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_REL to compute the relative uncertainty between two datasets. Our experiments with paraphrase generation datasets demonstrate that selecting data with lower TD-CONE_REL scores leads to better model performance and decreased validation perplexity.

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_{REL}; 2) proposing simple yet effective word alignment methods without any learned parameters for computing TD-CONE and TD-CONE_{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_{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 $V_x$, and $Y$ represent a word in the output vocabulary $V_y$, then this conditional entropy provides us a starting point of defining our task-specific data uncertainty.

**Definition 1 (TD-CONE).** The Task-Dataset Conditional Entropy (TD-CONE) is defined as

$$TD-CONE(Y \mid X) = \frac{H(Y \mid X)}{\log |V_y|} \quad (1)$$

where $H(Y \mid X)$ is the conditional entropy, and $|V_y|$ is the size of the output vocabulary.

The denominator $|V_y|$ normalizes the value of $H(Y \mid X)$ and guarantees TD-CONE$(Y \mid X)$ always bounded between 0 and 1. Specifically, we have $0 \leq H(Y \mid X) \leq H(Y) \leq \log |V_y|$ (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 $x$ map to a given word in $y$ 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 $x = \{x_1, \ldots, x_m\}$ represent one input sentence with $m$ words and $y = \{y_1, \ldots, y_n\}$ 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 \( D = \{(x^{(k)}, y^{(k)})\}_{k=1}^K \), where \( K \) is the total number of examples. With the dataset \( D \), we can define \( V_x \) as the input vocabulary constructed from \( \{x^{(k)}\} \) and \( V_y \) as the output vocabulary constructed from \( \{y^{(k)}\} \). Our problem setup is therefore to estimate the conditional probability \( P(Y \mid X) \) given the dataset \( D \), where \( X \in V_x \) and \( Y \in V_y \).

For a given dataset, the challenge of estimating \( P(Y \mid X) \) for a specific output word \( y^{(k)}_j \) is to identify which word (or words) in \( x^{(k)} \) “generate” (i.e. are aligned with) \( y^{(k)}_j \). 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 \( M \in \mathbb{R}^{|V_x| \times |V_y|} \) to record the alignment counts based on \( D \). 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 \( M \) has been estimated over the entire dataset \( D \), \( P(Y \mid X = w) \) is obtained by normalizing the corresponding row in \( M \).

We describe a deterministic version of the alignment algorithm using uniform probability alignments in Appendix C, which also had good preliminary results. 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

---

Algorithm 1 Calculating the alignment matrix with one pair of sentences

1: **Input**: a sentence pair \( x \) and \( y \), alignment matrix \( M \)
2: **Output**: the updated alignment matrix \( M \)
3: for word \( w \in x \) do
4: \[ \text{if } w \in x \cap y \text{ then } M(w, w) \leftarrow M(w, w) + 1 \]
5: \[ \text{if } w \notin y \setminus x \text{ then} \]
6: \[ \text{if } |y \setminus x| = 0 \text{ then} \]
7: \[ M(w, \text{NULL}) \leftarrow M(w, \text{NULL}) + 1 \]
8: \[ \text{else} \]
9: \[ \text{for } w' \in (y \setminus x) \text{ do} \]
10: \[ \text{score} \leftarrow (w, w' \in \text{EMBEDS})? \]
11: \[ \text{for } \frac{\parallel w' \parallel \parallel w \parallel}{\text{score}} : \frac{1}{|y \setminus x|} \]
12: \[ \text{if } x \subset y \text{ then} \]
13: \[ \text{for word } w' \in y \setminus x \text{ do } M(\text{NULL}, w') \leftarrow M(\text{NULL}, w') + \frac{1}{|y \setminus x|} \]

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

---

1 A detailed description can be found in Appendix D
2 Code for uniform and static alignments will be released.
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, romantic to humorous) (Gan et al., 2017), and 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).

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. 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

3Correlations are reported alongside other alignment methods in section 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 \( y \setminus x \) for each \( w \) when the symmetric difference \( x \Delta y \neq \emptyset \). 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: \( y \subset x \) and \( x \subset y \). If \( y \subset x \) we increment the target NULL by 1 to ensure accurate estimation of \( P(X) \), and if \( x \subset y \) we increment the source NULL uniformly over \( y \setminus x \) 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-CONEREL: 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-CONEREL).** 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-CONEREL or "Task-Dataset Conditional Entropy: Relative Entropy" can be defined as the normalized conditional relative entropy between \( P \) and \( Q \):

\[
\text{TD-CONEREL}_\text{relative} = \frac{KL(P(Y \mid X) \| Q(Y \mid X))}{KL(P(Y \mid X) \| U(Y \mid X))}
\]

(2)

where \( KL(P(Y \mid X) \| Q(Y \mid X)) = \sum_{X,Y} P(X,Y) \log \frac{P(Y \mid X)}{Q(Y \mid X)} \) 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 \( \text{TD-CONEREL} \geq 0 \). In addition, since \( U(Y \mid X) \) is a uniform distribution and therefore \( KL(P \| Q) \leq KL(P \| U) \) always holds, we have \( 0 \leq \text{TD-CONEREL} \leq 1 \). Given two datasets \( D_p \) and \( D_q \), \( P(Y \mid X) \) and \( Q(Y \mid X) \) can be estimated using the same algorithm proposed in section 3, enabling computation of TD-CONEREL prior to any model training.

6 TD-CONEREL Experiments

We expect that lower TD-CONEREL 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-CONEREL 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-CONEREL 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-CONEREL, 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-CONERel displays a relationship with selection threshold or model performance.

On Quora Question Pairs. We use the combination of TD-CONE and TD-CONERel 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-CONERel scoring subset, for which we perform random sampling five times and keep the subset with the lowest TD-CONERel 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. For each of the resulting five training sets, we compute TD-CONERel against the validation set and the training set TD-CONE score. We aim to identify if TD-CONERel can be used to select training data for a given validation set, whether there is a relationship between TD-CONE and TD-CONERel, and whether results across different data setups (Twitter, Quora) are consistent.

6.3 Results

TD-CONERel, TD-CONE, validation perplexity, and BLEU are reported in Table 3 for Twitter and Table 4 for Quora.

Lower TD-CONE ≠ lower TD-CONERel.

There is no distinguishable relationship strictly between TD-CONE and TD-CONERel. On Twitter higher selection thresholds indicated higher TD-CONERel and lower TD-CONE, yet we attribute

\[\text{ON} \Rightarrow \text{CONERel} \leq \text{ON} \]

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-CONERel relates to validation perplexity & TD-CONE relates to BLEU.** On both the Twitter and Quora datasets, TD-CONERel 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-CONERel 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-CONERel: learning undesirable patterns.** On Twitter, the thresholds exhibiting the highest TD-CONERel scores (0.75, 0.80) exhibit the greatest divergence in TD-CONE and TD-CONERel scores and are also those in which the TD-CONE score is lower than the TD-CONERel 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-CONERel, where TD-CONE < TD-CONERel 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-CONERel to inform the random sampling process with respect
Table 4: Model performance with different data selection thresholds. We denote the highest BLEU scores (best performance metric) and highest validation perplexity (most uncertain model) in bold.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TD-CONE</td>
<td>0.152</td>
<td>0.150</td>
<td>0.229</td>
<td>0.169</td>
<td>0.178</td>
<td>0.246</td>
<td>0.246</td>
<td>0.141</td>
<td>0.243</td>
<td>0.365</td>
</tr>
<tr>
<td>CopyNMT</td>
<td>PPLX</td>
<td>9.46</td>
<td>9.09</td>
<td>11.86</td>
<td>10.03</td>
<td>10.97</td>
<td>9.44</td>
<td>9.09</td>
<td>11.96</td>
<td>10.03</td>
<td>11.00</td>
</tr>
</tbody>
</table>

Table 3: Model performance on the Twitter validation set at different probability selection thresholds. We denote the highest BLEU scores (best performance metric) and lowestTD-CONE as a data selection method, we achieved highest performance on both NMT and CopyNMT: lowest TD-CONEREL, lowest perplexity, highest BLEU (other than the 0.0 dataset) with NMT, and lowest TD-CONEREL, 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 data 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-CONEREL to assess text generation data. We design a simple alignment procedure for computing TD-CONE and TD-CONEREL, 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


Ruoxi Jia, Fan Wu, Xuehui Sun, Jiachen Xu, David Dao, Bhavya Kakihara, 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.


A Dataset Details

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Tasks</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Captions</td>
<td>Romantic→Humorous</td>
<td>6k</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Captions</td>
<td>Humorous→Romantic</td>
<td>6k</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>Modern→Shakespeare</td>
<td>18.4k</td>
<td>1.2k</td>
<td>1.5k</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>Shakespeare→Modern</td>
<td>18.4k</td>
<td>1.2k</td>
<td>1.5k</td>
</tr>
<tr>
<td>GYAF-C-FR</td>
<td>Informal→Formal</td>
<td>52k</td>
<td>2.8k</td>
<td>1.3k</td>
</tr>
<tr>
<td>GYAF-C-EM</td>
<td>Informal→Formal</td>
<td>52.6k</td>
<td>2.9k</td>
<td>1.4k</td>
</tr>
<tr>
<td>Biased-word</td>
<td>Subjective→Neutral</td>
<td>53.8k</td>
<td>700</td>
<td>1k</td>
</tr>
<tr>
<td>Fluency</td>
<td>Disfluent→Fluent</td>
<td>173.7k</td>
<td>10.1k</td>
<td>7.9k</td>
</tr>
</tbody>
</table>

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 edits following 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 (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 paraphrases, 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 used a 6000-500-500 train-dev-test split in an aligned manner as in the original paper (Gan et al., 2017). For Biased-Word, a classifier trained on the manually annotated sentence pairs is used to automatically select 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 paraphrases, 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

---

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

7https://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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>TD-CONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>train</strong></td>
</tr>
<tr>
<td>Captions</td>
<td>Rom→Fun</td>
<td>0.3980</td>
</tr>
<tr>
<td>Captions</td>
<td>Fun→Rom</td>
<td>0.3839</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>Mod→Shake</td>
<td>0.2826</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>Shake→Mod</td>
<td>0.2787</td>
</tr>
<tr>
<td>GYAF-EM</td>
<td>Inf→Form</td>
<td>0.2433</td>
</tr>
<tr>
<td>GYAF-EM</td>
<td>Inf→Form</td>
<td>0.2205</td>
</tr>
<tr>
<td>Biased</td>
<td>Subj→Neut</td>
<td>0.0078</td>
</tr>
<tr>
<td>Fluency</td>
<td>Dis→Flt</td>
<td>0.0052</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Target</th>
<th>H(Target)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Captions</td>
<td>Funny</td>
<td>0.6743</td>
</tr>
<tr>
<td>Captions</td>
<td>Romantic</td>
<td>0.6726</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>Shake.</td>
<td>0.6505</td>
</tr>
<tr>
<td>Shakespeare</td>
<td>Modern</td>
<td>0.6436</td>
</tr>
<tr>
<td>GYAF-EM</td>
<td>Formal</td>
<td>0.6086</td>
</tr>
<tr>
<td>GYAF-EM</td>
<td>Formal</td>
<td>0.6172</td>
</tr>
<tr>
<td>Biased</td>
<td>Neutral</td>
<td>0.6445</td>
</tr>
<tr>
<td>Fluency</td>
<td>Fluent</td>
<td>0.6050</td>
</tr>
</tbody>
</table>

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

**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.
ingle 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).