TD-ConE: An Information-Theoretic Approach to Assessing Parallel Text Generation Data

Anonymous ACL submission

Abstract

Existing data assessment methods are mainly for classification-based datasets and limited for use in natural language generation (NLG) datasets. In this work, we focus on parallel NLG datasets and address this problem through an information-theoretic approach, TD-ConE, to assess *data uncertainty* using input-output sequence mappings. Our experiments on text style transfer datasets demonstrate that the proposed simple method leads to better measurement of data uncertainty compared to some complicated alternatives and demonstrates a high correlation with downstream model performance. As an extension of TD-ConE, we introduce TD-ConE $_{\text{REL}}$ to compute the relative uncertainty between two datasets. Our experiments with paraphrase generation datasets demonstrate that selecting data with lower TD-ConE$_{\text{REL}}$ scores leads to better model performance and decreased validation perplexity.

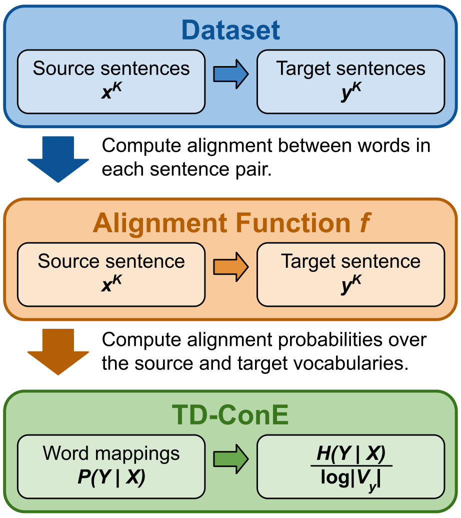

Figure 1: Procedure for computing TD-ConE. Given a dataset, define an alignment function to obtain $P(Y \mid X)$ over the source and target vocabularies for use with computation of TD-ConE.

1 Introduction

Assessing and understanding data in natural language processing (NLP) benefits research on learnability (Swayamdipta et al., 2020), reproducability (Beck et al., 2020), and generalizability (Bender and Friedman, 2018). Although existing methods show promising results from data assessment in detecting annotation artifacts (Gururangan et al., 2018; Poliak et al., 2018) and selecting training examples (Moore and Lewis, 2010; Ruder and Plank, 2017; Zhang and Plank, 2021), most are limited to certain types of NLP tasks and cannot directly apply to natural language generation.

There are three notable limitations of existing methods when considering NLG: application constraints from output formats, high computational cost (which covers model-dependent methods) and no corpus-level evaluation (cannot handle the cases with large-scale datasets). First, many existing methods are constrained to tasks with output labels, which enables computations from training dynamics such as model confidence or variability of predictions (Zhang and Plank, 2021; Swayamdipta et al., 2020). This leaves few existing methods that are applicable to sequential outputs as in text generation. Compounding on this limitation is the high computational cost of strictly model-dependent methods. At scale, NLG datasets can contain millions of training examples (e.g. 2.8 million candidate pairs in Twitter URL dataset (Lan et al., 2017)) with increasingly large parameter counts for state-of-the-art models (e.g. 1.5 billion parameters in GPT-2 (Radford et al., 2019)). Many previous methods that incorporate models, however, make instance-level evaluations and require model retraining, such as the Data Shapley (Ghorbani and Zou, 2019) (time complexity $O(2^N)$ for N data points). Finally, methods that incorporate learned parameters have a similar limitation due to multiple model initializations being computationally prohibitive, yet random initializations may produce undesirable variability in results.

To address these limitations, we propose a simple method to estimating the conditional probability of outputs given inputs, and measure data uncertainty using conditional entropy (Shannon, 1948), shown in Figure 1. This approach is further extended to measure the uncertainty of one dataset given another, using relative entropy (Kullback and Leibler, 1951). Specifically, our contributions are: 1) taking an information-theoretic perspective to measure data uncertainty in parallel NLG datasets with an **entropy-based metric** TD-CoNE and its extended version TD-CoNE$_{\text{REL}}$; 2) proposing simple yet effective **word alignment methods** without any learned parameters for computing TD-CoNE and TD-CoNE$_{\text{REL}}$; 3) with English text style transfer and paraphrase generation datasets, demonstrating the utility of using the proposed data uncertainty measures TD-CoNE and TD-CoNE$_{\text{REL}}$ as indicators of downstream model performance and validation perplexity, and as aids for selecting data or making comparisons between datasets.

2 TD-CoNE: Dataset-Level Uncertainty

Entropy in information theory offers a theoretical basis for measuring the uncertainty of a random variable (Shannon, 1948). In this work, we propose to use the definition of entropy for measuring the uncertainty of a dataset. Assume we have the conditional probability $P(Y \mid X)$ estimated from the dataset (the estimation is not trivial and will be detailed in section 3), then the conditional entropy $H(Y \mid X)$ measures the uncertainty of Y given X. Let X represent a word in the input vocabulary $\mathcal{V}_x$ and Y represent a word in the output vocabulary $\mathcal{V}_y$, then this conditional entropy provides us a starting point of defining our task-specific data uncertainty.

Definition 1 (TD-CoNE)**.** *The **T**ask-**D**ataset **Con**ditional **E**ntropy (TD-CoNE) is defined as*

$$\text{TD-CoNE}(Y \mid X) = \frac{H(Y \mid X)}{\log |\mathcal{V}_y|} \quad (1)$$

where $H(Y \mid X)$ is the conditional entropy, and $|\mathcal{V}_y|$ is the size of the output vocabulary.

The denominator $|\mathcal{V}_y|$ normalizes the value of $H(Y \mid X)$ and guarantees TD-CoNE$(Y \mid X)$ always bounded between 0 and 1. Specifically, we have $0 \leq H(Y \mid X) \leq H(Y) \leq \log |\mathcal{V}_y|$ (Shannon, 1948). Additionally, we generally have TD-CoNE$(Y \mid X) \neq$ TD-CoNE$(X \mid Y)$, because of $P(Y \mid X) \neq P(X \mid Y)$. This is consistent with the task setup in text generation, since mapping from X to Y should be a different task as mapping Y to X (e.g., in text style transfer).Therefore, our definition in Equation 1 is task-specific.

2.1 Challenges of Estimating $H(Y \mid X)$

$H(Y \mid X)$ is dependent on the joint probability $P(X, Y)$, which can be further decomposed as $P(X) \cdot P(Y \mid X)$. While $P(X)$ is essentially the unigram distribution estimated from the input sentences, we need a method to estimate the conditional probability $P(Y \mid X)$ from the data. For this, we can consider parallel NLG datasets analogously to monolingual translation and can utilize word alignments to identify mappings and estimate $P(Y \mid X)$ over a dataset (Wubben et al., 2010).

The estimation of $P(Y \mid X)$ with alignments poses several challenges: 1) word alignments that require identifying which word (or words) in $\boldsymbol{x}$ map to a given word in $\boldsymbol{y}$ are not directly observable in the data; 2) to accurately apply word alignments to estimate $P(Y \mid X)$ for measuring data uncertainty, we need to minimize uncertainty arising from the alignment method itself.

Many existing word alignment methods treat alignment as a latent factor to be learned by a model (Brown et al., 1993), which could introduce a secondary source of uncertainty. Specifically, prediction uncertainty $P(Y \mid X)$ usually contains two sources of uncertainty: data uncertainty and model uncertainty. *Model uncertainty* is dependent on learnable parameters and reducible with additional data or a more sophisticated modeling approach, whereas *data uncertainty* is inherent data noise that cannot be reduced through a better model (Gal, 2016). We need to reduce the model uncertainty as much as we can, so the estimated uncertainty will be primarily data uncertainty. For this, we propose a simple word alignment method that uses static embeddings and no learnable parameters, described in the next section.

3 Static Word Alignments

Let $\boldsymbol{x} = \{x_1, \ldots, x_m\}$ represent one input sentence with m words and $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ represent the corresponding output sentence with n words. To minimize model uncertainty through minimal learnable parameters, we assume that all $\{x_i\}_{i=1}^{m}$ in the same sentence are independent from each other. The same assumption also applies to the words in the output sentence $\{y_j\}_{j=1}^{n}$. Although this ignores the linguistic dependency in texts, it simplifies the probabilistic modeling and minimizes the uncertainty of learned dependencies, offering a good trade-off between model complexity and the empirical performance of TD-CONE. We demonstrate this advantage empirically in comparisons with existing statistical and transformer-based alignment methods in section 4.2. With this assumption, the only dependency we consider in the rest of this section is the dependency between input words $\{x_i\}_{i=1}^{m}$ and output words $\{y_j\}_{j=1}^{n}$.

Consider a set of sentence pairs for text generation as $\mathcal{D} = \{(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)})\}_{k=1}^{K}$, where K is the total number of examples. With the dataset $\mathcal{D}$, we can define $\mathcal{V}_x$ as the input vocabulary constructed from $\{\boldsymbol{x}^{(k)}\}$ and $\mathcal{V}_y$ as the output vocabulary constructed from $\{\boldsymbol{y}^{(k)}\}$. Our problem setup is therefore to estimate the conditional probability $P(Y \mid X)$ given the dataset $\mathcal{D}$, where $X \in \mathcal{V}_x$ and $Y \in \mathcal{V}_y$.

For a given dataset, the challenge of estimating $P(Y \mid X)$ for a specific output word $y_j^{(k)}$ is to identify which word (or words) in $\boldsymbol{x}^{(k)}$ "generate" (i.e. are aligned with) $y_j^{(k)}$. Essentially, the estimation relies on the alignment between input and output words, where an alignment between two words indicates a conditional dependency.

The proposed Algorithm 1 employs an alignment matrix $\boldsymbol{M} \in \mathbb{R}^{|\mathcal{V}_X| \times |\mathcal{V}_Y|}$ to record the alignment counts based on $\mathcal{D}$. The algorithm essentially makes one-to-one mappings where possible, distributes probabilities over potential alignments if one-to-one mappings cannot be made (either uniform or using cosine similarities with *static* embeddings), and utilizes alignments to a special NULL token when either the input is a subset of the output or vice versa. Once $\boldsymbol{M}$ has been estimated over the entire dataset D, $P(Y|X = w)$ is obtained by normalizing the corresponding row in $\boldsymbol{M}$.[1]

We describe a deterministic version of the alignment algorithm using uniform probability alignments in Appendix C, which also had good preliminary results. [2] In our primary experiments, we opted to use static GloVe word-embeddings (Pennington et al., 2014) to compute the alignment probability distributions. Although this introduces learned embeddings, as the embeddings are neither context-dependent nor trained on each individual dataset, we maintain limited learned parameters

[1] A detailed description can be found in Appendix D

[2] Code for uniform and static alignments will be released.

Algorithm 1 Calculating the alignment matrix with one pair of sentences

Input: a sentence pair $\boldsymbol{x}$ and $\boldsymbol{y}$, alignment matrix $\boldsymbol{M}$
Output: the updated alignment matrix $\boldsymbol{M}$
for word $w \in \boldsymbol{x}$ **do**
  if $w \in \boldsymbol{x} \cap \boldsymbol{y}$ **then** $\boldsymbol{M}(w, w) \leftarrow \boldsymbol{M}(w, w) + 1$
  if $w \notin \boldsymbol{y} \backslash \boldsymbol{x}$ **then**
    if $|\boldsymbol{y} \backslash \boldsymbol{x}| = 0$ **then**
      $\boldsymbol{M}(w, \text{NULL}) \leftarrow \boldsymbol{M}(w, \text{NULL}) + 1$
    else
      for $w' \in (\boldsymbol{y} \backslash \boldsymbol{x})$ **do**
        $\text{SCORE} = (w, w' \in \text{EMBEDS})\ ?\ \frac{\boldsymbol{w}^{\top}\boldsymbol{w}'}{\|\boldsymbol{w}\| \cdot \|\boldsymbol{w}'\|} : \frac{1}{|\boldsymbol{y} \backslash \boldsymbol{x}|}$
        $\boldsymbol{M}(w, w') \leftarrow \boldsymbol{M}(w, w') + \frac{\text{SCORE}}{\sum_{i=0}^{N=|\boldsymbol{y}\backslash\boldsymbol{x}|} \text{SCORE}^{w' i}}$
if $\boldsymbol{x} \subset \boldsymbol{y}$ **then**
  for word $w' \in \boldsymbol{y} \backslash \boldsymbol{x}$ **do** $\boldsymbol{M}(\text{NULL}, w') \leftarrow \boldsymbol{M}(\text{NULL}, w') + \frac{1}{|\boldsymbol{y} \backslash \boldsymbol{x}|}$

and ensure consistent results across datasets.

4 TD-CONE Experiments

As uncertainty corresponds with available information, we expect that too much or too little uncertainty is not ideal for representing task information: if data uncertainty is too low a dataset may have a restricted or limited representation of the underlying task, and if data uncertainty is too high a dataset may contain a level of noise that is not conducive to learning task-relevant information. To evaluate TD-CONE and test this hypothesis, we compute TD-CONE across datasets representing the same general task and evaluate correlations and observed patterns with downstream model performance.

Our task selection criteria included included tasks with: 1) parallel datasets available with one-to-one input-output sentence pairs, and 2) benchmarked datasets with standard data splits. Text style transfer fit this criteria and enabled us to test TD-CONE across a diverse set of datasets in terms of sub-tasks (style), sizes, and creation methods. We baseline our method's efficacy for data uncertainty measurement by evaluating correlation with model performance against TD-CONE computed with existing word alignment methods. Notably, there are several distinctions between the intended use of TD-CoNE vs. existing methods that evaluate text using concepts related to uncertainty, such as diversity, that negate direct comparison: 1) assessing datasets **prior** to training vs. active learning or evaluating *generated* text, 2) level of measurement (corpus-level vs. instance-level), and 3) use on input-output pairs vs. reference-generation pairs (Alihosseini et al., 2019; Zhang et al., 2018).

4.1 Experiment setup

Datasets. We select 6 English datasets representing 8 unique attribute-based text style transfer tasks: **Fluency** (disfluent to fluent) (Wang et al., 2020), **GYAFC-EM** and **GYAFC-FR** (informal to formal) (Rao and Tetreault, 2018), **Biased-word** (subjective to neutral) (Pryzant et al., 2020), **Captions (Flickr)** (humorous to romantic, romatic to humorous) (Gan et al., 2017), and **Shakespeare** (Shakespearean to modern English, modern English to Shakespearean) (Xu et al., 2012). For text style datasets in which stylistic transfer has been previously benchmarked in both directions, we report results for both directions of transfer. Detailed selection criteria, descriptions, and statistics can be found in Appendix A.

Generation models. We use five models with different neural architectures of varying complexity: SimpleCopy (directly copy input as output; baseline scores for no learned stylistic information), Neural MT (NMT) (Bahdanau et al., 2014), CopyNMT (See et al., 2017), BART (Lewis et al., 2020), and GPT-2 (Radford et al., 2019; Wang et al., 2019). Details can be found in Appendix E.

Evaluation metrics. To report model performance, we report BLEU (Papineni et al., 2002) using the implementation from Koehn et al. (2007) as an measure of content preservation and prediction accuracy on the stylistic attribute as an indicator of transfer intensity. We report BLEU as all datasets in use have been benchmarked with BLEU, enabling us to ensure our model performance aligns with the existing literature and thus ensuring internal validity for reporting correlations. Prediction accuracy is computed using fastText classifiers (Joulin et al., 2017) in line with recent style transfer research (Dai et al., 2019; Subramanian et al., 2018; Sudhakar et al., 2019).

Competitive alignment methods. As described in section 3, in addition to the proposed alignment method for estimating $P(Y \mid X)$, there are other options available from statistical machine translation. To demonstrate the competitiveness of the proposed method, we compare against IBM Models 1, 2, and 3 using the GIZA++ implementations (Och and Ney, 2003) and the recently proposed BERT-based SimAlign (Jalili Sabet et al., 2020). For SimAlign, we instantiate the model using Huggingface's implementation of BERT-base-uncased (Devlin et al., 2018) with argmax matching.

4.2 Results

TD-CoNE accurately measures data uncertainty. TD-CoNE scores across dataset splits are reported in Figure 2 (and shown numerically in Table 6 found in Appendix A) and model performance is reported in Table 1 and Table 2. TD-CoNE and BLEU scores for all model architectures have a *negative* correlation, indicating higher data uncertainty (more uncertain sequence mappings) results in lower content preservation.[3] Further, TD-CoNE accurately captures data uncertainty in terms of input-output mappings across all datasets rather than simply being a reflection of the target class entropy. The largest difference in target class normalized entropy (reported in Table 7 in Appendix B) across datasets is 0.0693, whereas the largest difference in TD-CoNE across datasets is 0.3928. We attribute this to the normalization in TD-CoNE. This aligns with the expectation that target classes all represented in the same language should have similar normalized entropies (Shannon, 1948), and supports the finding that the wide range of TD-CoNE scores indicates that TD-CoNE accurately measures the data uncertainty as a reflection of cross-class mapping complexity.

Further, the style transfer accuracies reported in Table 2 suggest that there is likely an optimal uncertainty range in terms of task representation (TD-CoNE between 0.22 and 0.28 in our experiments, but this may be task-dependent). When TD-CoNE scores are above this range (Captions datasets), the noise level in the dataset precludes the ability of the model to learn accurate, grammatical mappings as evidenced by low BLEU scores. Instead, via qualitative analysis of the outputs we found that the models revert to generating repetitive yet salient style words, evidenced by the high style transfer accuracies. However, when TD-CoNE scores are below the ideal range (Bias and Fluency

[3]Correlations are reported alongside other alignment methods in section 4.2.

	Captions		Shakespeare		GYAFC-FR	GYAFC-EM	Biased	Fluency
Methods	Rom→Fun	Fun→Rom	Mod→Shake	Shake→Mod	Inf→Form	Inf→Form	Subj→Neut	Disf→Flt
SimpleCopy	8.03	8.07	21.66	21.58	53.75	52.69	90.27	90.53
NeuralMT	2.85	2.99	13.12	12.55	58.89	47.80	74.64	92.28
CopyNMT	2.75	3.06	15.88	14.32	62.72	55.33	91.41	95.27
BART	3.63	4.46	21.01	21.58	66.73	65.42	90.86	91.33
GPT-2	**8.14**	**8.30**	**23.26**	**25.34**	**71.44**	**67.32**	**93.73**	**96.59**

Table 1: Test set BLEU scores for generation models.

	Captions		Shakespeare		GYAFC-FR	GYAFC-EM	Biased	Fluency
Methods	Rom→Fun	Fun→Rom	Mod→Shake	Shake→Mod	Inf→Form	Inf→Form	Subj→Neut	Disf→Flt
SimpleCopy	29.20	28.40	20.04	14.77	18.02	17.16	33.50	27.42
NeuralMT	86.80	84.40	**78.92**	**80.78**	**82.06**	**84.25**	**72.30**	35.72
CopyNMT	86.00	72.40	71.34	70.93	79.58	74.86	70.10	35.63
BART	**90.00**	**94.80**	63.34	75.44	80.78	80.15	56.90	**53.84**
GPT-2	86.00	64.00	57.87	77.15	81.23	83.90	65.40	36.28

Table 2: Test set accuracy scores for generation models.

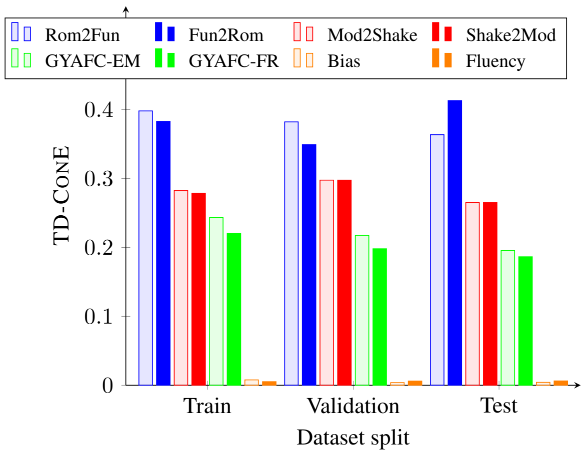

Figure 2: TD-CONE scores for each text style transfer task. Scores are also shown in Table 6 in Appendix B.

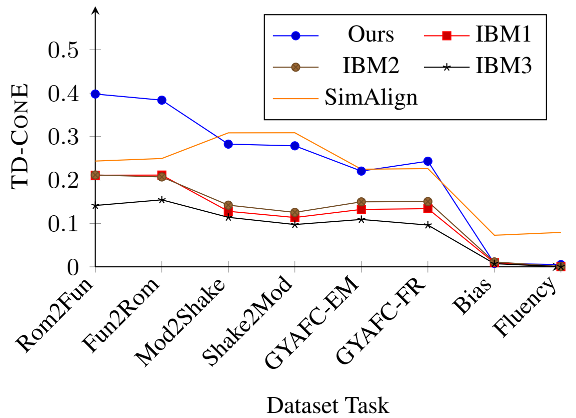

Figure 3: Comparison of TD-CONE using our alignment method and baseline methods. Datasets are sorted by ascending BLEU scores (ideal scores would be monotonically decreasing): our method outperforms existing methods (correlations reported in subsection 4.2).

datasets), the models learn to copy content information between classes, yet we see decreases in style transfer accuracy. We attribute this to a constrained representation of the task in the data.

Static word alignments outperform learned word alignments when estimating data uncertainty. We report TD-CONE computed with our proposed word alignment method, statistical IBM Models 1-3 using GIZA++ (Och and Ney, 2003), and BERT-based SimAlign (Jalili Sabet et al., 2020) in Figure 3. Our alignment method has an average correlation of -0.94 with BLEU scores across models, compared to $-0.87, -0.86, -0.89, -0.85$ for IBM 1 - 3 and SimAlign, respectively. We attribute this to our method better capturing data uncertainty by minimizing uncertainty attributable to the alignment model. In fact, correlation was lowest with SimAlign which used BERT contextual embeddings.

We also note several advantages of our algorithm due to its design for a monolingual setting: 1) our method leverages the ability to accurately assign one-to-one mappings for identical word pairs, which is ideal for measuring uncertainty; 2) our method utilizes distributed probabilities over $\boldsymbol{y} \backslash \boldsymbol{x}$ for each w when the symmetric difference $\boldsymbol{x} \Delta \boldsymbol{y} \neq \emptyset$. With static monolingual embeddings, we can utilize cosine similarities for this procedure, yet we have similarly good performance with the uniform distribution as presented in the Appendix; 3) while the typical usage of the NULL token in bilingual translation settings captures important structural dependencies across different languages, our usage is strictly designed to accurately estimate $P(X)$ and ensure dependency between input and output. Specifically, we use the NULL token in two scenarios: $\boldsymbol{y} \subset \boldsymbol{x}$ and $\boldsymbol{x} \subset \boldsymbol{y}$. If $\boldsymbol{y} \subset \boldsymbol{x}$ we increment the target NULL by 1 to ensure accurate estimation of $P(X)$, and if $\boldsymbol{x} \subset \boldsymbol{y}$ we increment the source NULL uniformly over $\boldsymbol{y} \backslash \boldsymbol{x}$ to ensure the dependency between input and output. In aggregate, these features tailor our method specifically for the task of estimating data uncertainty, as reflected in the experimental results.

5 TD-CONE$_{\text{REL}}$: Relative Uncertainty

While TD-CONE accurately measures the data uncertainty of a single dataset, with the estimation of $(P(Y \mid X)$ enabled using Algorithm 1, we can extend our methods to estimate the relative uncertainty of one dataset given another dataset. In a standard NLG setup, high validation set accuracy after training is desirable as it indicates generalization power to unseen data. However, there is the open question of how to select the optimal training set for a given validation set. Further, as it is standard practice to select the model with the highest validation perplexity, we hypothesize there is a relationship between relative data uncertainty of a validation set and downstream model validation perplexity (i.e. exponentiation of the entropy). Motivated by these questions, we can utilize Algorithm 1 to compute the conditional relative entropy (i.e Kullback–Leibler divergence) between two distributions, formally defined as follows:

Definition 2 (TD-CONE$_{\text{REL}}$)**.** *Consider $P(Y \mid X)$ and $Q(Y \mid X)$ to be two probability distributions on the same sample space $(X, Y) \in \mathcal{V}_x \times \mathcal{V}_y$. The* TD-CONE$_{\text{REL}}$ *or "**T**ask-**D**ataset **Con**ditional **E**ntropy: **Rel**ative Entropy" can be defined as the normalized conditional relative entropy between P and Q*

$$\text{TD-CONE}_{\text{RELATIVE}} = \frac{KL(P(Y \mid X) \| Q(Y \mid X))}{KL(P(Y \mid X) \| U(Y \mid X))} \tag{2}$$

where $KL(P(Y \mid X) \| Q(Y \mid X)) = \sum_{X,Y} P(X, Y) \log \frac{P(Y|X)}{Q(Y|X)}$ and $U(Y \mid X) = \frac{1}{|\mathcal{V}_y|}$ is the uniform distribution defined on the output vocabulary $\mathcal{V}_y$.

Due to the non-negative property of relative entropy, we have TD-CONE$_{\text{REL}} \geq 0$. In addition, since $U(Y \mid X)$ is a uniform distribution and therefore $KL(P \| Q) \leq \text{KL}(P \| U)$ always holds, we have $0 \leq$ TD-CONE$_{\text{REL}} \leq 1$. Given two datasets $\mathcal{D}_p$ and $\mathcal{D}_q$, $P(Y \mid X)$ and $Q(Y \mid X)$ can be estimated using the same algorithm proposed in section 3, enabling computation of TD-CONE$_{\text{REL}}$ prior to any model training.

6 TD-CONE$_{\text{REL}}$ Experiments

We expect that lower TD-CONE$_{\text{REL}}$ of a validation set given a training set (less uncertain validation set relative to a training set) will lead to better model performance in terms of model perplexity and automatic metrics on generated outputs. Our selection criteria for NLG tasks to evaluate TD-CONE$_{\text{REL}}$ included tasks which had: 1) parallel datasets available with one-to-one input-output sentence pairs, and 2) benchmarked datasets that lack standard data splits. Paraphrase generation fits these criteria and is advantageous to test the efficacy of TD-CONE$_{\text{REL}}$ for data split selection and comparison as: 1) existing literature has created purposefully difficult splits based on classification confidence thresholds (Li et al., 2018b) and 2) there are a wide range of reported metrics, limiting direct comparisons across studies (Du and Ji, 2019).

6.1 Experiment setup

Datasets. We use the **Quora Question Pairs** [4] and **Twitter URL** datasets (Lan et al., 2017) for paraphrase generation as 1) both are frequently used to evaluate paraphrase generation models, and 2) both have wide ranges of reported baseline model performance across studies (Li et al., 2018b; Du and Ji, 2019). Twitter URL contains both human (51k) and classifier (2.8 million) labeled sentence pairs. Quora Question Pairs contains 404k question pairs with binary labels indicating whether a pair are paraphrases. Detailed descriptions and usage can be found in Appendix A.

Models and metrics. Using the same implementations as subsection 4.1, we train GPT-2, NMT, and CopyNMT for paraphrase generation. In addition to TD-CONE$_{\text{REL}}$, report TD-CONE on each training set and validation perplexity and BLEU for model performance.

6.2 Methods

On Twitter URL. We manipulate selection thresholds (not frequently reported in existing work) and

[4] https://www.kaggle.com/c/quora-question-pairs

construct six training sets sampled from the automatically labeled candidate pairs meeting the respective probability thresholds: 0.4, 0.5, 0.6, 0.7, 0.75, 0.8. We follow the setup of Li et al. (2018b) and use 110k/1k/5k train/validation/test splits with validation and test examples sampled from the manually labeled examples. Validation and test sets are held constant across training thresholds. In line with standard practice, best models are selected as indicated by validation perplexity. By performing these manipulations, we aim to identify the impact and limitations of classifier scores for optimal training set selection. Additionally, as most datasets do not have classifier confidence scores readily available, we aim to identify whether TD-ConE$_{\text{REL}}$ displays a relationship with selection threshold or model performance.

On Quora Question Pairs. We use the combination of TD-ConE and TD-ConE$_{\text{REL}}$ to test training set selection efficacy using a 35k/1k/5k data split the Quora Question Pairs dataset. We experiment with five different selection methods: [1] randomly sampled from all potential paraphrases, [2] lowest randomly sampled TD-ConE$_{\text{REL}}$ scoring subset, for which we perform random sampling five times and keep the subset with the lowest TD-ConE$_{\text{REL}}$ score, [3] lowest TD-ConE 35k sentences, [4] for slight noise reduction via elimination of duplicates, lowest TD-ConE scoring 35k sentences with minimum TD-ConE $= 0.1$, and [5] highest TD-ConE scoring 35k sentences.[5] For each of the resulting five training sets, we compute TD-ConE$_{\text{REL}}$ against the validation set and the training set TD-ConE score. We aim to identify if TD-ConE$_{\text{REL}}$ can be used to select training data for a given validation set, whether there is a relationship between TD-ConE and TD-ConE$_{\text{REL}}$, and whether results across different data setups (Twitter, Quora) are consistent.

6.3 Results

TD-ConE$_{\text{REL}}$, TD-ConE, validation perplexity, and BLEU are reported in Table 3 for Twitter and Table 4 for Quora.

Lower TD-ConE $\not\Rightarrow$ lower TD-ConE$_{\text{REL}}$. There is no distinguishable relationship strictly between TD-ConE and TD-ConE$_{\text{REL}}$. On Twitter higher selection thresholds indicated higher TD-ConE$_{\text{REL}}$ and lower TD-ConE, yet we attribute this to selection via classifier confidence thresholds as the relationship does not hold with various selection methods on Quora. As an implication of this, the metrics reflect different but complementary information and are not merely interchangeable.

TD-ConE$_{\text{REL}}$ relates to validation perplexity & TD-ConE relates to BLEU. On both the Twitter and Quora datasets, TD-ConE$_{\text{REL}}$ scores generally align with downstream model validation perplexity, indicating a relationship between relative uncertainty of a validation set and the validation perplexity. Exemplifying this, the highest classifier confidence threshold on Twitter (0.80) had the largest between threshold increase in TD-ConE$_{\text{REL}}$ from 0.75 and a significant increase in validation perplexities across models. Interestingly, the inverse is also true with BLEU scores and TD-ConE: lower TD-ConE scores generally indicated higher BLEU scores. On Quora training set [3], in which no lower bound of TD-ConE score was imposed and therefore the set could contain identical sentence pairs, this effect was highly pronounced. When duplicate sentences were eliminated ([3] vs. [4]), we see a significant increase in uncertainty as measured by TD-ConE, which aligns with our definition of data uncertainty reflecting mapping complexities.

Divergences of lower TD-ConE & higher TD-ConE$_{\text{REL}}$: learning undesirable patterns. On Twitter, the thresholds exhibiting the highest TD-ConE$_{\text{REL}}$ scores (0.75, 0.80) exhibit the greatest divergence in TD-ConE and TD-ConE$_{\text{REL}}$ scores and are also those in which the TD-ConE score is lower than the TD-ConE$_{\text{REL}}$ score. This is observable on Quora as well with selection [3] (lowest TD-ConE sentences, no lower bound). Notably, these three columns are the only ones in which this pattern occurs, and have the highest validation perplexities while maintaining high BLEU scores. This suggests that divergence between TD-ConE and TD-ConE$_{\text{REL}}$ where TD-ConE < TD-ConE$_{\text{REL}}$ can indicate the model will bias towards undesirable patterns in the training data (i.e. simply copying input over to output), which limits the overall task information that is learned and increases the “surprise” the model experiences with unseen data.

Effective data selection for a given validation set. On Quora, we were able to utilize TD-ConE$_{\text{REL}}$ to inform the random sampling process with respect

[5] In [3, 4, 5] we treat each sentence pair as an individual corpus.

		Threshold					
Model	**Metric**	0.40	0.50	0.60	0.70	0.75	0.80
——	TD-CoNE$_{\text{REL}}$	0.240	0.242	0.244	0.244	0.251	0.272
	TD-CoNE	0.267	0.263	0.259	0.252	0.197	0.139
NMT	PPLX	73.46	73.80	73.99	75.20	88.14	**144.06**
	BLEU	20.01	20.6	19.69	19.27	20.55	**21.98**
CopyNMT	PPLX	57.97	63.65	58.51	65.68	80.60	**134.86**
	BLEU	20.71	20.75	21.12	20.92	22.32	**23.32**
GPT-2	PPLX	14.26	14.06	14.19	14.32	15.01	**17.84**
	BLEU	24.83	24.90	**24.99**	25.13	24.77	24.94

Table 3: Model performance on the Twitter validation set at different probability selection thresholds. We denote the highest BLEU scores (best performance metric) and highest validation perplexity (most uncertain model) in bold.

		Sampling Method				
		Random		Ordered		
Model	**Metric**	[1]	[2]	[3]	[4]	[5]
———	TD-CoNE$_{\text{REL}}$	0.152	0.150	0.229	0.169	0. 178
	TD-CoNE	0.246	0.246	0.141	0.243	0.365
NMT	PPLX	9.46	**9.09**	11.86	10.03	10.97
	BLEU	19.26	19.34	**20.17**	17.25	11.78
CopyNMT	PPLX	9.44	**9.09**	11.96	10.03	11.00
	BLEU	19.74	**20.35**	19.81	16.67	12.20

Table 4: Model performance with different data selection methods on the Quora dataset. Random (sampling) is performed using TD-CoNE$_{\text{REL}}$ and Ordered (sampling) is performed using TD-CoNE.

to the validation set as seen in training set [2]. Notably, when using TD-CoNE$_{\text{REL}}$ as a data selection method, we achieved highest performance on both NMT and CopyNMT: lowest TD-CoNE$_{\text{REL}}$, lowest perplexity, highest BLEU (other than the 0.0 dataset) with NMT, and lowest TD-CoNE$_{\text{REL}}$, lowest perplexity, highest BLEU with CopyNMT. As an ethical consideration, while validation sets are generally smaller with better documentation than large training sets, this could inadvertently propagate biases existing in a validation set by selecting a training set with similar biases.

7 Related work

Data quality evaluation. Data quality has received increased recent attention within both the natural language processing (NLP) and machine learning (ML) communities. Conceptually, quality is an abstract umbrella term that can encompass numerous dataset dimensions or characteristics. As a result, it has been operationally proxied through assessment of the value (Ghorbani and Zou, 2019), importance or influence (Jia et al., 2019; Pruthi et al., 2020), and learnability (Swayamdipta et al., 2020) of individual training instances, the presence and impact of dataset annotation artifacts or linguistic properties on task representativeness (Gururangan et al., 2018; Poliak et al., 2018), and the presence and impact of underlying dataset social (Rudinger et al., 2017) and gender (Lu et al., 2020) biases. Practically, the understanding of various quality dimensions informs dataset creators (Geva et al., 2019), enables bias mitigation strategies (Dixon et al., 2018), and contributes to development of data selection strategies (Moore and Lewis, 2010; Ruder and Plank, 2017). Our method contributes to the existing literature through proposing a method assess data for NLG tasks.

Alignment methods. There are a number of approaches to word alignment in bilingual settings, where a source language is mapped to a target language. These include statistical approaches such as the IBM Models (Brown et al., 1993) that utilize latent alignment variables, with implementations including GIZA++ (Och and Ney, 2003) and FastAlign (Dyer et al., 2013) which reparameterizes IBM Model 2, as well as statistical approaches using first order Hidden Markov Models (HMMs) (Vogel et al., 1996) and Markov Chain Monte Carlo inference (Östling et al., 2016). In addition to statistical approaches, recent approaches utilizing Transformers (Zenkel et al., 2020; Alkhouli et al., 2018) and pre-trained language models (Jalili Sabet et al., 2020) have shown success in neural machine translation. Additional approaches exist for alignment applications in monolingual settings, such as phrasal alignment (Yao et al., 2013), word sense alignment (Ahmadi and McCrae, 2021), text simplification (Albertsson et al., 2016), and disagreement detection (Gokcen and de Marneffe, 2015). Our method contributes to the literature by demonstrating how alignment can be utilized within a data assessment setting.

8 Conclusion

In this paper, we propose the method TD-CoNE and its extension TD-CoNE$_{\text{REL}}$ to assess text generation data. We design a simple alignment procedure for computing TD-CoNE and TD-CoNE$_{\text{REL}}$, and validate the metrics empirically using English text style transfer and paraphrase generation datasets. While currently limited to parallel data with one-to-one sentence pairs, future work can look at non-parallel data and multiple outputs.

References

Sina Ahmadi and John P. McCrae. 2021. Monolingual word sense alignment as a classification problem. In *Proceedings of the 11th Global Wordnet Conference*, pages 73–80, University of South Africa (UNISA). Global Wordnet Association.

Sarah Albertsson, Evelina Rennes, and Arne Jönsson. 2016. Similarity-based alignment of monolingual corpora for text simplification purposes. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 154–163.

Danial Alihosseini, Ehsan Montahaei, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Christin Beck, Hannah Booth, Mennatallah El-Assady, and Miriam Butt. 2020. Representation problems in linguistic annotations: Ambiguity, variation, uncertainty, error and bias. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 60–73, Barcelona, Spain. Association for Computational Linguistics.

Emily M Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Wanyu Du and Yangfeng Ji. 2019. An empirical comparison on imitation learning and reinforcement learning for paraphrase generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6012–6018, Hong Kong, China. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Yarin Gal. 2016. Uncertainty in deep learning.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. StyleNet: Generating Attractive Visual Captions with Styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 955–964, Honolulu, HI. IEEE.

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics.

Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520. IEEE.

Ajda Gokcen and Marie-Catherine de Marneffe. 2015. I do not disagree: leveraging monolingual alignment to detect disagreement in dialogue. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 94–99, Beijing, China. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of* the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19, Copenhagen, Denmark. Association for Computational Linguistics.

Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. 2019. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? *arXiv preprint arXiv:1911.07128*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of The 2017 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 1235–1245. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018a. Delete, retrieve, generate: a simple approach to sentiment and style transfer. 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 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018b. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. In *Logic, Language, and Security*, pages 189–202. Springer.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Robert Östling, Jörg Tiedemann, et al. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*.

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

Jeffrey Pennington, Richard Socher, and Christopher D 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.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically Neutralizing Subjective Bias in Text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):480–489. Number: 01.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. 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 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, Valencia, Spain. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020. Multi-Task Self-Supervised Learning for Disfluency Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9193–9200. Number: 05.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578, Hong Kong, China. Association for Computational Linguistics.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India. The COLING 2012 Organizing Committee.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Semi-Markov phrase-based monolingual alignment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 590–600, Seattle, Washington, USA. Association for Computational Linguistics.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2020. End-to-end neural word alignment outperforms GIZA++. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1605–1617, Online. Association for Computational Linguistics.

Mike Zhang and Barbara Plank. 2021. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. *Advances in Neural Information Processing Systems*, 31:1810–1820.

A Dataset Details

Dataset	Tasks	**Train**	**Dev**	**Test**
Captions	Romantic→Humorous Humorous→Romantic	6k	500	500
Shakespeare	Modern→Shakespeare Shakespeare→Modern	18.4k	1.2k	1.5k
GYAFC-FR	Informal→Formal	52k	2.8k	1.3k
GYAFC-EM	Informal→Formal	52.6k	2.9k	1.4k
Biased-word	Subjective→Neutral	53.8k	700	1k
Fluency	Disfluent→Fluent	173.7k	10.1k	7.9k

Table 5: Dataset statistics.

Dataset selection For text style transfer datasets, selection criteria included: parallel datasets with two classes pertaining to the presence or lack of a single stylistic attribute that had been previously benchmarked with BLEU and accuracy. Datasets can be obtained or requested through links found in the respective cited source papers.

Fluency Contains aligned sentence pairs from the English Switchboard (SWBD) Corpus (Godfrey et al., 1992). Each sentence is labeled as either fluent or disfluent (Wang et al., 2020).

GYAFC *GYAFC-EM* contains aligned sentence pairs from the *Entertainment & Music* domain of Yahoo Answers, a question answering forum (Rao and Tetreault, 2018). *GYAFC-FR* contains aligned sentence pairs from the *Family & Relationships* domain of Yahoo Answers (Rao and Tetreault, 2018). Since both datasets are sourced online from Yahoo Answers, there is some potential for offensive language. [6]

Biased-Word Contains aligned sentence pairs pre- and post- neutralization, crawled from 423,823 Wikipedia editor revisions between 2004 and 2019 (Pryzant et al., 2020).

Captions Contains sentences that describe an image, labeled romantic or humorous. A distribution of this dataset from (Li et al., 2018a) includes factual descriptions for 300 images and has been used for style transfer in an unaligned manner. However, in our context, we use the original Flickr dataset with a 6000/500/500 train-dev-test split in an aligned manner as in the original paper (Gan et al., 2017).

Shakespeare Contains aligned original and modern sentence pairs from 17 of Shakespeare's 36 plays, crawled from Sparknotes [7] (Xu et al., 2012). Following Jhamtani et al. (2017), we use 15 plays for training, leaving *Twelfth Night* for validation, and *Romeo and Juliet* for testing.

Paraphrase generation: Twitter URL & Quora: Twitter URL contains 51k human annotated sentence pairs labeled with the number of human annotators (out of six) that labeled a pair of sentences as paraphrases, and 2.87 million candidate pairs automatically labeled with predicted probability from a classifier trained on the manually annotated sentence pairs. In prior work (Li et al., 2018b; Du and Ji, 2019), a probability threshold is often picked to select a subset of the automatically annotated pairs as a training set, while the validation and test set are sampled from the manually annotated pairs: our experiments follow this procedure. Quora Question Pairs contains 404k question pairs with binary labels indicating whether the pair are paraphases, from which prior studies (Li et al., 2018b; Du and Ji, 2019) sample train/validation/test data splits.

Quora license information (License Other) can be found referenced at https://www.kaggle.com/quora/question-pairs-dataset/metadata. Twitter URL is released for non-commercial use under the CC BY-NC-SA 3.0 license, and can be requested at https://languagenet.github.io/.

Additional details about data usage: Where available, we used original or existing train-validation-test dataset splits, including the train-validation-test split for Shakespeare as in Jhamtani et al. (2017). For Captions (Flickr), as only the original 7k training instances are available, we create a 6000-500-500 dataset split, and for the GYAFC datasets, for the tuning and test sets we used the informal text and all 4 available human formal rewrites. Regarding consent, for datasets using online data sources, such as GYAFC (Yahoo) and Twitter, users consent to the website's terms

[6] GYAFC-EM & GYAFC-FR datasets can be requested at https://github.com/raosudha89/GYAFC-corpus

[7] https://www.sparknotes.com/

and conditions. Datasets utilizing annotators are also assumed to have annotator consent.

B Additional Tables

Tables for TD-CoNE scores and target sentence entropies for text style transfer datasets.

		TD-CoNE		
Dataset	**Task**	*train*	*dev*	*test*
Captions	Rom→Fun	0.3980	0.3821	0.3636
	Fun→Rom	0.3839	0.3491	0.3413
Shakespeare	Mod→Shake	0.2826	0.2976	0.2653
	Shake→Mod	0.2787	0.2866	0.2578
GYAFC-FR	Inf→Form	0.2433	0.2176	0.1954
GYAFC-EM	Inf→Form	0.2205	0.1980	0.1864
Biased	Subj→Neut	0.0078	0.0038	0.0042
Fluency	Disf→Flt	0.0052	0.0061	0.0063

Table 6: TD-CoNE scores on text style transfer datasets.

Dataset	Target	H(Target)
Captions	Funny	0.6743
Captions	Romantic	0.6726
Shakespeare	Shake.	0.6505
Shakespeare	Modern	0.6436
GYAFC-FR	Formal	0.6086
GYAFC-EM	Formal	0.6172
Biased	Neutral	0.6445
Fluency	Fluent	0.6050

Table 7: Entropies of target vocabulary distributions on style transfer datasets.

C Uniform Alignments

While the alignment method used for TD-CoNE and TD-CoNE$_{\text{REL}}$ utilizes the cosine similarities to map words across class boundaries in a sentence pair, we can also utilize a uniform alignment over the number of target words that cannot be aligned with 1-to-1 mappings, shown in Algorithm 2.

D Detailed Word Alignment Algorithm Description

We categorize potential alignments between the input and output words from a sentence pair $(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)})$ into three cases: (1) if a word is shared between $\boldsymbol{x}$ and $\boldsymbol{y}$, then we consider it to have

Algorithm 2 Calculating the alignment matrix with one pair of sentences

Input: a sentence pair $\boldsymbol{s}$ and $\boldsymbol{t}$, alignment matrix $\boldsymbol{M}$
Output: the updated alignment matrix $\boldsymbol{M}$
for word $w \in \boldsymbol{s}$ **do**
  if $w \in \boldsymbol{s} \cap \boldsymbol{t}$ **then**
    $\boldsymbol{M}(w, w) \leftarrow \boldsymbol{M}(w, w) + 1$
  if $w \notin \boldsymbol{t} \backslash \boldsymbol{s}$ **then**
    for $w' \in (\boldsymbol{t} \backslash \boldsymbol{s}) \cup \{\text{NULL}\}$ **do**
      $\boldsymbol{M}(w, w') \leftarrow \boldsymbol{M}(w, w') + \frac{1}{|(\boldsymbol{t} \backslash \boldsymbol{s}) \cup \{\text{NULL}\}|}$
if $\boldsymbol{s} \subset \boldsymbol{t}$ **then**
  for word $w' \in \boldsymbol{t} \backslash \boldsymbol{s}$ **do**
    $\boldsymbol{M}(\text{NULL}, w') \leftarrow \boldsymbol{M}(\text{NULL}, w') + \frac{1}{|\boldsymbol{t} \backslash \boldsymbol{s}|}$

a deterministic alignment from source to target (line 4 in algorithm 1); (2) if an input word w is not in the output sentence $\boldsymbol{y}$ and no other alignments can be made, w is aligned with NULL (where $|\boldsymbol{y} \backslash \boldsymbol{x}| = 0$). If $|\boldsymbol{y} \backslash \boldsymbol{x}| > 0$, a probability distribution is computed over the cosine similarities between the GloVe word embeddings (Pennington et al., 2014) of w and each w' in $\boldsymbol{y} \backslash \boldsymbol{x}$. If a w or w' is out-of-vocabulary, we utilize a uniform probability over the size of $\boldsymbol{y} \backslash \boldsymbol{x}$ (lines 5 – 11 in algorithm 1);[8] (3) all the unique words w' in $\boldsymbol{y} \backslash \boldsymbol{x}$ where $\boldsymbol{x} \subset \boldsymbol{y}$ have an alignment from the NULL token on the input side utilizing a uniform distribution over $|\boldsymbol{y} \backslash \boldsymbol{x}|$ (lines 12 – 13 in algorithm 1). Two special scenarios remain: $\boldsymbol{y} \subset \boldsymbol{x}$ and $\boldsymbol{x} \subset \boldsymbol{y}$. To accurately estimate $P(X)$, if $\boldsymbol{y} \subset \boldsymbol{x}$ we must increment the target NULL by 1, and if $\boldsymbol{x} \subset \boldsymbol{y}$ we must increment the source NULL uniformly over $\boldsymbol{y} \backslash \boldsymbol{x}$ to ensure the dependency between input and output. Once we have $\boldsymbol{M}$ estimated over the entire dataset D, $P(Y|X = w)$ is obtained by normalizing the corresponding row in $\boldsymbol{M}$.

E Training Details

Model Implementations For NMT and CopyNMT, we use implementations provided by OpenNMT (Klein et al., 2017). For GPT-2 we use the implementation code provided by (Wang et al., 2019).

Paraphrase Generation Experiments NMT, CopyNMT and GPT-2 models were run on a sin-

[8] We describe a simplified version of the alignment algorithm using uniform probability alignments in Appendix C

gle NVIDIA GTX 1080 Ti GPU. For CopyNMT and NMT, we utilized 2-layer LSTMs for the encoder and decoder with attention (Bahdanau et al., 2014) and 500 hidden states. Adam optimization (Kingma and Ba, 2014) was used for both models with learning rate 0.001. While most model parameters were simply set to the default OpenNMT parameter settings, we chose our optimization method and learning rate after noting issues with convergence when using stochastic gradient descent. We utilized a random seed for consistency. For decoding, we utilized argmax decoding after finding performance degradation with beam search with beam sizes 2 and 3. All models were selected based on highest validation performance.

The GPT-2 model was run on a single NVIDIA GTX 1080 Ti GPU. We use the implementation code provided by (Wang et al., 2019), which can be found at https://github.com/jimth001/formality_emnlp19. For training, we chose the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.00001, set batch size to 16, and set total training steps to 50000, which are the default settings in the original implementation. During training, we found the training loss decreased rapidly. In order to save the optimal model checkpoint and avoid overfitting, we performed auto validation on the development set every 10 steps, and applied early stopping when the validation loss did not drop after 100 steps. For generation, we applied beam search with beam size 4. We set the maximum generation length to 100, since the majority of sentences had a length of less than 100 tokens.

Style Transfer Experiments The GPT-2 model was run on a single NVIDIA GTX 1080 Ti GPU. We use the implementation code provided from (Wang et al., 2019). For experiments across all 6 datasets, we chose the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.00001, set batch size to 16, and set total training steps to 50000, which are the default settings in the original implementation. As the training loss decreased rapidly, in order to save the optimal model checkpoint and avoid overfitting, we performed auto validation on the development set every 10 steps, and applied early stopping when the validation loss did not decrease after 100 steps. For generation, we apply beam search with beam size 4. We set the maximum generation length to 200, since the majority of sentences was less than 200 tokens in length.

NeuralMT (NMT) models were run on a single NVIDIA GeForce RTX 2080 GPU. Default OpenNMT hyper-parameters were used, including stochastic gradient descent (SGD) optimization with a learning rate of 1.0. CopyNMT models were also run on a single NVIDIA GeForce RTX 2080 GPU. We set word vector size to 300 and used an SGD optimizer with a learning rate of 1.0. We used an MLP attention mechanism and reused attention scores for copying scores.

For BART models, we used Adam optimization with warmup and polynomial decaying. The maximum learning rate was set to 1e-5, and warmup steps were set to 500. Batch size was 8192 tokens. We also used dropout and attention dropout with a 0.1 dropout rate. Label smoothing was used with a 0.1 label smoothing rate. We used 0.01 as the weight for weight decay. Other hyper-parameters were set to default Fairseq hyper-parameters. We followed the default hyper-parameters used for text summarization and adjusted the max learning rate from 3e-5 to 1e-5 for better convergence.

License Information License details for OpenNMT (NMT and CopyNMT models) can be found at https://github.com/OpenNMT/OpenNMT-py/blob/master/LICENSE.md. Assets from Huggingface (GPT-2 and BERT-base-uncased) are Licensed under the Apache License, Version 2.0 (Copyright 2020, The Hugging Face Team).