ParaTag: A Dataset of Paraphrase Tagging for Fine-Grained Labels, NLG Evaluation, and Data Augmentation

Microsoft Azure Cognitive Services Research
{shuowa, ruox, yaliu10, chezhu, nzeng}@microsoft.com

Abstract
Paraphrase identification has been formulated as a binary classification task to decide whether two sentences hold a paraphrase relationship. Existing paraphrase datasets only annotate a binary label for each sentence pair. However, after a systematical analysis of existing paraphrase datasets, we found that the degree of paraphrase cannot be well characterized by a single binary label. And the criteria of paraphrase are not even consistent within the same dataset. We hypothesize that such issues would limit the effectiveness of paraphrase models trained on these data. To this end, we propose a novel fine-grained paraphrase annotation schema that labels the minimum spans of tokens in a sentence that don’t have the corresponding paraphrases in the other sentence. Under this setting, we frame paraphrasing as a sequence tagging task. We collect 30k sentence pairs in English with the new annotation schema, resulting in the ParaTag dataset. In addition to reporting baseline results on ParaTag using state-of-art language models, we show that ParaTag is especially useful for training an automatic scorer for language generation evaluation. Finally, we train a paraphrase generation model from ParaTag and achieve better data augmentation performance on the GLUE benchmark than other public paraphrasing datasets.

1 Introduction
Paraphrase identification and paraphrase generation are important problems in natural language processing. They are the essential tasks to verify whether machine learning models genuinely understand the text. A lot of research (Das and Smith, 2009; Yin et al., 2016; Devlin et al., 2019; He et al., 2021) is devoted to identifying paraphrase between sentences, on various annotated datasets such as Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), Quora Question Pairs Dataset (QQP)2(Wang et al., 2017a), and Paraphrase Adversaries from Word Scrambling (PAWS) (Zhang et al., 2019). Similarly, paraphrase generation has prompted various works (Wieting and Gimpel, 2018; Hu et al., 2019; Chen et al., 2019) to investigate how to generate diverse sentences mainly for data augmentation.

High-quality paraphrasing datasets are pivotal in both paraphrase identification and paraphrase generation. Most existing paraphrasing datasets (Dolan and Brockett, 2005; Wang et al., 2017a; Cer et al., 2017; Williams et al., 2018) adopt the binary label schema, i.e., collecting a pair of sentences and having labelers annotate a single binary label to indicate whether the two sentences are paraphrases of each other or not. However, directly identifying whether a pair of sentences are paraphrases is usually not a trivial task, and the level of agreement among labelers could be low (Wang et al., 2021).

Table 1: Examples of tags in ParaTag. M: sentences from MRPC. G: sentences generated by GPT-3. The phrases in bold are judged by human annotators as not being paraphrased by the other sentence. Note that the first sentence pair is originally labeled as paraphrase in MRPC. However, the fine-grained annotation indicates that not all the information is paraphrased.

<table>
<thead>
<tr>
<th>No.</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-1</td>
<td>However, commercial use of the 2.6.0 kernel is still months off for most customers.</td>
</tr>
<tr>
<td>M-2</td>
<td>Commercial releases of the 2.6 kernel by major Linux distributors still remain months away.</td>
</tr>
<tr>
<td>G-1</td>
<td>It covers some of the same ground as Spotlight but has nothing new to add.</td>
</tr>
<tr>
<td>G-2</td>
<td>It revisits the same topic as Spotlight, but does not say anything new.</td>
</tr>
<tr>
<td>G-3</td>
<td>Heavy rainstorms have caused floods in the area every year since 2003.</td>
</tr>
<tr>
<td>G-4</td>
<td>Since 2003, the area has been hit every year by heavy rainstorms and floods.</td>
</tr>
</tbody>
</table>

* Equal contribution.

1 Data and code: https://github.com/microsoft/ParaTag

2https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
The main reason is that a sentence could contain multiple pieces of information, i.e., thought groups, and two sentences may only partially overlap on a few thought groups. This fine-grained information cannot be captured by a single binary paraphrase label. For instance, we show a sentence pair labeled as paraphrase in MRPC dataset in the first row of Table 1. However, upon close examination, we find that the phrases “most customers” and “major Linux distributors” do not have corresponding paraphrases in the other sentence.

To this end, we build a large-scale and diverse fine-grained paraphrasing dataset, ParaTag. For each sentence in a pair, the annotators must tag all spans that do not have the corresponding paraphrase in the other sentence. Thus, we frame paraphrase detection as a sequence tagging problem, with a binary label for each word in each sentence.

First, we hire labelers to re-annotate MRPC dataset with the fine-grained labeling schema. It turns out that actually 90% of the sentence pairs labeled as paraphrase in MRPC dataset are tagged with at least one phrase not being paraphrased by the other sentence. Then, to improve the diversity of the dataset, we generate more sentence pairs by feeding paraphrases in MRPC as prompts to the language model GPT-3 (Brown et al., 2020). In total, we collect 30k sentence pairs and 840k word-level annotations in ParaTag dataset. We then fine-tune a variety of pre-trained language models on ParaTag and report the result.

Next, we use the above sequence labeling models (Tsai et al., 2019) trained on ParaTag as evaluators for natural language generation tasks like machine translation and summarization. Different from rule-based heuristic methods such as BLEU (Papineni et al., 2002) and unsupervised alignment-based methods such as BERTScore (Zhang et al., 2020), a sequence labeling model trained on the word-level annotations in ParaTag can provide a more accurate comparison between reference output and NLG model’s output. On Summarization and Data2Text benchmarks, we show that models trained on ParaTag outperform existing matching based NLG evaluator models by 13.0% and 6.0% respectively, and outperform BARTScore (Yuan et al., 2021) trained with ParaBank (Hu et al., 2019) by 3.8% and 2%.

Finally, we train a paraphrase generator based on sentence pairs in ParaTag for data augmentation. Experiments show that the generator model trained on ParaTag can reach better data augmentation quality compared with models trained on existing paraphrasing datasets and further improves the BERT-base (Devlin et al., 2019) model by 1.5% on GLUE dataset.

We summarize our contributions in this work as follows:

1. We collect a novel word-level paraphrasing dataset ParaTag in English to characterize the fine-grained relationship between sentence pairs better.3
2. The paraphrase detection model built on ParaTag can be used as a better evaluator for NLG tasks.
3. The paraphrase generation model built on ParaTag can improve data augmentation quality.

2 Related Work

Paraphrase identification Paraphrase identification is to identify whether a pair of sequences is a paraphrase or not, such as Microsoft Research Paraphrase Corpus (MRPC) Dataset (Dolan and Brockett, 2005). Another similar dataset, Quora Question Pairs dataset4(Wang et al., 2017a) is to classify whether two questions are duplicated or not. Besides, Lan et al. (2017) collect a sentential paraphrase dataset based on Twitter. Unlike the above human annotated dataset on the sentence level, we focus on more fine-grain labeling. We ask annotators to tag all the differences between sentences in word level without explicitly distinguish between important and unimportant differences (Kovatchev et al., 2018; Gold et al., 2019). Further research can be done to loose our hard restrictions by involving more syntactic information to identify the importance.

Paraphrase bank All the above datasets are formulated as a binary classification task and annotated by humans. Another series of work is to collect a large-scale paraphrase dataset for paraphrase generation. Large-scale paraphrase corpora, such as ParaBank (Hu et al., 2019) and ParaNMT-50M (Wieting and Gimpel, 2018), are usually based on unsupervised methods, like back-translation and words co-occurrence. Dong et al. (2021) discover paraphrases in scientific field by

3We will release our data, code, and model checkpoints after the anonymity period.
4https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
leveraging BERT representations. In our dataset, all of the paraphrases are verified by humans. Moreover, to keep the diversity, all the sentence pairs are generated by GPT-3 and the BLEU score of the sentence pair is not higher than 20.

**NLG evaluator** Paraphrase related tasks strongly correlate to the natural language generation (NLG) evaluators, which give higher scores when generated sequences can paraphrase reference sequences. BLEU (Papineni et al., 2002) and Rouge (Lin, 2004) are the most widely used evaluation metric for NLG tasks, such as summarization, machine translation, etc. Meteor (Banerjee and Lavie, 2005) further involve WordNet to match sequences. With the success of pre-training language models (PLM), they are also used for NLG evaluators. BERTScore (Zhang et al., 2020) makes use of the similarities between all the hidden states in different sequences to evaluate the generation model. Unlike the unsupervised matching based method above, our dataset provides the label to tell the matched phrases. Models trained with our dataset can better align with human evaluation. Moreover, the paraphrases from our dataset can also be used for the generation based methods, such as PRISM (Thompson and Post, 2020) BARTScore (Yuan et al., 2021). We can further improve BARTScore based on our new dataset according to our experiment results. Our method will not train with any in-domain data for scoring, such as BLEURT (Sellam et al., 2020) and COMET (Rei et al., 2020) trained with Machine Translation scoring data.

3 Rethinking Paraphrase Dataset

Our work is motivated by a retrospect of the widely used MRPC paraphrasing dataset (Dolan and Brockett, 2005) via a more strict standard of paraphrasing. Specifically, we define that a pair of sentences are paraphrases only if they contain the same meanings for all semantic units. For instance, in the first row of Table 1, even though the two sentences convey the same high-level meanings, the “Linux distributors” in the second sentence does not have a corresponding paraphrase in the first sentence. Thus, if this sentence pair is used as a golden paraphrase pair (which it is in MRPC), it could contaminate the model with hallucinations and suspicious correlations, e.g., “kernel” would always co-occur with “Linux”. Using this strict criterion, we asked labelers to re-annotate all sentence pairs in the MRPC dataset. Among the 1604 sentence pairs with paraphrase labels in MRPC, the new annotations show that 90% of them are not strictly paraphrases of each other.

Surprised by the fact that most of the “paraphrased” sentence pairs in MRPC are not strict paraphrases, we ask the following research questions:

1. Given the small fraction (10%) of sentence pairs labeled as strict paraphrases, is a single sentence-level binary label still the best annotation schema?

   We believe the sentence-level binary labeling schema is not suitable because the labelers may ignore some non-paraphrase phrases by hallucination, which leads to false positives, as in the example mentioned above. Thus, we propose a novel annotation schema for fine-grained paraphrasing labels in Section 4.2.

2. In addition to enforcing a strict paraphrasing criterion, can we further improve the diversity of sentences in the dataset?

   To increase the data diversity, we leverage the strong generation capacity of GPT-3 to produce candidate sentence pairs. We then filter the generated data to make sure it is non-trivial to judge
the paraphrasing relationship between sentences in each. We describe the details in Section 4.1.

(3) What downstream tasks can benefit from these fine-grained and diverse paraphrase annotations?

In natural language generation (NLG) evaluation, one needs to compare a system prediction $S$ with a reference $R$ and output a score representing how well $S$ aligns with $R$. A good evaluator model should produce scores highly correlated with human judgment. Humans are sensitive to hallucinations, modifiers, and factual inconsistencies when judging a NLG system. Thus, we believe that a model trained on fine-grained and diverse annotations can better capture these delicate aspects in a sentence, which is verified by detailed experiments in Section 6.

Additionally, given the fine-grained word-level labels in ParaTag, we can select a subset of sentence pairs with the highest degrees of paraphrasing to train a data augmentation (DA) model. Section 6 shows that DA models trained on ParaTag are more effective than those trained on other public paraphrasing datasets.

4 ParaTag Dataset

In this section, we will introduce the details of our dataset annotation. We hire 4 native speakers, who come from an annotation company and are trained by a program manager based on our instructions, to annotate the dataset for 5 weeks with around $24K in total. We have two targets in building this dataset: 1) annotating fine-grained word-level paraphrasing labels between sentences, and 2) a high quality and diverse paraphrase corpus. Thus, the labeling process is partitioned into the following three phases (Figure 1).

4.1 Pairwise Data Collection

Asking annotators to paraphrase a given sentence directly is always time-consuming, and it is hard to control the diversity. Thus, we reuse sentence pairs in MRPC dataset and then also leverage GPT-3 (Brown et al., 2020) to generate more pairwise data for annotation, as shown in Figure 1 (a).

As GPT-3 adopts prompt-based inference, we randomly select 20 paraphrasing sentence pairs into the prompt and let GPT-3 generate another 300 tokens. To construct prompt, we use the template like “Premise: However, commercial use of the 2.6.0 kernel is still months off for most customers. Paraphrase: Commercial releases of the 2.6 kernel by major Linux distributors still remain months away.”. After GPT-3 generation, we only keep the decoded sentence pairs containing the pattern “Premise: … Paraphrase: …”. In the first week, the GPT-3 prompts come from sampling the sentence pairs annotated as paraphrase from MRPC dataset. In the following weeks, the GPT-3 prompts will leverage the sentence pairs annotated from the previous week.

To improve the quality and diversity of candidate pairs generated from GPT-3, we only keep sentence pairs with a intra-pair BLEU score larger than 5 and less than 20 for annotation. Furthermore, both sentences should have at least 10 characters, and the longer sentence should have fewer than 2.5 times the number of words in the shorter sentence in the pair.

4.2 Paraphrase Tagging

We then ask the labelers to tag the sentence pairs with the following instructions, with examples in Table 1.

1. Highlight a set of spans in each sentence that the other sentence cannot paraphrase. We require each span in this set to be a minimum span, which should be the shortest consecutive words/phrases/clauses that can be modified to make the sentence pair paraphrases. For example, if one sentence mentions “2.5 billion dollars” and the other contains “1.6 billion dollars”, only 2.5 and 1.6 need to be highlighted.

2. If the two sentences have completely different meaning, highlight the whole sentences. However, this is only allowed when no shorter spans can be identified.

3. If the two sentences are paraphrases, no highlights are needed.

As described above, we frame paraphrasing dataset labeling as a sequence tagging problem, with highlight (1) or non-highlight (0) for each word in each sentence.

We first ask the labelers to annotate training data, of which 90% instances are annotated once, and 10% instances are annotated twice by different labelers to track the annotation quality. When the inter-agreement goes lower, we will ask the program manager to re-train the labelers. We compute the Cohen’s kappa coefficient (Smeeton, 1985) on
word-level annotation to measure the agreement level between labelers on the 10% instances. The value is 0.61 for annotations on the sentence pairs from MRPC and 0.55 for annotations on sentence pairs generated by GPT-3.

Next, we ask labelers to annotate validation and test sets. 800 instances are first annotated based on the discussion and agreement between 4 labelers. Then we split it half for validation set and half for test set, as shown in Table 2 “val” and “test1”. To assess the agreement level between annotators, we collect another test set. It contains sentence pairs independently annotated by 4 labelers split into two groups. For each group, we filter the annotations with strong disagreement on labeling (the f1 score of two tagging sequences are smaller than 0.4). This leads to another test set of 310 instances (“test2” in Table 2). We randomly select one annotation from each group to compute Cohen’s kappa coefficient which is 0.663. And we treat one annotation as golden label to compute human performance on this set, F1 0.751 and accuracy 0.869. In total, ParaTag consists of 28,671, 400, and 710 sentence pairs in training, validation and test sets, respectively, as shown in Table 2.

4.3 High-quality Paraphrases Collection

After the above annotation process, we find only 9.5% of the data, 2k, are paraphrases without highlighted annotations. It means prompt-based GPT-3 is still not good at generating paraphrases. Due to the budget limit, to efficiently collect more diverse and high-quality paraphrases for building a data augmentation model later. We first build a sequence tagging model on previously collected data. This model is then applied to sentence pairs generated by GPT-3. We ask labelers to annotate word-level paraphrase tags for the sentence pairs which receive no highlights from the model. Finally, in this set, 47.6% of the data also receive no highlights from labelers, as shown in Table 2 Train2. In the end, we get 5,461 paraphrases in total.

5 Model

This section introduces how to train models on the ParaTag dataset and apply them to various tasks.

5.1 Paraphrasing Detection

Based on ParaTag annotations, we formulate paraphrasing detection as a sequence labeling task (Figure 2 (a)), which identifies all spans that the other sentence in the pair cannot paraphrase. To do that,
we first tokenize both sentences into words and then concatenate two sentences with a special token “[SEP]”. The new sequence is encoded by Transformer, DeBERTa-v3 (He et al., 2021). The hidden states of each word in the last layer are used for token-level classification with cross-entropy loss. Formally,

\[ H = \text{Transformer}([X_1; [SEP]; X_2]) \quad (1) \]
\[ G = WH \quad (2) \]
\[ \mathcal{L} = \text{CrossEntropyLoss}(G, Y), \quad (3) \]

where \( X_1 \) and \( X_2 \) are the two tokenized sentences, with \( l_1 \) and \( l_2 \) words respectively. \( H \in \mathbb{R}^{d \times (l_1+l_2)} \) is the hidden states of Transformer in the final layer. Here, the hidden state of the special token [SEP] is not used. And if a word consists of multiple subwords, we use the hidden states of its first subword. \( W \in \mathbb{R}^{2 \times d} \) is learnable weights to compute the logits \( G \in \mathbb{R}^{2 \times (l_1+l_2)}. \) \( Y \in \mathbb{R}^{l_1+l_2} \) is the annotated binary labels in ParaTag.

5.2 NLG Evaluator

The goal is to build a scorer model to evaluate model outputs against ground-truth reference outputs in NLG tasks. We design two types of NLG evaluators based on ParaTag datasets.

Matching to score Here, the model needs to find word or phrase level matching between two sequences to compute the evaluation score. Typical evaluators of this type include ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020). We directly employ the paraphrasing detection model built in the previous subsection to compute the score, as shown in Figure 2 (b). Specifically, suppose \( X_1 \) is the model output and \( X_2 \) is the reference:

\[ S = G[0,:]-G[1,:], \quad (4) \]
\[ FScore = \text{Mean}(S), \quad (5) \]
\[ PScore = \text{Mean}(S[0:l_1]), \quad (6) \]
\[ RScore = \text{Mean}(S[l_1:l_1+l_2]), \quad (7) \]

where \( G[0,:] \in \mathbb{R}^{l_1+l_2} \) are all the predicted logits for word-level paraphrase and \( G[1,:] \) are for non-paraphrases. \( S \in \mathbb{R}^{l_1+l_2} \) is thus the word-level scores for the sequence pair. FScore, PScore and RScore are computed as the mean value over the whole sequence pair, over \( X_1 \) and over \( X_2 \) respectively. These scores will be used for meta-evaluation based on different perspectives of model evaluation. In details, PScore can be used for checking factuality in Summarization tasks, including SummEval, Q-CNN, Q-XSUM datasets, as factuality is checking the precision of the system generated sequence. We mainly follow the instructions from BARTScore (Yuan et al., 2021), and use RScore for Rank19 dataset, and FScore for the other tasks, including Machine Translation and Data2Text tasks.

Generation to score Here, one uses the perplexity of a Seq2Seq model generating a sequence given another sequence as the evaluation score. Typical evaluators of this type include BARTScore (Yuan et al., 2021). If we use reference sequence \( X_2 \) as input and model’s output \( X_1 \) as target, the perplexity of \( X_1 \) can be used as the PScore (Figure 2 (d)). If \( X_1 \) is the input and \( X_2 \) is the target, the perplexity is the RScore. FScore can then be computed by PScore and RScore, which is same as computing F1. Thus, we fine-tune BART (Lewis et al., 2020a) with strictly paraphrased pairs from ParaTag and use it as the NLG evaluator.

5.3 Paraphrasing Generator

Given a paraphrasing dataset, one can train a generation model that produces a paraphrased version given an input sentence. This model can be used to augment training data of any NLP task. Thus, we fine-tune the BART-Large model (Lewis et al., 2020a) on a subset of ParaTag. Instead of selecting the strictly paraphrased sentence pairs, we set a threshold \( \theta = 0.05 \) and select sentence pairs whose non-paraphrase spans take at most \( \theta \) portion of the whole sentence pair. The resulting subset has 38,663 sentence pairs. Given the fine-grained annotation and diversity in the dataset, models trained on ParaTag become suitable for data augmentation (Feng et al., 2021). In Section 6, we show that a paraphrasing generator model trained on ParaTag is a better data augmenter than that trained on other public paraphrasing datasets.

6 Experiment

6.1 Dataset

ParaTag Table 2 shows the details of our collected dataset. There are two subsets in the training data: “Train1” consists of sentence pairs from the MRPC dataset and GPT-3 generation without model filtering, and it is used to train the paraphrasing detector. “Train2” set is to efficiently collect high-quality paraphrases, as described in section 4.3. This set is not used to train paraphrase
Table 2: Statistics of ParaTag dataset. #Pairs: number of sentence pairs. #Paraphrase: number of pairs where all the words are tagged as paraphrase. #Word Avg: average number of words in a sentence pair. #Words All: Total number of words. Para Tag Rate: percentage of words with paraphrase tags.

<table>
<thead>
<tr>
<th></th>
<th>Train1</th>
<th>Train2</th>
<th>Val</th>
<th>Test1</th>
<th>Test2</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Pairs</td>
<td>21,471</td>
<td>7,200</td>
<td>400</td>
<td>400</td>
<td>310</td>
</tr>
<tr>
<td>#Paraphrases</td>
<td>2,036</td>
<td>3,425</td>
<td>51</td>
<td>73</td>
<td>54</td>
</tr>
<tr>
<td>#Word Avg</td>
<td>39.0</td>
<td>26.5</td>
<td>31.38</td>
<td>31.25</td>
<td>31.35</td>
</tr>
<tr>
<td>#Words All</td>
<td>837,747</td>
<td>190,499</td>
<td>12,524</td>
<td>12,471</td>
<td>9,688</td>
</tr>
<tr>
<td>Para Tag Rate</td>
<td>0.688</td>
<td>0.897</td>
<td>0.669</td>
<td>0.733</td>
<td>0.744</td>
</tr>
</tbody>
</table>

Table 3: Paraphrasing detection experiment results on ParaTag. Test set is a combination of Test1 and Test2 in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Val</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>F1</td>
</tr>
<tr>
<td>BERT-base</td>
<td>0.8263</td>
<td>0.6871</td>
</tr>
<tr>
<td>BERT-large</td>
<td>0.8255</td>
<td>0.6822</td>
</tr>
<tr>
<td>RoBERTa-base</td>
<td>0.8299</td>
<td>0.6845</td>
</tr>
<tr>
<td>RoBERTa-large</td>
<td>0.8308</td>
<td>0.6876</td>
</tr>
<tr>
<td>DeBERTa-v3-base</td>
<td>0.8274</td>
<td>0.6803</td>
</tr>
<tr>
<td>DeBERTa-v3-large</td>
<td>0.8297</td>
<td>0.6885</td>
</tr>
</tbody>
</table>

Data Augmentation We evaluate the quality of data augmentor models on the GLUE benchmark (Wang et al., 2019), with a focus on RTE (Dagan et al., 2005), WNLI (Levesque et al., 2012), STSB (Cer et al., 2017), QNLI (Rajpurkar et al., 2016), QQP (Wang et al., 2017a), SST-2 (Socher et al., 2013), and MNLI (Williams et al., 2018) datasets.

6.2 Experiment results

Paraphrasing detection We train BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DeBERTa-v3 (He et al., 2021) with base (12 layers) and large (24 layers) structures on our ParaTag dataset for 1000 steps with batch size from [256, 512], learning rate [1e-5, 3e-5, 5e-5], and AdamW (Loshchilov and Hutter, 2019) optimizer. Our experiment results are shown in Table 3. Based on our experiment results, DeBERTa-large-v3 works the best on test set regarding both word-level metric Accuracy and F1 score. Thus, we will also use DeBERTa-large-v3 as NLG evaluator. Our code is based on token-classification from Huggingface Transformers.

NLG evaluator Our experiments on NLG evaluator are shown in Table 4,5,6 for Summarization, Data2Text, and Machine Translation respectively. We evaluate generation system in two directions: 1) Matching to score, where the baselines includes ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), BERTScore (Zhang et al., 2020), MoverScore (Zhao et al., 2019), and AdamW (Loshchilov and Hutter, 2019) optimizer. Our experiment results are shown in Table 4,5,6. These methods use either lexical features or embeddings to compute how well the system generated sequence and the reference are matched. Our paraphrasing detection model trained with ParaTag is under this direction. We use PScore for SummEval, Q-CNN, and Q-XSUM datasets, RScore for REALSumm,

---

7https://github.com/neulab/BARTScore

8https://github.com/huggingface/transformers/blob/main/examples/pytorch/token-classification/run_ner.py
Table 4: NLG Evaluator experiments on Summarization datasets.

<table>
<thead>
<tr>
<th>BAGEL</th>
<th>SFRES</th>
<th>SFHOT</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUGE-1</td>
<td>0.234</td>
<td>0.115</td>
<td>0.118</td>
</tr>
<tr>
<td>ROUGE-2</td>
<td>0.199</td>
<td>0.116</td>
<td>0.088</td>
</tr>
<tr>
<td>ROUGE-L</td>
<td>0.189</td>
<td>0.103</td>
<td>0.110</td>
</tr>
<tr>
<td>BERTScore</td>
<td>0.289</td>
<td>0.156</td>
<td>0.135</td>
</tr>
<tr>
<td>MoverScore</td>
<td>0.284</td>
<td>0.153</td>
<td>0.172</td>
</tr>
<tr>
<td>ParaTag</td>
<td>0.318</td>
<td>0.206</td>
<td>0.264</td>
</tr>
</tbody>
</table>

Table 5: NLG evaluator experiments on Data2Text and FScore for the others. Our method can significantly improve the baselines by 13% on average of 5 Summarization tasks, 6% on average of 3 Data2Text tasks, 1.4% on average of 4 languages in Machine Translation. Moreover, from Table 4, we can see that our method is quite good at evaluating the factuality and achieves the best performance over all baselines on SummEval FAC, Rank19, and Q-XSUM. In the other direction, 2) Generation to score, it makes use of the perplexity of generating one sequence from the other one as the evaluation score. We compare with the baselines PRISM (Thompson and Post, 2020) and BARTScore (Yuan et al., 2021). Our method use the same architecture as BARTScore. The only difference is that BARTScore is trained with Parabank but our method is trained with all the paraphrases from ParaTag for 3 epochs. Our method ParaTag-BARTScore can improve ParaBank-BARTScore by 3.8% on average of Summarization tasks, 2% on average of Data2Text tasks.

Paraphrasing generator We fine-tuned our paraphrasing model using BART-Large (Lewis et al., 2020b) on ParaTag. Given the paraphrase generator, we generate pseudo data on the training set of GLUE benchmark (Wang et al., 2019). For comparison, we use a paraphrase generator with BART-Large finetuned on QQP (Wang et al., 2017b), PAWS (Zhang et al., 2019) and MRPC (Dolan and Brockett, 2005).

We first compute the ROUGE-2 scores between the original sentences in GLUE and the paraphrased ones from the paraphrase generator. We find that our model finetuned on ParaTag get much lower ROUGE-2 score across GLUE datasets than our paraphraser baseline, as shown in table 7. It shows our method generates more diverse paraphrases than the model finetuned on other public paraphrase datasets.

Meanwhile, BERT-base (Devlin et al., 2019) trained with our data augmentation method can improve the baseline without data augmentation by 1.5% on average and improve the data augmentation baseline by 1% on average, as shown in table 8. These empirical results confirm that ParaTag...
Table 7: ROUGE-2 F scores between the original sentence and the paraphrased one for the public paraphrase model and our model finetuned on ParaTag. Lower ROUGE means that the model can generate more diverse paraphrases.

<table>
<thead>
<tr>
<th>Task</th>
<th>RTE</th>
<th>WNLI</th>
<th>STS-B</th>
<th>QNLI</th>
<th>QQP</th>
<th>SST-2</th>
<th>MNLI_m</th>
<th>MNLI_mm</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Augmentation</td>
<td>63.9</td>
<td>35.21</td>
<td>87.22</td>
<td>90.26</td>
<td>90.58</td>
<td>92.32</td>
<td>83.59</td>
<td>84.11</td>
<td>78.40</td>
</tr>
<tr>
<td>Paraphraser trained on QQP, PAWS &amp; MRPC</td>
<td><strong>64.98</strong></td>
<td>36.62</td>
<td>88.67</td>
<td>90.48</td>
<td>90.57</td>
<td>91.97</td>
<td>83.62</td>
<td>83.88</td>
<td>78.85</td>
</tr>
<tr>
<td>Paraphraser trained on ParaTag</td>
<td>62.09</td>
<td><strong>46.48</strong></td>
<td><strong>88.79</strong></td>
<td><strong>90.52</strong></td>
<td><strong>90.67</strong></td>
<td>92.2</td>
<td>84.07</td>
<td>84.30</td>
<td>79.89</td>
</tr>
</tbody>
</table>

Table 8: The performance on the GLUE dev set without data augmentation and with data augmentation from models trained on different paraphrasing datasets. STS-B evaluation metric is Pearson and the others are Accuracy.

is able to train a paraphrase generator with more diverse output but higher data augmentation quality at the same time.

7 Conclusion

We propose a novel dataset ParaTag for a more fine-grained and diverse paraphrase dataset. We set baselines with state-of-art pretrained language models for the ease of future comparison. In addition, we utilize ParaTag to train NLG evaluators that better correlate with human judgment. Finally, we are able to build a better data argumentation model from ParaTag for general NLU tasks.

8 Limitations

When labeling the data, the human agreement is 0.66 on test set and 0.6 on train set, which is not high enough. Due to the limitation of budget, we cannot hire more labelers to further improve the quality and collect more data. 90% of the training data is only annotated by a single labeler.

References


Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In Empirical Methods in Natural Language Processing (EMNLP).


### A Appendix

Table 9 shows the annotation examples.

<table>
<thead>
<tr>
<th>Tag Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noun phrase not paraphrased</strong></td>
<td>Sentence1: Hillary Clinton, meanwhile, continues to hold a steady lead in New Hampshire, the [Post-ABC News poll] found.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: Meanwhile, the New Hampshire survey showed Hillary Clinton [leading Barack Obama by 19 points].</td>
</tr>
<tr>
<td><strong>Negation and noun phrase not paraphrased</strong></td>
<td>Sentence1: [Chairman Bill Owens said] the decision [to split the jobs of chairman and chief executive] had [not] been taken to end a damaging feud with his co-founder and biggest shareholder, James Cayne.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: Co-founder and biggest shareholder James Cayne ended the feud by forcing this decision.</td>
</tr>
<tr>
<td><strong>Coreference not paraphrased</strong></td>
<td>Sentence1: [Stern] said that if the state doesn’t begin the process by spring, the most likely scenario is a court-ordered election of its own to review it.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: [He] said the state will likely have to institute a court-ordered election if no progress is made by the spring.</td>
</tr>
<tr>
<td><strong>Verb and number not paraphrased</strong></td>
<td>Sentence1: Yucaipa [owned] Dominick’s before selling the chain to Safeway in 1998 for $[2.5] billion.</td>
</tr>
<tr>
<td><strong>Clause not paraphrased</strong></td>
<td>Sentence1: [That compared with $35.18 million, or 24 cents per share,] in the year-ago period.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: [Earnings were affected by a non-recurring $8 million tax benefit] in the year-ago period.</td>
</tr>
<tr>
<td><strong>Paraphrase</strong></td>
<td>Sentence1: An Indianapolis Star report is included.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: Material from The Indianapolis Star supplemented this report.</td>
</tr>
<tr>
<td></td>
<td>Sentence1: Analysts at Merrill Lynch suggested drug makers’ earnings this quarter could grow 15 percent.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: Merrill Lynch’s analysts predicted that quarterly earnings for the nation’s pharmaceutical companies will rise 15 percent.</td>
</tr>
<tr>
<td></td>
<td>Sentence1: Amrozi accused his brother, whom he called &quot;the witness&quot;, of deliberately distorting his evidence.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: Referring to him as (only) &quot;the witness&quot;, Amrozi accused his brother of deliberately distorting his evidence.</td>
</tr>
<tr>
<td><strong>Whole pair is non-paraphrase</strong></td>
<td>Sentence1: It failed to deliver on a 15.3 percent rise in 2002, and most analysts expect another miss in 2003.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: [Earnings are expected to slip in 2003, as most analysts anticipate a year-on-year decline in operating profit.]</td>
</tr>
<tr>
<td></td>
<td>Sentence1: [That compared with $35.18 million, or 24 cents per share.] T</td>
</tr>
<tr>
<td><strong>Entity Coreference</strong></td>
<td>Sentence1: He told The Sun [newspaper] that [Mr. Hussein]’s daughters had British schools and hospitals in mind when they decided to ask for asylum.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: &quot;[Saddam]’s daughters had British schools and hospitals in mind when they decided to ask for asylum – [especially the schools]&quot;, he told The Sun.</td>
</tr>
<tr>
<td></td>
<td>Sentence1: A federal magistrate in Fort Lauderdale ordered [him] held without bail.</td>
</tr>
<tr>
<td></td>
<td>Sentence2: [Zuccarini] was ordered held without bail [Wednesday] by a federal judge in Fort Lauderdale, [Fla]</td>
</tr>
</tbody>
</table>

Table 9: Annotation Examples.