting word embedding for contrasting meaning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 106–115. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

HongSeok Choi and Hyunju Lee. 2018. Gist at semeval-2018 task 12: A network transferring inference knowledge to argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 773–777. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics.

Johannes Daxenberger, Steffen Eger, Ivan Habernal, Christian Stab, and Iryna Gurevych. 2017. What is the essence of a claim? cross-domain claim identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2066. Association for Computational Linguistics.

Trudy Govier. 2010. *A Practical Study of Argument*. Cengage Learning, Wadsworth.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. 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 1930–1940. Association for Computational Linguistics.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762. Association for Computational Linguistics.

Xinyu Hua and Lu Wang. 2018. Neural argument generation augmented with externally retrieved evidence. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, Melbourne, Australia. Association for Computational Linguistics.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Julien Kloetzer, Stijn De Saeger, Kentaro Torisawa, Chikara Hashimoto, Jong-Hoon Oh, Motoki Sano, and Kiyonori Ohtake. 2013. Two-stage method for large-scale acquisition of contradiction pattern pairs using entailment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 693–703. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Dieu-Thu Le, Cam Tu Nguyen, and Kim Anh Nguyen. 2018. Dave the debater: a retrieval-based and generative argumentative dialogue agent. In *Proceedings of the 5th Workshop on Argument Mining*, pages 121–130. Association for Computational Linguistics.

Piroska Lendvai and Uwe Reichel. 2016. Contradiction detection for rumorous claims. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM)*, pages 31–40. The COLING 2016 Organizing Committee.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio. Association for Computational Linguistics.

Tsvetomila Mihaylova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Mitra Mohtarami, Georgi Karadzhov, and James R. Glass. 2018. Fact checking in community forums. *CoRR*, abs/1803.03178.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41. Association for Computational Linguistics.

Elena Musi. 2018. How did you change my view? a corpus-based study of concessions argumentative role. *Discourse Studies*, 20(2):270–288.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459. Association for Computational Linguistics.

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

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Michael Paul, ChengXiang Zhai, and Roxana Girju. 2010. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 66–76. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge.

Alan Ritter, Stephen Soderland, Doug Downey, and Oren Etzioni. 2008. It's a contradiction – no, it's not: A case study using functional relations. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 11–20. Association for Computational Linguistics.

E. Schiappa and J.P. Nordin. 2013. *Argumentation: Keeping Faith with Reason*. Pearson Education.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting sequence models for sentence correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813, Copenhagen, Denmark. Association for Computational Linguistics.

Claudia Schulz, Steffen Eger, Johannes Daxenberger, Tobias Kahse, and Iryna Gurevych. 2018. Multi-task learning for argumentation mining in low-resource settings. *CoRR*, abs/1804.04083.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Yu Takabatake, Hajime Morita, Daisuke Kawahara, Sadao Kurohashi, Ryuichiro Higashinaka, and Yoshihiro Matsuo. 2015. Classification and acquisition of contradictory event pairs using crowdsourcing. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 99–107. Association for Computational Linguistics.

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

Stephen Toulmin. 1958. *The Uses of Argument*. Cambridge At the University Press.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22. Association for Computational Linguistics.

Ivan Vulić. 2018. Injecting lexical contrast into word vectors by guiding vector space specialisation. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 137–143. Association for Computational Linguistics.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, Copenhagen, Denmark. Association for Computational Linguistics.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251. Association for Computational Linguistics.

Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 592–596. Association for Computational Linguistics.

Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 47–57, San Diego, California. Association for Computational Linguistics.

Zhongqing Wang and Yue Zhang. 2017. Opinion recommendation using a neural model. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1627–1638. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, 1(2):270–280.