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

Shuohang Wang*, Ruochen Xu*, Yang Liu*, Chenguang Zhu, Michael Zeng
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]

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.

M-1	***However***, commercial use of the 2.6.0 kernel is still months off for ***most customers***.
M-2	Commercial releases of the 2.6 kernel by ***major Linux distributors*** still remain months away.
G-1	It ***covers some of*** the same ***ground*** as Spotlight but has nothing new to add.
G-2	It ***revisits*** the same ***topic*** as Spotlight, but does not say anything new.
G-3	Heavy rainstorms have caused floods in the area every year since 2003.
G-4	Since 2003, the area has been hit every year by heavy rainstorms and floods.

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.

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).

* Equal contribution.
[1]Data and code: https://github.com/microsoft/ParaTag
[2]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 7111 - 7122
December 7-11, 2022

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 dataset[4](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

[3]We will release our data, code, and model checkpoints after the anonymity period.

[4]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

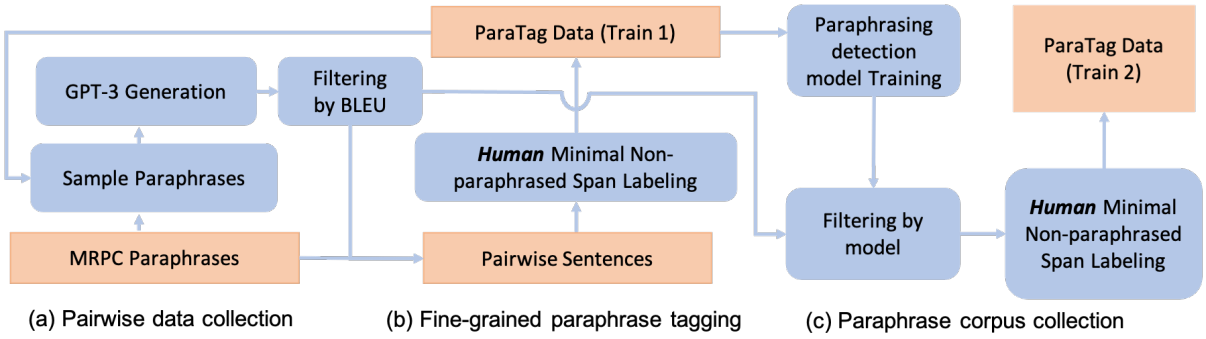

Figure 1: ParaTag annotation process. BLEU filtering: we kept sentence pair with BLEU score between 5 and 20.

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 $\mathcal{S}$ with a reference $\mathcal{R}$ and output a score representing how well $\mathcal{S}$ aligns with $\mathcal{R}$. 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. \n Paraphrase: Commercial releases of the 2.6 kernel by major Linux distributors still remain months away.\n"*. After GPT-3 generation, we only keep the decoded sentence pairs containing the pattern *"Premise: \n Paraphrase: \n"*. 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)[5] on

[5]Package: sklearn.metrics.cohen_kappa_score

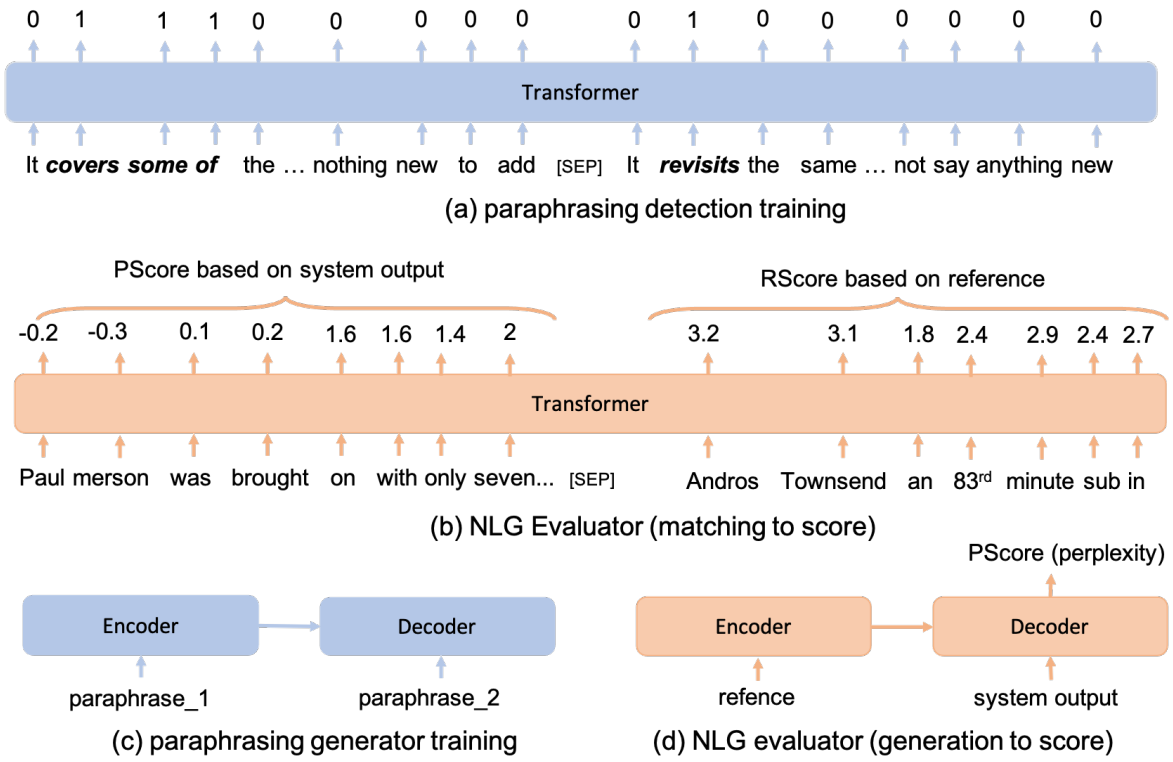

Figure 2: Model training with ParaTag and application to NLG evaluator.

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[6] 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.

[6]Package to compute F1 score: https://github.com/chakki-works/seqeval

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,

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

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 θ 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

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

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.

dection model. “Test1" and “Test2" are two sets labeled in two ways as described in section 4.2. We merge these two sets as the final test set.

Meta-Evaluation for language generation We are exploring NLG evaluator with ParaTag on the tasks of Summarization, Data2Text, and Machine Translation. We follow the datasets and evaluation metric from BARTScore[7]. For Summarization task, we work on SummEval (Fabbri et al., 2021), REALSumm(Bhandari et al., 2020), Rank19 (Falke et al., 2019), Q-CNN (Wang et al., 2020), Q-XSUM (Wang et al., 2020) datasets. The measure metric calculates the correlation between human scores and NLG evaluater scores by Spearman, Pearson, or accuracy. Factuality (FAC) is one of the popular meta-evaluation perspectives, which is adopted by 4 datasets. Factuality is to evaluate whether all the statements in the generated summary are factual. There are also several other meta-evaluation. Coherence (COH) and Informativeness (INFO) are to evaluate whether the generated summary has the same topic or shares the key ideas with the document. Fluency (FLU) is more on the evaluation of readability. Coverage (COV) evaluates the semantic coverage of reference sequence by generated summary. For Data2Text task, which generates textual sequence based on features in phrases, we work on the datasets of BAGEL (Mairesse et al., 2010), SFRES (Wen et al., 2015), SFHOT (Wen et al., 2015). The measurement of meta-evaluation is Spearman correlation. And the evaluation is conducted concerning the informativeness of the system generated sequence. For the Machine Translation task, we work on WMT19 metrics shared task (Ma et al., 2019). The measure metric is based on Kendall’s Tau correlation.

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

	Val		Test	
	Acc	F1	Acc	F1
BERT-base	0.8263	0.6871	0.8471	0.6853
BERT-large	0.8255	0.6822	0.8498	0.6745
RoBERTa-base	**0.8299**	0.6845	0.8574	0.6923
RoBERTa-large	0.8308	0.6876	0.8586	0.6955
DeBERTa-v3-base	0.8274	0.6803	0.8600	0.6998
DeBERTa-v3-large	0.8297	**0.6885**	**0.8639**	**0.7063**

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

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), STS-B (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.[8]

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). 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,

[8]https://github.com/huggingface/transformers/blob/main/examples/pytorch/token-classification/run_ner.py

Dataset	SummEval				REALSumm	Rank19	Q-CNN	Q-XSUM	
Perspective	COH	FAC	FLU	INFO	COV	FAC	FAC	FAC	Average
Measure	Spearman	Spearman	Spearman	Spearman	Spearman	Acc	Pearson	Pearson	
Matching to score									
ROUGE-1	0.167	0.160	0.115	0.326	0.498	0.568	0.338	-0.008	0.271
ROUGE-2	0.184	0.187	0.159	0.290	0.423	0.630	0.459	0.097	0.304
ROUGE-L	0.128	0.115	0.105	0.311	0.488	0.587	0.357	0.024	0.264
BERTScore	**0.284**	0.110	0.193	0.312	0.440	0.713	0.576	0.024	0.332
MoverScore	0.159	0.157	0.129	**0.318**	0.372	0.713	0.414	0.054	0.290
ParaTag	0.253	**0.472**	**0.357**	0.219	**0.504**	**0.853**	**0.694**	**0.342**	**0.462**
Generation to score									
PRISM	0.249	0.345	0.254	0.212	0.411	0.780	0.479	0.025	0.344
ParaBank-BARTScore	0.424	**0.401**	**0.378**	0.313	0.471	0.788	0.680	0.074	0.441
ParaTag-BARTScore	**0.472**	0.396	0.369	**0.367**	**0.471**	**0.845**	**0.727**	**0.188**	**0.479**

Table 4: NLG Evaluator experiments on Summarization datasets.

	BAGEL	SFRES	SFHOT	Average
Matching to score				
ROUGE-1	0.234	0.115	0.118	0.156
ROUGE-2	0.199	0.116	0.088	0.134
ROUGE-L	0.189	0.103	0.110	0.134
BERTScore	0.289	0.156	0.135	0.193
MoverScore	0.284	0.153	0.172	0.203
ParaTag	**0.318**	**0.206**	**0.264**	**0.263**
Generation to score				
PRISM	0.305	0.155	0.196	0.219
ParaBank-BARTScore	0.330	0.185	0.211	0.242
ParaTag-BARTScore	**0.360**	**0.209**	**0.217**	**0.262**

Table 5: NLG evaluator experiments on Data2Text

	gu-en	kk-en	lt-en	ru-en	Average
Matching to score					
BLEURT	0.313	0.372	0.388	0.220	0.323
COMET	0.316	0.378	0.405	0.226	0.331
BLEU	0.194	0.276	0.249	0.115	0.209
BERTScore	0.292	0.351	0.381	0.221	0.311
ParaTag	**0.305**	**0.380**	**0.382**	**0.231**	**0.325**
Generation to score					
ParaBank-BARTScore	**0.316**	**0.378**	0.386	0.219	**0.325**
ParaTag-BARTScore	0.306	0.356	**0.388**	**0.225**	0.319

Table 6: NLG evaluator on Machine Translation

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) [9] 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

[9] We initialize model from https://huggingface.co/eugenesiow/bart-paraphrase

Argument Model	cola-s1	mnli-s1	mnli-s2	mrpc-s1	mrpc-2	qnli-s1	qnli-s2	qqp-s1
Baseline Paraphraser	85.33	90.14	91.43	87.73	93.19	78.73	91.95	66.34
Our Model	60.12	65.41	73.65	77.64	83.73	61.57	76.92	62.15
	qqp-s2	rte-s1	rte-s2	sst2-s1	stsb-s1	stsb-s2	wnli-s1	wnli-s2
Baseline Paraphraser	69.19	81.09	85.52	84.26	81.83	82.33	94.57	91.3
Our Model	65.1	69.39	67.49	72.98	63.09	63.74	70.46	69.83

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.

Task	RTE	WNLI	STS-B	QNLI	QQP	SST-2	MNLI_m	MNLI_mm	Avg.
No Augmentation	63.9	35.21	87.22	90.26	90.58	**92.32**	83.59	84.11	78.40
Paraphraser trained on									
- QQP, PAWS & MRPC	**64.98**	36.62	88.67	90.48	90.57	91.97	83.62	83.88	78.85
- ParaTag	62.09	**46.48**	**88.79**	**90.52**	**90.67**	92.2	**84.07**	**84.30**	**79.89**

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

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.

Manik Bhandari, Pranav Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in neural information processing systems (NeurIPS)*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. In *International Workshop on Semantic Evaluations*.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Association for Computational Linguistics (ACL)*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*. Springer.

Dipanjan Das and Noah A Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP)*.

Qingxiu Dong, Xiaojun Wan, and Yue Cao. 2021. Parasci: A large scientific paraphrase dataset for longer paraphrase generation. In *European Chapter of the Association for Computational Linguistics (EACL)*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics (TACL)*.

Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Association for Computational Linguistics (ACL)*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Association for Computational Linguistics (ACL)*.

Darina Gold, Venelin Kovatchev, and Torsten Zesch. 2019. Annotating and analyzing the interactions between meaning relations. In *Proceedings of the 13th Linguistic Annotation Workshop*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

J Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Venelin Kovatchev, M Antònia Martí, and Maria Salamó. 2018. Etpc-a paraphrase identification corpus annotated with extended paraphrase typology and negation. In *International Conference on Language Resources and Evaluation (LREC 2018)*.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Conference on the Principles of Knowledge Representation and Reasoning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020a. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Association for Computational Linguistics (ACL)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Qingsong Ma, Johnny Tian-Zheng Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. Association for Computational Linguistics.

François Mairesse, Milica Gasic, Filip Jurcicek, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Association for Computational Linguistics (ACL)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Association for Computational Linguistics (ACL)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv preprint arXiv:2009.09025*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Association for Computational Linguistics (ACL)*.

Nigel C Smeeton. 1985. Early history of the kappa statistic. *Biometrics*.

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)*.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical bert models for sequence labeling. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Association for Computational Linguistics (ACL)*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations (ICLR)*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017a. Bilateral multi-perspective matching for natural language sentences. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017b. Bilateral multi-perspective matching for natural language sentences. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Empirical Methods in Natural Language Processing (EMNLP)*.

John Wieting and Kevin Gimpel. 2018. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Association for Computational Linguistics (ACL)*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational (TACL)*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Empirical Methods in Natural Language Processing (EMNLP)*.

A Appendix

Table 9 shows the annotation examples.

Tag Type	Example
Noun phrase not paraphrased	Sentence1: Hillary Clinton , meanwhile , continues to hold a steady lead in New Hampshire , the [Post-ABC News poll] found . Sentence2: Meanwhile , the New Hampshire survey showed Hillary Clinton [leading Barack Obama by 19 points] .
Negation and noun phrase not paraphrased	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 . Sentence2: Co-founder and biggest shareholder James Cayne ended the feud by forcing this decision .
Coreference not paraphrased	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 . Sentence2: [He] said the state will likely have to institute a court-ordered election if no progress is made by the spring .
Verb and number not paraphrased	Sentence1: Yucaipa [owned] Dominick 's before selling the chain to Safeway in 1998 for $ [2.5] billion . Sentence2: Yucaipa [bought] Dominick 's in [1995] for $ [693 million] and sold it to Safeway for $ [1.8] billion in 1998 .
Clause not paraphrased	Sentence1: [That compared with $ 35.18 million , or 24 cents per share] , in the year-ago period . Sentence2: [Earnings were affected by a non-recurring $ 8 million tax benefit] in the year-ago period .
Paraphrase	Sentence1: An Indianapolis Star report is included . Sentence2: Material from The Indianapolis Star supplemented this report .
	Sentence1: Analysts at Merrill Lynch suggested drug makers ' earnings this quarter could grow 15 percent . Sentence2: Merrill Lynch 's analysts predicted that quarterly earnings for the nation 's pharmaceutical companies will rise 15 percent .
	Sentence1: Amrozi accused his brother , whom he called " the witness " , of deliberately distorting his evidence . Sentence2: Referring to him as (only) " the witness " , Amrozi accused his brother of deliberately distorting his evidence .
Whole pair is non-paraphrase	Sentence1: [It failed to deliver on a 15.3 percent rise in 2002 , and most analysts expect another miss in 2003 .] Sentence2: [Earnings are expected to slip in 2003 , as most analysts anticipate a year-on-year decline in operating profit .]
	Sentence1: [That compared with $ 35.18 million , or 24 cents per share.] Sentence2: [Earnings were affected by a non-recurring $ 8 million tax benefit.]
Entity Coreference	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 . Sentence2: " [Saddam] 's daughters had British schools and hospitals in mind when they decided to ask for asylum – [especially the schools]", he told The Sun .
	Sentence1: A federal magistrate in Fort Lauderdale ordered [him] held without bail . Sentence2: [Zuccarini] was ordered held without bail [Wednesday] by a federal judge in Fort Lauderdale , [Fla]

Table 9: Annotation Examples.