Abstract

In this paper, we present a novel method for measurably adjusting the semantics of text while preserving its sentiment and fluency, a task we call semantic text exchange (STE). This is useful for text data augmentation and the semantic correction of text generated by chatbots and virtual assistants. We introduce a pipeline called SMERTI that combines entity replacement, similarity masking, and text infilling. We measure our pipeline’s success by its Semantic Text Exchange Score (STES): the ability to preserve the original text’s sentiment and fluency while adjusting semantic content. We propose to use masking (replacement) rate threshold as an adjustable parameter to control the amount of semantic change in the text. Our experiments demonstrate that SMERTI can outperform baseline models on Yelp reviews, Amazon reviews, and news headlines.

1 Introduction

There has been significant research on style transfer, with the goal of changing the style of text while preserving its semantic content. The alternative where semantics are adjusted while keeping style intact, which we call semantic text exchange (STE), has not been investigated to the best of our knowledge. Consider the following example, where the replacement entity defines the new semantic context:

| Original Text | Desired Text |
|---------------|--------------|
| It is sunny outside! Ugh, that means I must wear sunscreen. I hate being sweaty and sticky all over. | It is rainy outside! Ugh, that means I must bring an umbrella. I hate being wet and having to carry it around. |

The weather within the original text is sunny, whereas the actual weather may be rainy. Not only is the word sunny replaced with rainy, but the rest of the text’s content is changed while preserving its negative sentiment and fluency.

With the rise of natural language processing (NLP) has come an increased demand for massive amounts of text data. Manually collecting and scraping data requires a significant amount of time and effort, and data augmentation techniques for NLP are limited compared to fields such as computer vision. STE can be used for text data augmentation by producing various modifications of a piece of text that differ in semantic content.

Another use of STE is in building emotionally aligned chatbots and virtual assistants. This is useful for reasons such as marketing, overall enjoyment of interaction, and mental health therapy. However, due to limited data with emotional content in specific semantic contexts, the generated text may contain incorrect semantic content. STE can adjust text semantics (e.g. to align with reality or a specific task) while preserving emotions.

One specific example is the development of virtual assistants with adjustable socio-emotional personalities in the effort to construct assistive technologies for persons with cognitive disabilities. Adjusting the emotional delivery of text in subtle ways can have a strong effect on the adoption of the technologies (Robillard et al., 2018). It is challenging to transfer style this subtly due to lack of datasets on specific topics with consistent emotions. Instead, large datasets of emotionally consistent interactions not confined to specific topics exist. Hence, it is effective to generate text with a particular emotion and then adjust its semantics.

We propose a pipeline called SMERTI (pronounced ‘smarty’) for STE. Combining entity replacement (ER), similarity masking (SM), and text
infilling (TI), SMERTI can modify the semantic content of text. We define a metric called the Semantic Text Exchange Score (STES) that evaluates the overall ability of a model to perform STE, and an adjustable parameter masking (replacement) rate threshold (MRT/RRT) that can be used to control the amount of semantic change.

We evaluate on three datasets: Yelp and Amazon reviews (He and McAuley, 2016), and Kaggle news headlines (Misra, 2018). We implement three baseline models for comparison: Noun WordNet Semantic Text Exchange Model (NWN-STEM), General WordNet Semantic Text Exchange Model (GWN-STEM), and Word2Vec Semantic Text Exchange Model (W2V-STEM).

We illustrate the STE performance of two SMERTI variations on the datasets, demonstrating outperformance of the baselines and pipeline stability. We also run a human evaluation supporting our results. We analyze the results in detail and investigate relationships between the semantic change, fluency, sentiment, and MRT/RRT. Our major contributions can be summarized as:

- We define a new task called semantic text exchange (STE) with increasing importance in NLP applications that modifies text semantics while preserving other aspects such as sentiment.
- We propose a pipeline SMERTI capable of multi-word entity replacement and text infilling, and demonstrate its outperformance of baselines.
- We define an evaluation metric for overall performance on semantic text exchange called the Semantic Text Exchange Score (STES).

## 2 Related Work

### 2.1 Word and Sentence-level Embeddings

Word2Vec (Mikolov et al., 2013a,b) allows for analogy representation through vector arithmetic. We implement a baseline (W2V-STEM) using this technique. The Universal Sentence Encoder (USE) (Cer et al., 2018) encodes sentences and is trained on a variety of web sources and the Stanford Natural Language Inference corpus (Bowman et al., 2015). Flair embeddings (Akbik et al., 2018) are based on architectures such as BERT (Devlin et al., 2019). We use USE for SMERTI as it is designed for transfer learning and shows higher performance on textual similarity tasks compared to other models (Perone et al., 2018).

### 2.2 Text Infilling

Text infilling is the task of filling in missing parts of sentences called masks. MaskGAN (Fedus et al., 2018) is restricted to a single word per mask token, while SMERTI is capable of variable length infilling for more flexible output. Zhu et al. (2019) uses a transformer-based architecture. They fill in random masks, while SMERTI fills in masks guided by semantic similarity, resulting in more natural infilling and fulfillment of the STE task.

### 2.3 Style and Sentiment Transfer

Notable works in style/sentiment transfer include (Shen et al., 2017; Fu et al., 2018; Li et al., 2018; Xu et al., 2018). They attempt to learn latent representations of various text aspects such as its context and attributes, or separate style from content and encode them into hidden representations. They then use an RNN decoder to generate a new sentence given a targeted sentiment attribute.

### 2.4 Review Generation

Hovy (2016) generates fake reviews from scratch using language models. (Lipton et al., 2015; Dong et al., 2017; Juuti et al., 2018) generate reviews from scratch given auxiliary information (e.g. the item category and star rating). Yao et al. (2017) generates reviews using RNNs with two components: generation from scratch and review customization (Algorithm 2 in Yao et al. (2017)). They define review customization as modifying the generated review to fit a new topic or context, such as from a Japanese restaurant to an Italian one. They condition on a keyword identifying the desired context, and replace similar nouns with others using WordNet (Miller, 1995). They require a “reference dataset” (required to be “on topic”; easy enough for restaurant reviews, but less so for arbitrary conversational agents). As noted by Juuti et al. (2018), the method of Yao et al. (2017) may also replace words independently of context. We implement their review customization algorithm (NWN-STEM) and a modified version (GWN-STEM) as baseline models.

## 3 SMERTI

### 3.1 Overview

The task is to transform a corpus $C$ of lines of text $S_i$ and associated replacement entities $RE_i : C = \{(S_1, RE_1), (S_2, RE_2), \ldots, (S_n, RE_n)\}$ to a modified corpus $\hat{C} = \{\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_n\}$, where
Figure 1: Overall architecture and example, showing the three modules: Entity Replacement (ERM), Similarity Masking (SMM), and Text Infilling (TIM)

$\hat{S}_i$ are the original text lines $S_i$ replaced with $RE_i$ and overall semantics adjusted. SMERTI consists of the following modules, shown in Figure 1:

1. **Entity Replacement Module (ERM)**: Identify which word(s) within the original text are best replaced with the $RE$, which we call the Original Entity (OE). We replace OE in $S$ with $RE$. We call this modified text $S'$.

2. **Similarity Masking Module (SMM)**: Identify words/phrases in $S'$ similar to OE and replace them with a [mask]. Group adjacent [mask]s into a single one so we can fill a variable length of text into each. We call this masked text $S''$.

3. **Text Infilling Module (TIM)**: Fill in [mask] tokens with text that better suits the $RE$. This will modify semantics in the rest of the text. This final output text is called $\hat{S}$.

### 3.2 Entity Replacement Module (ERM)

For entity replacement, we use a combination of the Universal Sentence Encoder (USE) and Stanford Parser (Chen and Manning, 2014).

**Stanford Parser**

The Stanford Parser is a constituency parser that determines the grammatical structure of sentences, including phrases and part-of-speech (POS) labelling. By feeding our $RE$ through the parser, we are able to determine its parse-tree. Iterating through the parse-tree and its sub-trees, we can obtain a list of constituent tags for the $RE$.

We then feed our input text $S$ through the parser, and through a similar process, we can obtain a list of leaves (where leaves under a single label are concatenated) that are equal or similar to any of the $RE$ constituent tags. This generates a list of entities having the same (or similar) grammatical structure as the $RE$, and are likely candidates for the $OE$. We then feed these entities along with the $RE$ into the Universal Sentence Encoder (USE).

**Universal Sentence Encoder (USE)**

The USE is a sentence-level embedding model that comes with a deep averaging network (DAN) and transformer model (Cer et al., 2018). We choose the transformer model as these embeddings take context into account, and the exact same word/phrase will have a different embedding depending on its context and surrounding words.

We compute the semantic similarity between two embeddings $u$ and $v$: $\text{sim}(u, v)$, using the angular (cosine) distance, defined as: $\cos(\theta_{u,v}) = (u \cdot v) / (||u|| ||v||)$, such that $\text{sim}(u, v) = 1 - \frac{\pi}{2} \arccos(\cos(\theta_{u,v}))$. Results are in $[0, 1]$, with higher values representing greater similarity.

Using USE and the above equation, we can identify words/phrases within the input text $S$ which are most similar to $RE$. To assist with this, we use the Stanford Parser as described above to obtain a list of candidate entities. In the rare case that this list is empty, we feed in each word of $S$ into USE, and identify which word is the most similar to $RE$. We then replace the most similar entity or word (OE) with the $RE$ and generate $S'$.

An example of this entity replacement process is in Figure 2. Two parse-trees are shown: for $RE$ (a) and $S$ (b) and (c). Figure 2(d) is a semantic similarity heat-map generated from the USE embeddings of the candidate OEs and $RE$, where values are similarity scores in the range $[0, 1]$.

As seen in Figure 2(d), we calculate semantic similarities between $RE$ and entities within $S$ which have noun constituency tags. Looking at the row for our $RE$ *restaurant*, the most similar entity (excluding itself) is *hotel*. We can then generate:

$S' = \text{i love this restaurant ! the beds are comfortable and the service is great!}$

### 3.3 Similarity Masking Module (SMM)

Next, we mask words similar to OE to generate $S''$ using USE. We look at semantic similarities between every word in $S$ and $OE$, along with semantic similarities between $OE$ and the candidate entities determined in the previous ERM step to broaden the range of phrases our module can mask. We ignore $RE$, $OE$, and any entities or phrases containing $OE$ (for example, *this hotel*).

After determining words similar to the OE (discussed below), we replace each of them with a
[mask] token. Next, we replace [mask] tokens adjacent to each other with a single [mask].

We set a base similarity threshold (ST) that selects a subset of words to mask. We compare the actual fraction of masked words to the masking rate threshold (MRT), as defined by the user, and increase ST in intervals of 0.05 until the actual masking rate falls below the MRT. Some sample masked outputs ($S'$) using various MRT-ST combinations for the previous example are shown in Table 1 (more examples in Appendix A).

The MRT is similar to the temperature parameter used to control the “novelty” of generated text in works such as Yao et al. (2017). A high MRT means the user wants to generate text semantically different from the original, and may be desired in cases such as text recovery, grammar correction, or correcting a minor semantic error in text. By varying the MRT, various pieces of text that differ semantically in subtle ways can be generated, assisting greatly with text data augmentation. The MRT also affects sentiment and fluency, as we show in Section 6.5.

3.4 Text Infilling Module (TIM)

We use two seq2seq models for our TIM: an RNN (recurrent neural network) model (Sutskever et al., 2014) (called SMERTI-RNN), and a transformer model (called SMERTI-Transformer).

Bidirectional RNN with Attention

We use a bidirectional variant of the GRU (Cho et al., 2014), and hence two RNNs for the encoder: one reads the input sequence in standard sequential order, and the other is fed the sequence in reverse. The outputs are summed at each time step, giving us the ability to encode information from both past and future context.

The decoder generates the output in a sequential token-by-token manner. To combat information loss, we implement the attention mechanism (Bahdanau et al., 2015). We use a Luong attention layer (Luong et al., 2015) which uses global attention, where all the encoder’s hidden states are considered, and use the decoder’s current time-step

Table 1: Masked outputs for different masking rate thresholds (MRT) and base similarity thresholds (ST)

| MRT | ST  | Masked Outputs                                      |
|-----|-----|-----------------------------------------------------|
| 0.2 | 0.4 | I love this restaurant! [mask] are comfortable and the [mask] is great! |
| 0.4 | 0.3 | I love this restaurant! [mask] are [mask] and [mask] is great! |
| 0.6 | 0.2 | [mask] this restaurant! [mask] are [mask] and [mask] is great! |
| 0.8 | 0.1 | [mask] restaurant! [mask] and [mask] great! |

Figure 2: ERM example with $S = i love this hotel! the beds are comfortable and the service is great!$ and RE = restaurant showing (a) Parse tree for RE; (b) and (c) Parse tree for S; (d) Semantic similarity heat map.
hidden state to calculate attention weights. We use the dot score function for attention, where \( h_t \) is the current target decoder state and \( \bar{h}_s \) is all encoder states: \[ \text{score}(h_t, \bar{h}_s) = h_t^T \bar{h}_s. \]

**Transformer**

Our second model makes use of the transformer architecture, and our implementation replicates Vaswani et al. (2017). We use an encoder-decoder structure with a multi-head self-attention token decoder to condition on information from both past and future context. It maps a query and set of key-value pairs to an output. The queries and keys are of dimension \( d_k \), and values of dimension \( d_v \). To compute the attention, we pack a set of queries, keys, and values into matrices \( Q, K, \) and \( V \), respectively. The matrix of outputs is computed as:

\[
\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V
\]

Multi-head attention allows the model to jointly attend to information from different positions. The decoder can make use of both local and global semantic information while filling in each [mask].

**4 Experiment**

**4.1 Datasets**

We train our two TIMs on the three datasets. The Amazon dataset (He and McAuley, 2016) contains over 83 million user reviews on products, with duplicate reviews removed. The Yelp dataset includes over six million user reviews on businesses. The news headlines dataset from Kaggle contains approximately 200,000 news headlines from 2012 to 2018 obtained from HuffPost (Misra, 2018).

We filter the text to obtain reviews and headlines which are English, do not contain hyperlinks and other obvious noise, and are less than 20 words long. We found that many longer than twenty words ramble on and are too verbose for our purposes. Rather than filtering by individual sentences we keep each text in its entirety so SMERTI can learn to generate multiple sentences at once. We preprocess the text by lowercasing and removing rare/duplicate punctuation and space.

For Amazon and Yelp, we treat reviews greater than three stars as containing positive sentiment, equal to three stars as neutral, and less than three stars as negative. For each training and testing set, we include an equal number of randomly selected positive and negative reviews, and half as many neutral reviews. This is because neutral reviews only occupy one out of five stars compared to positive and negative which occupy two each. Our dataset statistics can be found in Appendix B.

**4.2 Experiment Details**

To set up our training and testing data for text-in-filling, we mask the text. We use a tiered masking approach: for each dataset, we randomly mask 15% of the words in one-third of the lines, 30% of the words in another one-third, and 45% in the remaining one-third. These masked texts serve as the inputs, while the original texts serve as the ground-truth. This allows our TIM models to learn relationships between masked words and relationships between masked and unmasked words.

For the transformer, we use scaled dot-product attention and the same hyperparameters as Vaswani et al. (2017). We use the Adam optimizer (Kingma and Ba, 2014) with \( \beta_1 = 0.9, \beta_2 = 0.98, \) and \( \epsilon = 10^{-9} \). As in Vaswani et al. (2017), we increase the learning rate linearly for the first \( \text{warmup steps} \) training steps, and then decrease the learning rate proportionally to the inverse square root of the step number. We set \( \text{factor} = 1 \) and use \( \text{warmup steps} = 2000 \). We use a batch size of 4096, and we train for up to 40 epochs.

**4.3 Baseline Models**

We implement three models to benchmark against.\(^{3}\) First is NWN-STEM (Algorithm 2 from Yao et al. (2017)). We use the training sets as the “reference review sets” to extract similar nouns to the \( RE \) (using \( \text{MIN}_{\text{sim}} = 0.1 \)). We then replace nouns in the text similar to the \( RE \) with nouns extracted from the associated reference review set.

Secondly, we modify NWN-STEM to work for verbs and adjectives\(^{4}\), and call this GWN-STEM. From the reference review sets, we extract similar nouns, verbs, and adjectives to the \( RE \) (using

---

\(^{3}\)See Appendix C for more implementation details

\(^{4}\)WordNet can only work for single words (and not phrases). Also, it turns out that it cannot work for most adjective \( REs \), as discussed in Appendix C
MIN\textsubscript{sim} = 0.1), where the RE is now not restricted to being a noun. We replace nouns, verbs, and adjectives in the text similar to the RE with those extracted from the associated reference review set.

Lastly, we implement W2V-STEM using GenSim (Rehurek and Sojka, 2010). We train uni-gram Word2Vec models for single word REs, and four-gram models for phrases. Models are trained on the training sets. We use cosine similarity to determine the most similar word/phrase in the input text to \( R_E \), which is the replaced \( O_E \). For all other words/phrases, we calculate \( w'_i = w_i - w_{OE} + w_{RE} \), where \( w_i \) is the original word/phrase’s embedding vector, \( w_{OE} \) is the \( O_E \)’s, \( w_{RE} \) is the \( RE \)’s, and \( w'_i \) is the resulting embedding vector. The replacement word/phrase is \( w'_i \)’s nearest neighbour. We use similarity thresholds to adjust replacement rates (RR) and produce text under various replacement rate thresholds (RRT).

5 Evaluation

5.1 Evaluation Setup

We manually select 10 nouns, 10 verbs, 10 adjectives, and 5 phrases from the top 10% most frequent words/phrases in each test set as our evaluation REs. We filter the verbs and adjectives through a list of sentiment words (Hu and Liu, 2004) to ensure we do not choose REs that would obviously significantly alter the text’s sentiment.\(^5\)

For each evaluation RE, we choose one-hundred lines from the corresponding test set that does not already contain \( RE \). We choose lines with at least five words, as many with less carry little semantic meaning (e.g. ‘Great!’, ‘It is okay’). For Amazon and Yelp, we choose 50 positive and 50 negative lines per \( RE \).\(^6\) We repeat this process three times, resulting in three sets of 1000 lines per dataset per POS (excluding phrases), and three sets of 500 lines per dataset for phrases. Our final results are averaged metrics over these three sets.

For SMERTI-Transformer, SMERTI-RNN, and W2V-STEM, we generate four outputs per text for MRT/RRT of 20%, 40%, 60%, and 80%, which represent upper-bounds on the percentage of the input that can be masked and/or replaced. Note that NWN-STEM and GWN-STEM can only evaluate on limited POS and their maximum replacement rates are limited.\(^7\) We select \( MIN_{sim} \) values of 0.075 and 0 for nouns and 0.1 and 0 for verbs, as these result in replacement rates approximately equal to the actual MR/RR of the other models’ outputs for 20% and 40% MRT/RRT, respectively.

5.2 Key Evaluation Metrics

Fluency (SLOR) We use syntactic log-odds ratio (SLOR) (Kann et al., 2018) for sentence level fluency and modify from their word-level formula to character-level (\( SLOR_c \)). We use VADER (Hutto and Gilbert, 2014) to evaluate sentiment as positive, negative, or neutral. It handles typos, emojis, and other aspects of online text.

\[
SLOR_c(S) = \frac{1}{|S|} \left( \ln(p_M(S)) - \frac{\ln(\prod_{w \in S} p_M(w))}{\sum_{w \in S} |w|} \right) = -\ln(PPL_w) + \frac{\sum_{w \in S} |w| \ln(PPL_w)}{\sum_{w \in S} |w|} \tag{2}
\]

where \(|S|\) and \(|w|\) are the character lengths of the input text \( S \) and the word \( w \), respectively, \( p_M(S) \) and \( p_M(w) \) are the probabilities of \( S \) and \( w \) under the language model \( M \), respectively, and \( PPL_S \) and \( PPL_w \) are the character-level perplexities of \( S \) and \( w \), respectively. SLOR (from hereon we refer to character-level SLOR as simply SLOR) measures aspects of text fluency such as grammaticality. Higher values represent higher fluency.

We rescale resulting SLOR values to the interval [0,1] by first fitting and normalizing a Gaussian distribution. We then truncate normalized data points outside [-3,3], which shifts approximately 0.69% of total data. Finally, we divide each data point by six and add 0.5 to each result.

Sentiment Preservation Accuracy (SPA) is defined as the percentage of outputs that carry the same sentiment as the input. We use VADER (Hutto and Gilbert, 2014) to evaluate sentiment as positive, negative, or neutral. It handles typos, emojis, and other aspects of online text.

Content Similarity Score (CSS) ranges from 0 to 1 and indicates the semantic similarity between generated text and the \( RE \). A value closer to 1 indicates stronger semantic exchange, as the output is closer in semantic content to the \( RE \). We also use the USE for this due to its design and strong performance as previously mentioned.

5.3 Semantic Text Exchange Score (STES)

We come up with a single score to evaluate overall performance of a model on STE that combines...
Table 2: Overall average results by model (with % changes from the input)

| Model   | SPA   | SLOR  | CSS   | STES  |
|---------|-------|-------|-------|-------|
| Input   | 53.962| 53.258| 29.437| 43.37 |
| SMERTI-Transformer | 0.6066| 0.5258| 0.2943| 0.4237|
| SMERTI-NN | 0.6574| 0.5122| 0.2927| 0.4254|
| W2V-STEM | 0.6667| 0.6072| 0.2891| 0.4197|
| GWN-STEM | 0.8993| 0.8484| 0.1419| 0.2994|
| NWN-STEM | 0.9116| 0.8332| 0.1335| 0.2814|

The key evaluation metrics. It uses the harmonic mean, similar to the F₁ score (or F-score) (Chinchor, 1992; Rijksbergen, 1979), and we call it the Semantic Text Exchange Score (STES):

\[
STES = \frac{3 \times A \times B \times C}{A \times B + A \times C + B \times C}
\]  

where \( A \) is SPA, \( B \) is SLOR, and \( C \) is CSS. STES ranges between 0 and 1, with scores closer to 1 representing higher overall performance. Like the \( F_1 \) score, STES penalizes models which perform very poorly in one or more metrics, and favors balanced models achieving strong results in all three.

5.4 Automatic Evaluation Results

Table 2 shows overall average results by model. As observed from Table 3 (see also Appendix F), SMERTI is able to generate high quality output text similar to the \( RE \) while flowing better than other models’ outputs. It can replace entire phrases and sentences due to its variable length infilling. Note that for nouns, the outputs from GWN-STEM and NWN-STEM are equivalent.

5.5 Human Evaluation Setup

We conduct a human evaluation with eight participants, 6 males and 2 females, that are affiliated project researchers aged 20-39 at the University of Waterloo. We randomly choose one evaluation line for a randomly selected word or phrase for each POS per dataset. The input text and each model’s output (for 40% MRT/RRT - chosen as a good middle ground) for each line is presented to participants, resulting in a total of 54 pieces of text, and rated on the following criteria from 1-5:

- **RE Match**: “How related is the entire text to the concept of \([X]\)?”, where \([X]\) is a word or phrase (1 - not at all related, 3 - somewhat related, 5 - very related). Note here that \([X]\) is a given RE.
- **Fluency**: “Does the text make sense and flow well?” (1 - not at all, 3 - somewhat, 5 - very)
- **Sentiment**: “How do you think the author of the text was feeling?” (1 - very negative, 3 - neutral, 5 - very positive)

Each participant evaluates every piece of text. They are presented with a single piece of text at a time, with the order of models, POS, and datasets completely randomized.

5.6 Human Evaluation Results

Average human evaluation scores are displayed in Table 4. **Sentiment Preservation** (between 0 and 1) is calculated by comparing the average Sentiment rating for each model’s output text to the Sentiment rating of the input text, and if both are less than 2.5 (negative), between 2.5 and 3.5 inclusive (neutral), or greater than 3.5 (positive), this is counted as a valid case of Sentiment Preservation. We repeat this for every evaluation line to calculate the final values per model. Harmonic means of all three metrics (using rescaled 0-1 values of RE Match and Fluency) are also displayed.

6 Analysis

6.1 Performance by Model

As seen in Table 2, both SMERTI variations achieve higher STES and outperform the other models overall, with the WordNet models performing the worst. SMERTI excels especially on fluency and content similarity. The transformer variation achieves slightly higher SLOR, while the RNN variation achieves slightly higher CSS. The WordNet models perform strongest in sentiment preservation (SPA), likely because they modify little of the text and only verbs and nouns. They achieve by far the lowest CSS, likely in part due to this limited text replacement. They also do not account for context, and many words (e.g. proper nouns) do not exist in WordNet. Overall, the WordNet models are not very effective at STE.

The WordNet models perform strongest in sentiment preservation (SPA), likely because they modify little of the text and only verbs and nouns. They achieve by far the lowest CSS, likely in part due to this limited text replacement. They also do not account for context, and many words (e.g. proper nouns) do not exist in WordNet. Overall, the WordNet models are not very effective at STE.

W2V-STEM achieves the lowest SLOR, especially for higher RRT, as supported by the example in Table 3 (see also Appendix F). W2V-STEM and WordNet models output grammatically incorrect text that flows poorly. In many cases, words are
Input text: great food, large portions! my family and i really enjoyed our saturday morning breakfast.
Replacement entity: pizza

Table 3: Generated output text by model for various masking rates on a Yelp evaluation example

| Model          | Input Text | SMERTI-Transformer | SMERTI-RNN | W2V-STEM | GWN/NWN-STEM |
|----------------|------------|--------------------|------------|----------|--------------|
| RE Match (1-5) | 1.828      | 3.503              | 3.500      | 3.4792   | 2.2590       |
| Fluency (1-5)  | 4.1250     | 2.8750             | 2.8230     | 2.0830   | 2.5000       |
| Sentiment Preservation (0-1) | 0.7500 | 0.8333 | 0.6677 | 0.8333 | 1.0000 |
| Harmonic Mean (0-1) | 0.5098 | 0.5446 | 0.4408 | 0.4245 | 0.4547 |

Table 4: Average human evaluation scores by model

| POS          | Nouns | Verbs | Adjectives | Phrases |
|--------------|-------|-------|------------|---------|
| Input SLOR   | 0.5971| 0.5960| 0.5977     | 0.5939  |
| Input CSS    | 0.1175| 0.1287| 0.0935     | 0.1267  |
| SMERTI SPA   | 0.7118| 0.6586| 0.6195     | 0.6460  |
| SMERTI SJOR  | 0.5100| 0.5078| 0.4963     | 0.5613  |
| SMERTI CSS   | 0.2982| 0.2579| 0.2434     | 0.3572  |
| SMERTI STES  | 0.4464| 0.4073| 0.3876     | 0.4895  |

Table 5: Average TTR values by model

| Model | Input Text | SMERTI-Transformer | SMERTI-RNN | W2V-STEM | GWN/NWN-STEM |
|-------|------------|--------------------|------------|----------|--------------|
| Avg TTR | 0.9488 | 0.9429 | 0.9367 | 0.9025 | 0.9042 | 0.8950 |

6.2 Performance By Model - Human Results

As seen in Table 4, the SMERTI variations outperform all baseline models overall, particularly in RE Match. SMERTI-Transformer performs the best, with SMERTI-RNN second. The WordNet models achieve high Sentiment Preservation, but much lower on RE Match. W2V-STEM achieves comparably high RE Match, but lowest Fluency.

These results correspond well with our automatic evaluation results in Table 2. We look at the Pearson correlation values between RE Match, Fluency, and Sentiment Preservation with CSS, SLOR, and SPA, respectively. These are 0.9952, 0.9327, and 0.8768, respectively, demonstrating that our automatic metrics are highly effective and correspond well with human ratings.

6.3 SMERTI’s Performance By POS

As seen from Table 6, SMERTI’s SPA values are highest for nouns, likely because they typically carry little sentiment, and lowest for adjectives, likely because they typically carry the most.

SLOR is lowest for adjectives and highest for phrases and nouns. Adjectives typically carry less semantic meaning and SMERTI likely has more trouble figuring out how best to infill the text. In contrast, nouns typically carry more, and phrases the most (since they consist of multiple words).

SMERTI’s CSS is highest for phrases then nouns, likely due to phrases and nouns carrying

\[ \text{Note that the SMERTI values in Tables 6 to 8 refer to the average between SMERTI-Transformer and SMERTI-RNN} \]
more semantic meaning, making it easier to generate semantically similar text. Both SMERTI’s and the input text’s CSS are lowest for adjectives, likely because they carry little semantic meaning.

Overall, SMERTI appears to be more effective on nouns and phrases than verbs and adjectives.

### 6.4 SMERTI’s Performance By Dataset

As seen in Table 7, SMERTI’s SPA is lowest for news headlines. Amazon and Yelp reviews naturally carry stronger sentiment, likely making it easier to generate text with similar sentiment.

Both SMERTI’s and the input text’s SLO are lower for Yelp reviews. This may be due to many reasons, such as more typos and emojis within the original reviews, and so forth.

SMERTI’s CSS values are slightly higher for news headlines. This may be due to them typically being shorter and carrying more semantic meaning as they are designed to be attention grabbers.

Overall, it seems that using datasets which inherently carry more sentiment will lead to better sentiment preservation. Further, the quality of the dataset’s original text, unsurprisingly, influences the ability of SMERTI to generate fluent text.

### 6.5 SMERTI’s Performance By MRT/RRT

From Table 8, it can be seen that as MRT/RRT increases, SMERTI’s SPA and SLO decrease while CSS increases. These relationships are very strong as supported by the Pearson correlation values of -0.9972, -0.9183, and 0.9078, respectively. When SMERTI can alter more text, it has the opportunity to replace more related to sentiment while producing more of semantic similarity to the RE.

Further, SMERTI generates more of the text itself, becoming less similar to the human-written input, resulting in lower fluency. To further demonstrate this, we look at average SMERTI BLEU (Papineni et al., 2002) scores against MRT/RRT, shown in Table 8. BLEU generally indicates how close two pieces of text are in content and structure, with higher values indicating greater similarity. We report our final BLEU scores as the average scores of 1 to 4-grams. As expected, BLEU decreases as MRT/RRT increases, and this relationship is very strong as supported by the Pearson correlation value of -0.9960.

It is clear that MRT/RRT represents a trade-off between CSS against SPA and SLO. It is thus an adjustable parameter that can be used to control the generated text, and balance semantic exchange against fluency and sentiment preservation.

### 7 Conclusion and Future Work

We introduced the task of semantic text exchange (STE), demonstrated that our pipeline SMERTI performs well on STE, and proposed an STES metric for evaluating overall STE performance. SMERTI outperformed other models and was the most balanced overall. We also showed a trade-off between semantic exchange against fluency and sentiment preservation, which can be controlled by the masking (replacement) rate threshold.

Potential directions for future work include adding specific methods to control sentiment, and fine-tuning SMERTI for preservation of persona or personality. Experimenting with other text infilling models (e.g. fine-tuning BERT (Devlin et al., 2019)) is also an area of exploration. Lastly, our human evaluation is limited in size and a larger and more diverse participant pool is needed.

We conclude by addressing potential ethical misuses of STE, including assisting in the generation of spam and fake-reviews/news. These risks come with any intelligent chatbot work, but we feel that the benefits, including usage in the detection of misuse such as fake-news, greatly outweigh the risks and help progress NLP and AI research.

### Acknowledgments

We thank our anonymous reviewers, study participants, and Huawei Technologies Co., Ltd. for financial support.
References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

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

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhtoni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Philipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. CoRR, abs/1312.3005.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 740–750.

Nancy Chinchor. 1992. Muc-4 evaluation metrics. In Proceedings of the 4th conference on Message understanding, pages 22–29. Association for Computational Linguistics.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Shaoqian Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 623–632.

William Fedus, Ian J. Goodfellow, and Andrew M. Dai. 2018. Maskgan: Better text generation via filling in the ______. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In Thirty-Second AAAI Conference on Artificial Intelligence.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th international conference on world wide web, pages 507–517. International World Wide Web Conferences Steering Committee.

Dirk Hovy. 2016. The enemy in your own camp: How well can we detect statistically-generated fake reviews–an adversarial study. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), volume 2, pages 351–356.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177. ACM.

Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eighth international AAAI conference on weblogs and social media.

Mika Juuti, Bo Sun, Tatsuya Mori, and N Asokan. 2018. Stay on-topic: Generating context-specific fake restaurant reviews. In European Symposium on Research in Computer Security, pages 132–151. Springer.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: References help, but can be spared! In Proceedings of the 22nd Conference on Computational Natural Language Learning, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. International Conference on Learning Representations.
Juncen Li, Robin Jia, He He, and Percy Liang. 2018. 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.

Zachary Chase Lipton, Sharad Vikram, and Julian J. McAuley. 2015. Capturing meaning in product reviews with character-level generative text models. ArXiv, abs/1511.03683.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41.

Rishabh Misra. 2018. News headlines dataset for sarcasm detection. https://rishabhmisra.github.io/publications/.

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 on Association for Computational Linguistics, ACL ’02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. arXiv preprint arXiv:1806.06259.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

C. J. Van Rijsbergen. 1979. Information Retrieval, 2nd edition. Butterworth-Heinemann, Newton, MA, USA.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. 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.

Zachary Chase Lipton, Sharad Vikram, and Julian J. McAuley. 2015. Capturing meaning in product reviews with character-level generative text models. ArXiv, abs/1511.03683.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41.

Rishabh Misra. 2018. News headlines dataset for sarcasm detection. https://rishabhmisra.github.io/publications/.

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 on Association for Computational Linguistics, ACL ’02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian S Perone, Roberto Silveira, and Thomas S Paula. 2018. Evaluation of sentence embeddings in downstream and linguistic probing tasks. arXiv preprint arXiv:1806.06259.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta. ELRA. http://is.muni.cz/publication/884893/en.

C. J. Van Rijsbergen. 1979. Information Retrieval, 2nd edition. Butterworth-Heinemann, Newton, MA, USA.
Appendices for “Keep Calm and Switch On! Preserving Sentiment and Fluency in Semantic Text Exchange”

Steven Y. Feng∗ Aaron W. Li∗ Jesse Hoey

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
{sy2feng, w89li, jhoey}@uwaterloo.ca

A Masked Output Examples

Table1 includes various example masked outputs by the ERM and SMM modules. We illustrate a variety of word-to-word, phrase-to-word, word-to-phrase, and phrase-to-phrase entity replacements and similarity masking.

B Dataset Statistics

See Table 2 for our training and testing splits by dataset and sentiment.

C Details of Baseline Implementations

C.1 NWN-STEM

This follows Algorithm 2 in Yao et al. (2017). For each dataset and each RE (note that this model is restricted to noun REs), we go through the dataset’s training set (which acts as the “reference review set”) and extract a list of text lines that contain the RE (where the RE acts as the “topic keyword”). For each of these lines, with the help of the Stanford Parser, we extract all single-word nouns, and for each of them (which we call nouni), we check if MINsim(nouni, RE) > 0.1. If so, we add them to the list of nouns similar to the RE, which we call simnouns.

For each evaluation line and associated RE, we extract all singular nouns within the text that are similar to the RE. For our evaluation purposes, we choose two MINsim values of 0.075 and 0 to produce two outputs per input. These two MINsim values result in actual replacement rates similar to the actual masking/replacement rates of other models (SMERTI and W2V-STEM) for MRT/RRT of 20% and 40%, respectively. Each similar noun is replaced with the noun in simnouns that is most similar to it to produce the output text.

We noticed that GWN-STEM only works for noun and verb REs, as for most adjectives, WordNet cannot calculate similarity scores. Hence, it was infeasible to evaluate on adjective REs. Further, most similarity scores only exist between noun-noun pairs and verb-verb pairs, and when we

∗ Authors contributed equally

1 Tables and figures mentioned in this appendices document refer to the tables and figures here
| S | family enjoyed the food quite a bit especially the sweet and sour chicken. |
|---|---|
| RE | bitter |
| S′_1 | family enjoyed the food quite a bit especially the sweet and bitter chicken. |
| S′_2 | family [mask] the [mask] a bit especially the sweet and bitter [mask]. |
| S′_3 | [mask] the [mask] bit [mask] the [mask] and bitter [mask]. |
| S′_4 | [mask] bit [mask] bitter [mask]. |
| S | terrible customer service. couldn’t make a wire transfer because they are out of paper. |
| RE | amazing |
| S′_1 | amazing customer [mask]. couldn’t make a wire transfer [mask] they are out of paper. |
| S′_2 | amazing [mask]. [mask] make a wire transfer [mask] out [mask] paper. |
| S′_3 | amazing [mask]. [mask] a [mask] out [mask]. |
| S′_4 | amazing [mask]. [mask] a [mask] out [mask]. |
| S | Heather enjoyed her movie date with Jim last night. |
| RE | Yesterday |
| S′_1 | heather [mask] her movie [mask] with jim yesterday. |
| S′_2 | heather [mask] her movie [mask] with jim yesterday. |
| S′_3 | heather [mask] her movie [mask] with jim yesterday. |
| S′_4 | heather [mask] yesterday. |
| S | My son took his math test yesterday and failed. He cried all day and I hate him now. |
| RE | Medical examination |
| S′_1 | [mask] took medical examination yesterday and [mask]. he cried all day and i hate [mask] now. |
| S′_2 | [mask] took medical examination yesterday and [mask]. he cried all day and i hate [mask] now. |
| S′_3 | [mask] took medical examination yesterday and [mask]. he cried all day and i hate [mask] now. |
| S′_4 | [mask] medical examination [mask] and [mask]. [mask]. |
| S | The car crashed into the building and exploded, killing hundreds. |
| RE | Caught on fire |
| S′_1 | the car [mask] caught on fire and [mask], killing hundreds. |
| S′_2 | [mask] caught on fire and [mask], killing hundreds. |
| S′_3 | [mask] caught on fire and [mask], killing hundreds. |
| S′_4 | [mask] caught on fire [mask]. |
| S | I took my dog for a walk in the park. He really enjoyed it! |
| RE | The river |
| S′_1 | i took my dog for [mask] in the river . he really enjoyed it ! |
| S′_2 | i [mask] for [mask] in the river . he really enjoyed it ! |
| S′_3 | i [mask] for [mask] in the river . he [mask] ! |
| S′_4 | i [mask] the river . he [mask] ! |
| S | It is very sunny outside so I am very sweaty. |
| RE | Extremely snowy |
| S′_1 | it is extremely snowy outside so i am [mask]. |
| S′_2 | it is extremely snowy [mask] so i am [mask]. |
| S′_3 | it [mask] extremely snowy [mask] [mask]. |
| S′_4 | [mask] extremely snowy [mask]. |
| S | I went to my friend Amy’s house last night. |
| RE | My husband Jim’s |
| S′_1 | i went to my husband jim’s house last night. |
| S′_2 | i went to my husband jim’s house [mask]. |
| S′_3 | i went to my husband jim’s [mask]. |
| S′_4 | i [mask] my husband jim’s [mask]. |

| S′_1 | i went to my husband jim’s boat last night. |
| S′_2 | i went to my husband jim’s boat last night. |
| S′_3 | i went to my husband jim’s boat [mask]. |
| S′_4 | i [mask] my husband jim’s boat [mask]. |

Table 1: Example masked outputs. S is the original input text; RE is the replacement entity; S′_1 corresponds to \( MRT = 0.2 \), base \( ST = 0.4 \); S′_2 corresponds to \( MRT = 0.4 \), base \( ST = 0.3 \); S′_3 corresponds to \( MRT = 0.6 \), base \( ST = 0.2 \); S′_4 corresponds to \( MRT = 0.8 \), base \( ST = 0.1 \).
tried to produce $\text{sim}_{\text{verbs}}$ and $\text{sim}_{\text{adjs}}$ for noun $RE$s, almost all resulted in empty lists. Hence, GWN-STEM actually produced the same outputs as NWN-STEM for noun $RE$s (and was also limited to two replacement rates), which is why the two models have the same outputs and resulting metrics for noun $RE$s. Even for verb $RE$s, we were limited to two sets of outputs (mimicking the two replacement rates above) since similarity calculations between verb-noun pairs and verb-adjective pairs were limited, so few were replaced.

### C.3 W2V-STEM

This uses Word2Vec (W2V) models trained using Gensim. We train six W2V models: one unigram model per dataset, and one four-gram model per dataset, where each is trained using the corresponding dataset’s training set. To train the four-gram models, we begin by applying a bi-gram phrasing model on top of the original text, and then the bi-gram phrasing model again on top of this resulting text. We call this a four-gram phrasing model. We then use this to generate text that is grouped into phrases up to four-grams long. We then train W2V models on this four-gram text to generate the four-gram W2V models.

For the unigram models, we use an embedding vector size of 50, a context window of 3, a minimum token count of 0, and the skip-gram model. For the four-gram models, we use an embedding vector size of 10, a context window of 1, a minimum token count of 0, and the CBOW (continuous bag-of-words) model.

For evaluation lines with noun, verb, and adjective $RE$s, we go through the line of text and with the help of the Stanford parser, extract all words that are the same POS as the $RE$ (which become the candidate $OE$s). Then, using cosine similarity between the W2V embedding vectors (from the unigram W2V models) of the $RE$ and candidate $OE$s, we find the word with the maximum similarity to the $RE$ which becomes the replaced $OE$. Then, we replace other words in the input text similar to the $OE$ using vector addition and subtraction as described in Section 4.3. We start with a similarity threshold value of 0.1 that increases by 0.05 each time to generate text satisfying varying replacement rate thresholds.

For evaluation lines with phrase $RE$s, we first generate text files of the evaluation sets that are grouped into phrases up to four-grams long using the four-gram phrasing model. Then, using the four-gram W2V models, we proceed similarly as above to determine the replaced $OE$, which can now be either a single word or a phrase. Other words and phrases in the input text are replaced using vector addition and subtraction. We start with a similarity threshold value of 0.3 that increases by 0.01 each time to generate text satisfying varying replacement rate thresholds.

### D Evaluation Keywords/Phrases

We select ten keywords (five keyphrases) to act as our $RE$s per dataset per POS. See Tables 3 to 6 for the chosen $RE$s. To do so, we iterate through our test sets and with the Stanford Parser, we extract a list of nouns and noun phrases, verbs and verb phrases, and adjectives and adjective phrases. We sort these lists by frequency, and limit our selections to the top 10% most frequent. For the verbs and adjectives and their phrases, we further filter them through a list of sentiment words (Hu and Liu, 2004) to ensure the $RE$s we choose do not carry significant sentiment-related meaning, as inserting them into the original text would obviously lead to major changes in sentiment. From these, we manually select ten per dataset per POS (except phrases, where we select five per dataset) that are significant and carry strong meaning. These work well as the $RE$s for evaluation purposes. Note that for phrases, we choose three noun phrases, one verb phrase, and one adjective phrase per dataset.

We choose from the most frequent words and phrases as they are more common and likely hold more significant meaning compared to less frequent ones (e.g. names and typos). Manual selection was required as some of the most frequent words/phrases hold little semantic meaning (e.g. it, they, is, was, and so forth). We only choose half the number of phrases as words as we find that frequent phrases carrying significant semantic meaning with little sentiment are much rarer.

| Dataset       | Sentiment | Training Set | Testing Set |
|---------------|-----------|--------------|-------------|
| Amazon        | Positive  | 30K          | 5K          |
|               | Negative  | 30K          | 5K          |
|               | Neutral   | 15K          | 2.5K        |
| Yelp          | Positive  | 30K          | 5K          |
|               | Negative  | 30K          | 5K          |
|               | Neutral   | 15K          | 2.5K        |
| News Headlines|           | 120K         | 20K         |
Table 3: Chosen evaluation noun REs. *Obama does not exist in WordNet, so we instead use the word President for NWN-STEM and GWN-STEM.

| Yelp  | Amazon  | News Headlines |
|-------|---------|----------------|
| Food  | Book    | Trump          |
| Service | Product | Photos        |
| Place | Time   | Video          |
| Staff | Price  | World          |
| Time | Quality | Women          |
| Customer | Money | Life          |
| Atmosphere | Game | Kids          |
| Pizza | Story  | People         |
| Restaurant | Movie | Week        |
| Chicken | Phone  | Obama / President* |

Table 4: Chosen evaluation verb REs

| Yelp  | Amazon  | News Headlines |
|-------|---------|----------------|
| Ordered | Buy | Live          |
| Closed | Ordered | Think        |
| Tasted | Received | Make        |
| Return | Expected | Stop         |
| Waiting | Purchased | Watch   |
| Serve | Reading | Save         |
| Eating | Advertised | Avoid      |
| Visiting | Install | Learn        |
| Looking | Playing | Create       |
| Priced | Recommend | Teach    |

Table 5: Chosen evaluation adjective REs

| Yelp  | Amazon  | News Headlines |
|-------|---------|----------------|
| Small | Different | Black         |
| Little | Light | White         |
| New | Predictable | Old         |
| Big | Little | American      |
| Mexican | Heavy | National      |
| First | Thick | Sexual        |
| Long | Plastic | Single      |
| High | Old | Global         |
| Open | New | Presidential  |
| Busy | First | Female        |

Table 6: Chosen evaluation phrase REs

| Yelp  | Amazon  | News Headlines |
|-------|---------|----------------|
| Customer Service | My Daughter | Donald Trump |
| Your Money | Another One | Hillary Clinton |
| Las Vegas | This Game | Climate Change |
| Come Back | Put it Down | Need to Know |
| Very Small | Much Smaller | Most Important |

E Detailed Evaluation Results - Tables and Graphs

See Figure 1 for a graph of overall average results, Table 7 and Figure 2 for average results by POS, Table 8 and Figure 3 for average results by dataset, and Table 9 and Figure 4 for average results by MRT/RRT. Note that the bolded values in the tables show which model performs better on that particular metric, on average, for the category.

![Overall Average Results](image)

Table 7: Overall average results (referring to the data found in Table 2 of the main body)

F Model Output Examples

See Tables 10 to 21 for example outputs from every model for all datasets and POS.

References

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177. ACM.

Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y Zhao. 2017. Automated crowd-turfing attacks and defenses in online review systems. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 1143–1158. ACM.
| Model      | Nouns | Verbs | Adjectives | Phrases |
|------------|-------|-------|------------|---------|
|            | MRT/ RRT | SPA | SLOR | CSS | SPA | SLOR | CSS | SPA | SLOR | CSS | SPA | SLOR | CSS |
| Input      |       | ---  | 0.5971 | 0.1175 | ---  | 0.5960 | 0.1287 | ---  | 0.5977 | 0.0935 | ---  | 0.5939 | 0.1267 |
| SMERTI-Transformer | 20%  | 0.8560 | 0.5493 | 0.2591 | 0.7887 | 0.5358 | 0.2152 | 0.7517 | 0.5421 | 0.2003 | 0.7971 | 0.5806 | 0.3028 |
|            | 40%  | 0.7610 | 0.5382 | 0.2683 | 0.7143 | 0.5286 | 0.2215 | 0.6688 | 0.5326 | 0.2059 | 0.7060 | 0.5753 | 0.3178 |
|            | 60%  | 0.6707 | 0.5200 | 0.2899 | 0.6328 | 0.5193 | 0.2461 | 0.5798 | 0.5091 | 0.2286 | 0.6016 | 0.5661 | 0.3516 |
|            | 80%  | 0.5638 | 0.4592 | 0.3613 | 0.5003 | 0.4783 | 0.3409 | 0.4832 | 0.4360 | 0.3222 | 0.4933 | 0.5379 | 0.4395 |
|            | AVG  | 0.7129 | 0.5167 | 0.2947 | 0.6590 | 0.5155 | 0.2559 | 0.6209 | 0.5049 | 0.2592 | 0.6495 | 0.5650 | 0.3529 |
| STES       | 0.4456 | 0.4073 | 0.3861 | 0.4884 |
| SMERTI-RNN | 20%  | 0.8568 | 0.5443 | 0.2620 | 0.7946 | 0.5299 | 0.2170 | 0.7576 | 0.5338 | 0.2048 | 0.7924 | 0.5751 | 0.3075 |
|            | 40%  | 0.7602 | 0.5273 | 0.2725 | 0.7204 | 0.5145 | 0.2243 | 0.6641 | 0.5161 | 0.2109 | 0.7040 | 0.5673 | 0.3242 |
|            | 60%  | 0.6688 | 0.5023 | 0.2979 | 0.6290 | 0.4967 | 0.2507 | 0.5792 | 0.4871 | 0.2365 | 0.5996 | 0.5565 | 0.3604 |
|            | 80%  | 0.5570 | 0.4397 | 0.3744 | 0.4839 | 0.4590 | 0.3477 | 0.4718 | 0.4139 | 0.3376 | 0.4736 | 0.5313 | 0.4541 |
|            | AVG  | 0.7107 | 0.5034 | 0.3017 | 0.6582 | 0.5000 | 0.2599 | 0.6182 | 0.4877 | 0.2474 | 0.6424 | 0.5576 | 0.3615 |
| STES       | 0.4472 | 0.4073 | 0.3891 | 0.4905 |
| W2V-STEM   | 20%  | 0.9038 | 0.5342 | 0.2804 | 0.8207 | 0.5140 | 0.2360 | 0.7724 | 0.5158 | 0.2251 | 0.8987 | 0.6078 | 0.2992 |
|            | 40%  | 0.7628 | 0.4734 | 0.3063 | 0.7160 | 0.4848 | 0.2694 | 0.6348 | 0.4525 | 0.2490 | 0.7693 | 0.5626 | 0.2954 |
|            | 60%  | 0.6354 | 0.4216 | 0.3232 | 0.6124 | 0.3941 | 0.2965 | 0.5108 | 0.4016 | 0.2668 | 0.6280 | 0.5237 | 0.2913 |
|            | 80%  | 0.5233 | 0.3864 | 0.3374 | 0.5168 | 0.3551 | 0.3170 | 0.4209 | 0.3702 | 0.2794 | 0.5411 | 0.5136 | 0.2892 |
|            | AVG  | 0.7063 | 0.4539 | 0.3118 | 0.6665 | 0.4279 | 0.2797 | 0.5847 | 0.4350 | 0.2551 | 0.7093 | 0.5519 | 0.2938 |
| STES       | 0.4395 | 0.4047 | 0.3784 | 0.4528 |
| GWN-STEM   | 20%  | 0.9291 | 0.5043 | 0.1288 | 0.9177 | 0.5421 | 0.1372 | 0.9177 | 0.5421 | 0.1372 | 0.9177 | 0.5421 | 0.1372 |
|            | 40%  | 0.8941 | 0.4761 | 0.1318 | 0.8263 | 0.4372 | 0.1635 | 0.8263 | 0.4372 | 0.1635 | 0.8263 | 0.4372 | 0.1635 |
|            | AVG  | 0.9116 | 0.4832 | 0.1335 | 0.8690 | 0.4896 | 0.1504 | 0.8690 | 0.4896 | 0.1504 | 0.8690 | 0.4896 | 0.1504 |
| STES       | 0.2814 | 0.3048 |
| NWN-STEM   | 20%  | 0.9291 | 0.5043 | 0.1288 |       |       |       |       |       |       |       |       |       |
|            | 40%  | 0.8941 | 0.4621 | 0.1301 |       |       |       |       |       |       |       |       |       |
|            | AVG  | 0.9116 | 0.4832 | 0.1335 |       |       |       |       |       |       |       |       |       |
| STES       | 0.2814 | 0.3048 |

Table 7: Average results by POS

Figure 2: Graph of average results by POS
## Table 8: Average results by dataset

| Model        | Amazon Reviews | Yelp Reviews | News Headlines |
|--------------|----------------|--------------|----------------|
|              | SPA | SLOR | CSS | SPA | SLOR | CSS | SPA | SLOR | CSS |
| Input        | --- | 0.6230 | 0.1113 | --- | 0.5532 | 0.1429 | --- | 0.6123 | 0.0955 |
| SMERTI-Transformer | 20% | 0.8140 | 0.5773 | 0.2336 | 0.8544 | 0.5162 | 0.2497 | 0.7267 | 0.5623 | 0.2497 |
|              | 40% | 0.7168 | 0.5709 | 0.2410 | 0.7800 | 0.5055 | 0.2577 | 0.6408 | 0.5547 | 0.2615 |
|              | 60% | 0.6363 | 0.5550 | 0.2663 | 0.6981 | 0.4804 | 0.2759 | 0.5293 | 0.5503 | 0.2950 |
|              | 80% | 0.5408 | 0.4776 | 0.3598 | 0.5781 | 0.4045 | 0.3370 | 0.4116 | 0.5114 | 0.4011 |
| AVG          | 0.6770 | **0.5452** | 0.2752 | 0.7276 | **0.4767** | 0.2801 | 0.5771 | **0.5547** | 0.3018 |
| STES         | 0.4319 |    |    | 0.4260 |    |    | 0.4380 |    |
| SMERTI-RNN   | 20% | 0.8113 | 0.5779 | 0.2354 | 0.8488 | 0.5174 | 0.2522 | 0.7409 | 0.5420 | 0.2559 |
|              | 40% | 0.7198 | 0.5634 | 0.2441 | 0.7733 | 0.5037 | 0.2621 | 0.6434 | 0.5268 | 0.2677 |
|              | 60% | 0.6318 | 0.5390 | 0.2714 | 0.6970 | 0.4786 | 0.2831 | 0.5286 | 0.5144 | 0.3047 |
|              | 80% | 0.5195 | 0.4636 | 0.3709 | 0.5608 | 0.4058 | 0.3505 | 0.4132 | 0.5136 | 0.4140 |
| AVG          | 0.6706 | 0.5360 | **0.2804** | 0.7200 | 0.4764 | 0.2870 | 0.5815 | 0.5242 | **0.3106** |
| STES         | **0.4333** |    |    | **0.4302** |    |    | **0.4382** |    |
| W2V-STEM     | 20% | 0.8628 | 0.5649 | 0.2526 | 0.8868 | 0.5158 | 0.2661 | 0.7971 | 0.5481 | 0.2619 |
|              | 40% | 0.7468 | 0.4902 | 0.2728 | 0.7675 | 0.4634 | 0.2865 | 0.6478 | 0.4990 | 0.2808 |
|              | 60% | 0.6201 | 0.4258 | 0.2867 | 0.6391 | 0.4167 | 0.3007 | 0.5308 | 0.4633 | 0.2960 |
|              | 80% | 0.5185 | 0.3898 | 0.2978 | 0.5316 | 0.3879 | 0.3118 | 0.4515 | 0.4413 | 0.3077 |
| AVG          | 0.6870 | 0.4677 | 0.2775 | 0.7063 | 0.4459 | **0.2912** | 0.6068 | 0.4880 | 0.2866 |
| STES         | 0.4168 |    |    | 0.4230 |    |    | 0.4175 |    |
| GWN-STEM     | 20% | 0.9605 | 0.5509 | 0.1232 | 0.9638 | 0.4924 | 0.1674 | 0.8458 | 0.5262 | 0.1085 |
|              | 40% | 0.9078 | 0.4649 | 0.1330 | 0.9347 | 0.4250 | 0.1783 | 0.7292 | 0.4591 | 0.1410 |
|              | 80% | 0.9342 | 0.5079 | 0.1281 | 0.9493 | 0.4587 | 0.1729 | 0.7875 | 0.4926 | 0.1248 |
| AVG          | 0.9455 | 0.5053 | 0.1112 | **0.9615** | 0.4517 | 0.1686 | **0.8278** | 0.4926 | 0.1206 |
| STES         | 0.2493 |    |    | 0.3266 |    |    | 0.2602 |    |

Table 8: Average results by dataset

![Graph of average results by dataset](image-url)

Figure 3: Graph of average results by dataset
| Model          | MRT/RRT | Actual MR/RR | SPA  | SLOR  | CSS   | STES  |
|---------------|---------|--------------|------|-------|-------|-------|
| Input         | ---     | ---          | 0.5962 | 0.1166 | ---   |       |
| SMERTI-Transformer | 20%     | 13.98%       | 0.7984 | 0.5519 | 0.2443 | 0.4192 |
|               | 40%     | 31.58%       | 0.7125 | 0.5437 | 0.2534 | 0.4173 |
|               | 60%     | 51.50%       | 0.6212 | 0.5286 | 0.2791 | 0.4234 |
|               | 80%     | 74.58%       | 0.5102 | 0.4778 | 0.3659 | 0.4421 |
|               | AVG     | 42.91%       | 0.6606 | **0.5255** | 0.2857 | 0.4337 |
| SMERTI-RNN    | 20%     | 13.99%       | 0.8003 | 0.5458 | 0.2478 | 0.4215 |
|               | 40%     | 31.61%       | 0.7122 | 0.5313 | 0.2580 | 0.4189 |
|               | 60%     | 51.55%       | 0.6191 | 0.5107 | 0.2864 | 0.4246 |
|               | 80%     | 74.60%       | 0.4978 | 0.4610 | 0.3785 | 0.4399 |
|               | AVG     | 42.94%       | 0.6574 | 0.5122 | **0.2927** | **0.4354** |
| W2V-STEM      | 20%     | 12.98%       | 0.8489 | 0.5429 | 0.2602 | 0.4371 |
|               | 40%     | 29.00%       | 0.7207 | 0.4842 | 0.2800 | 0.4271 |
|               | 60%     | 48.95%       | 0.5967 | 0.4353 | 0.2945 | 0.4071 |
|               | 80%     | 70.01%       | 0.5005 | 0.4063 | 0.3057 | 0.3881 |
|               | AVG     | 40.24%       | 0.6667 | 0.4672 | 0.2851 | 0.4197 |
| GWN-STEM      | 20%     | 16.52%       | 0.9234 | 0.5232 | 0.1330 | 0.2854 |
|               | 40%     | 33.85%       | 0.8572 | 0.4496 | 0.1508 | 0.2993 |
|               | AVG     | 25.18%       | 0.8903 | 0.4864 | 0.1419 | 0.2934 |
| NWN-STEM      | 20%     | 19.05%       | 0.9291 | 0.5043 | 0.1288 | 0.2772 |
|               | 40%     | 30.26%       | 0.8941 | 0.4621 | 0.1381 | 0.2851 |
|               | AVG     | 24.65%       | **0.9116** | 0.4832 | 0.1335 | 0.2814 |

Table 9: Average results by MRT/RRT

Figure 4: Graph of average results by MRT/RRT
| MRT/RRT                | Generated Output                                                                 |
|-----------------------|----------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                 |
| 20%                   | bought this in a few chapters and i love the book . highly recommend purchasing   |
|                       | this as well as the product .                                                   |
| 40%                   | bought this in my kindle and absolutely love the book . highly recommend         |
|                       | purchasing this as a gift as well .                                             |
| 60%                   | bought this in my kindle and absolutely love the book . highly recommend         |
|                       | purchasing this as a gift as well .                                             |
| 80%                   | bought this book in august and love the book . i highly recommend purchasing it . |
| **SMERTI-RNN**        |                                                                                 |
| 20%                   | bought this in a few months and i love the book . highly recommend purchasing    |
|                       | this as well as needed .                                                        |
| 40%                   | bought this in my style and i love the book . highly recommend purchasing this    |
|                       | as a gift as advertised .                                                        |
| 60%                   | bought this in my review and went through in a leg and in the book . i          |
|                       | recommend purchasing this .                                                      |
| 80%                   | bought this in my review and went through in a leg and in the book . i          |
|                       | recommend purchasing this .                                                      |
| **W2V-STEM**          |                                                                                 |
| 20%                   | bought this in a few item and simply love the book . highly recommend purchasing |
|                       | this as well as other item .                                                    |
| 40%                   | bought this in a few item application thing episode the book . highly recommend  |
|                       | purchasing this as well as other item .                                          |
| 60%                   | bought preview page a few item application thing episode page book . despite    |
|                       | recommend purchasing preview as fully as other item .                           |
| 80%                   | purchased preview page information few item application thing episode page book .|
|                       | despite recommend refunded preview possible fully possible reviews item .        |
| **GWN-STEM / NWN-STEM**|                                                                                 |
| 20% (0.075 $MIN_{im}$)| bought this in a few decal and simply love the style . highly recommend         |
|                       | purchasing this as well as other decal .                                         |
| 40% (0 $MIN_{im}$)    | bought movie in a few decal and simply love the locations . highly recommend     |
|                       | purchasing movie as well as other decal .                                       |

Table 10: Example outputs for an Amazon evaluation line with noun $RE$
| Input                                              | Generated Output                                                                 |
|---------------------------------------------------|----------------------------------------------------------------------------------|
| **Amazon - Verb RE**                               |                                                                                  |
| **Input**: i didn't even get these. and i think i ordered them like last year. and i think that is ridiculous! |
| **RE**: buy                                        |                                                                                  |
| **MRT/RRT**                                        |                                                                                  |
| **SMERTI-Transformer**                             |                                                                                  |
| 20% i don't even buy these. and i think i ordered a like last year. and i think that was ridiculous! |
| 40% do not buy these. and i ordered the wrong one for my year. and they are ridiculous! |
| 60% do not buy this product. i ordered another pair. it's ridiculous! |
| 80% don't buy this product. terrible screen protector. will not buy again! |
| **SMERTI-RNN**                                     |                                                                                  |
| 20% i don't even buy these. and i think i ordered more like last year. and i think that it's ridiculous! |
| 40% i would buy these. and i ordered them like a year. and from a ridiculous! |
| 60% don't buy this product. i ordered a x. w extremely ridiculous! |
| 80% don't buy this. a good rip. recommended!      |
| **W2V-STEM**                                       |                                                                                  |
| 20% i didn't even buy these. and i yes i ordered them like last year. and i yes that is ridiculous! |
| 40% purchasing didnat buyers buy these. definitely purchasing yes purchasing ordered them like last year. definitely purchasing yes that smells ridiculous! |
| 60% purchasing didn't buyers buy these. definitely purchasing yes purchasing bought theses like chinese year. definitely purchasing yes yes smells worthy! |
| 80% purchasing didn't buyers buy pair. definitely purchasing yes purchasing bought theses recomend chinese yrs. definitely purchasing yes yes smells worthy! |
| **GWN-STEM**                                       |                                                                                  |
| 20% (0.1 MINsim) i didn't even get these. and i think i ordered them like last yrs. and i think that was ridiculous! |
| 40% (0 MINsim) i didn't even get knowledge. and i think i requested knowledge like last yrs. and i think knowledge was ridiculous! |

Table 11: Example outputs for an Amazon evaluation line with verb *RE*
### Table 12: Example outputs for an Amazon evaluation line with adjective \textit{RE}

| MRT/RRT          | Generated Output                                                                 |
|------------------|----------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                 |
| 20%               | was not too predictable. i had ordered a wow, it probably was a waste of money.   |
| 40%               | was too predictable. i had ordered a wow, but the book probably was a waste of time. |
| 60%               | was too predictable. i had ordered a wow, but the book probably was a waste of time. |
| 80%               | easily predictable. i like a twist, it took a long time to finish.                |
| **SMERTI-RNN**    |                                                                                 |
| 20%               | was too predictable. i had ordered a refund, it probably was a little disappointing. |
| 40%               | was too very predictable. i had ordered a refund, but probably was a disappointing format. |
| 60%               | was too very predictable. i had ordered a refund, but probably was a disappointing format. |
| 80%               | very predictable. i like a refund, but a little heavy.                            |
| **W2V-STEM**      |                                                                                 |
| 20%               | was way too predictable. i had ordered a xlt, it probably was a plot.             |
| 40%               | was plot plot predictable. i had ordered a xlt, it probably was a plot.           |
| 60%               | was plot plot predictable. i had trilogy action xlt, plot author was action plot. |
| 80%               | ending plot plot predictable, plot had trilogy action xlt, plot author ending action plot. |

### Table 13: Example outputs for an Amazon evaluation line with phrase \textit{RE}

| MRT/RRT          | Generated Output                                                                 |
|------------------|----------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                 |
| 20%               | bought for my friendz's baby i hope she likes it. it looks cute. baby will like it. small cute |
| 40%               | bought for my daughter i hope she likes it. it looks cute. she will like it. very nice |
| 60%               | bought this for my daughter i and she loved it. it fits great. i will like it. thanks |
| 80%               | bought this for my daughter for christmas and she loves it. she like it. the color is great |
| **SMERTI-RNN**    |                                                                                 |
| 20%               | bought for my daughter i hope she likes it. it looks cute. i will like it. thank you |
| 40%               | bought for my daughter i and she likes it. it worked well. i will like it. thank you |
| 60%               | bought this item for my daughter and she loves it. she loves it. she will like it again. i love it |
| 80%               | love my daughter loves it. very good. he                                         |
| **W2V-STEM**      |                                                                                 |
| 20%               | bought for my daughter friendz's baby i hope she likes it it looks cute baby will like it small small cute |
| 40%               | bought for my daughter friendz's thing i hope she likes great smell cute baby will like great quality quality if |
| 60%               | bought for my daughter friendz's thing i would not rock solid great smell cute baby will like great quality quality if |
| 80%               | balinese my daughter friendz's thing i would not rock solid great smell cute baby as easy great quality quality if |
| Yelp - Noun $RE$ |  |
|------------------|------------------|
| **Input:**       | i love this place! very nice people running the cafe and the food is always good . stars! |
| **$RE$:**        | service |
| **MRT/RRT**      | **Generated Output** |
| **SMERTI-Transformer** | |
| 20%               | i love this place! very nice people running the service and the food is always good. stars! |
| 40%               | i love this place! very nice people running the service and the food is always good. food is delicious! |
| 60%               | i love this place! very nice people running the service and the food is always good. food is delicious! |
| 80%               | i love this place! great service and food. best egg rolls! |
| **SMERTI-RNN**   | |
| 20%               | i love this place! very nice people running the service and the food is always good. stars! |
| 40%               | i love this place! very nice people running the service and food is always good. thanks! |
| 60%               | i love this place! very nice people running the service and food is always good. thanks! |
| 80%               | i love this place! good service and fast. exceptional customer service! |
| **W2V-STEM**     | |
| 20%               | i love this place ! very nice people running the cafe also the service is always great . stars ! |
| 40%               | i love staff gym ! extremely knowledgable people running unbelievable boutique also unbelievable service was always great . stars ! |
| 60%               | i love staff gym ! extremely knowledgable people running unbelievable boutique also unbelievable service was always great . stars ! |
| 80%               | i love staff gym ! extremely knowledgable people running unbelievable boutique also unbelievable service was always great . stars ! |
| **GWN-STEM / NWN-STEM** | |
| 20% (0.075 $MIN_{2n}$) | i quality this quantities ! very nice people running the cafe and the quantities is always good . stars ! |
| 40% (0 $MIN_{2n}$)      | i quality this quantities ! very nice people running the movie and the quantities is always good . movie ! |

Table 14: Example outputs for a Yelp evaluation line with noun $RE$
| Input: | my favorite spot for lunch, they are quick and the food is always consistently excellent. |
|-------|----------------------------------------------------------------------------------------|
| RE:   | ordered                                                                               |

| MRT/RRT | Generated Output |
|---------|------------------|
| SMERTI-Transformer | |
| 20%     | my favorite spot for lunch, i ordered quick and the food is always consistently excellent. |
| 40%     | my favorite spot for lunch, i ordered quick service and the food is always consistently excellent. |
| 60%     | our favorite spot for lunch, i ordered quick and always fresh and excellent. |
| 80%     | fun spot for lunch, i ordered sashimi it was excellent. |
| SMERTI-RNN | |
| 20%     | my favorite spot for lunch, i ordered quick and the food is always consistently excellent. |
| 40%     | my favorite spot for lunch, always ordered quick delivery food and always consistently excellent. |
| 60%     | my favorite spot for lunch, always ordered quick and always always excellent. |
| 80%     | great spot for lunch, everything we ordered was excellent. |
| W2V-STEM | |
| 20%     | my favorite spot for lunch, they are quick and the food ordered always teriyaki excellent. |
| 40%     | my favorite spot for lunch, they are quick raw the lamb ordered always teriyaki excellent. |
| 60%     | my favorite spot got lunch, they were takeout raw pulled lamb ordered picked teriyaki excellent. |
| 80%     | my favorite spot got lunch, they were takeout raw pulled lamb ordered picked teriyaki excellent. |
| GWN-STEM | |
| 20% (0.1 $MIN_{sim}$) | my favorite spot for lunch, they are quick and the food is always consistently excellent. |
| 40% (0 $MIN_{sim}$)  | my favorite spot for lunch, they dropped quick and the food dropped always consistently excellent. |

Table 15: Example outputs for a Yelp evaluation line with verb $RE$
### Table 16: Example outputs for a Yelp evaluation line with adjective RE

| MRT/RRT          | Generated Output                                                                 |
|------------------|-----------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                   |
| 20%              | bad ass mexican options ! everything we had was awesome ! keep it rad guys thanks ! |
| 40%              | bad service and mexican options ! everything we had was awesome ! keep up your rad guys thanks ! |
| 60%              | great mexican food ! we had was awesome ! keep it up thanks !                       |
| 80%              | best mexican food ! wings are awesome ! love the service thanks !                   |
| **SMERTI-RNN**   |                                                                                   |
| 20%              | bad ass mexican options ! everything we had was awesome ! keep it rad guys thanks ! |
| 40%              | bad little mexican options ! everything we had was awesome ! keep to crowded and funny guys thanks ! |
| 60%              | best mexican food ! we had was awesome ! keep coming back thanks !                 |
| 80%              | best mexican cuisine ! super awesome ! great service and wonderful thanks !         |
| **W2V-STEM**     |                                                                                   |
| 20%              | bad ass mexican court ! everything we had was awesome ! keep it rad guys thanks !  |
| 40%              | bad ass mexican court ! chinese we had was awesome ! keep it rad guys thanks !      |
| 60%              | bad ass mexican court ! chinese we experienced was outstanding ! keep it hall guys thanks ! |
| 80%              | shitty ass mexican court ! chinese visiting experienced is outstanding ! keep just hall folks thanks ! |

Table 16: Example outputs for a Yelp evaluation line with adjective RE

### Table 17: Example outputs for a Yelp evaluation line with phrase RE

| MRT/RRT          | Generated Output                                                                 |
|------------------|-----------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                   |
| 20%              | cheap ! employees are nice ! great place to go to have a tuna sandwich !           |
| 40%              | cheap ! employees were nice ! great place to come back and have a tuna sandwich ! |
| 60%              | cheap ! employees were nice ! great place to come back for a tuna shop !          |
| 80%              | cheap ! employees are rude ! ill come back for a tuna sandwich !                   |
| **SMERTI-RNN**   |                                                                                   |
| 20%              | cheap ! employees are nice ! great will come back and have a tuna sandwich !       |
| 40%              | cheap ! employees are nice ! great will come back for a tuna meal !                |
| 60%              | cheap ! employees are amazing ! will never come back with a tuna and cheese !     |
| 80%              | cheap ! employees are amazing ! will definitely come back with a friend !          |
| **W2V-STEM**     |                                                                                   |
| 20%              | come back employees are nice great place to go to have a tuna sandwich              |
| 40%              | come back employees are nice great place to go to have a tuna sandwich              |
| 60%              | come back employees are nice great place to go to have a tuna sandwich              |
| 80%              | come back didn't even patio our server patio would be our server tuna sandwich      |

Table 17: Example outputs for a Yelp evaluation line with phrase RE
### News Headlines - Noun RE

| Input:         | trump's tactic twitter fights distract from absence of policy positions |
|---------------|------------------------------------------------------------------------|
| **RE:**       | video                                                                  |

| MRT/RRT       | Generated Output                                                                 |
|---------------|----------------------------------------------------------------------------------|
| SMERTI-Transformer |
| 20%           | trump's election campaign video exposes hypocrisy from absence of policy positions |
| 40%           | trump's campaign says twitter video is from a surge of policy positions          |
| 60%           | trump's latest video captures news from the power of controversial positions     |
| 80%           | trump's new video of                                                              |

| SMERTI-RNN    | Generated Output                                                                 |
|---------------|----------------------------------------------------------------------------------|
| 20%           | trump's america's twitter video goes from absence of policy positions             |
| 40%           | trump's instagram twitter video goes from a state of policy positions             |
| 60%           | trump's instagram video looks from women of russia positions                       |
| 80%           | trump's pick video about thursday                                                |

| W2V-STEM      | Generated Output                                                                 |
|---------------|----------------------------------------------------------------------------------|
| 20%           | trump's tactic twitter fights distract from absence of policy video               |
| 40%           | trump's photo twitter photo photo from photo of policy video                       |
| 60%           | trump's photo instagram photo photo from photo of election video                  |
| 80%           | trump's photo Instagram photo photo during photo captures election video          |

| GWN-STEM / NWN-STEM |
|---------------------|
| 20% (0.075 $MIN_{llm}$) | evidence plan twitter battle distract from lack of plan location |
| 40% (0 $MIN_{llm}$)     | evidence plan twitter battle distract from lack of plan location |

Table 18: Example outputs for a news headlines evaluation line with noun RE

### News Headlines - Verb RE

| Input:            | identity theft it's not just for grownups ! |
|-------------------|---------------------------------------------|
| **RE:**           | save                                        |

| MRT/RRT          | Generated Output                                                                 |
|-------------------|----------------------------------------------------------------------------------|
| SMERTI-Transformer |
| 20%               | identity theft it's not to save grownups !                                      |
| 40%               | identity theft it's to save grownups !                                          |
| 60%               | identity theft it's to save grownups !                                          |
| 80%               | let's save money !                                                              |

| SMERTI-RNN       | Generated Output                                                                 |
|-------------------|----------------------------------------------------------------------------------|
| 20%               | identity theft it's not to save grownups !                                      |
| 40%               | identity theft it's not to save grownups !                                      |
| 60%               | identity theft doesn't save grownups !                                          |
| 80%               | ways to save yourself !                                                         |

| W2V-STEM         | Generated Output                                                                 |
|-------------------|----------------------------------------------------------------------------------|
| 20%               | identity theft itsave not just for improve !                                    |
| 40%               | improve cell itsave not just for improve !                                      |
| 60%               | improve cell itsave not just for improve !                                      |
| 80%               | improve cell itsave not soon improve improve !                                  |

| GWN-STEM          | Generated Output                                                                 |
|-------------------|----------------------------------------------------------------------------------|
| 20% (0.1 $MIN_{llm}$) | space theft it's not just for grownups !                                      |
| 40% (0 $MIN_{llm}$)     | space space space's not just for jeopardy !                                    |

Table 19: Example outputs for a news headlines evaluation line with verb RE
| Input: | hasan minhaj comparing donald trump to stereotypical indian uncles is gold |
|--------|--------------------------------------------------------------------------------------------------|
| RE:    | american                                                                                         |

**Table 20: Example outputs for a news headlines evaluation line with adjective RE**

| Method          | Generated Output                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                                                                             |
| 20%              | the american comparing donald trump to create indian uncles is gold                                                                      |
| 40%              | the american comparing donald trump to artists who made me uncles                                                                          |
| 60%              | the american evolution of trump to win is gold                                                                                           |
| 80%              | defining american anti trump struggle                                                                                                    |
| **SMERTI-RNN**   |                                                                                                                                             |
| 20%              | this american comparing donald trump to the indian uncles is gold                                                                        |
| 40%              | the american comparing donald trump to tell uncles is gold                                                                               |
| 60%              | african american university president trump to congress is gold                                                                          |
| 80%              | african american leader on trump rally                                                                                                   |
| **W2V-STEM**     |                                                                                                                                             |
| 20%              | hasan minhaj comparing donald trump to stereotypical american uncles is gold                                                           |
| 40%              | hypochondriacs hypochondriacs comparing donald trump to hypochondriacs american uncles is gold                                        |
| 60%              | hypochondriacs hypochondriacs comparing donald trump to hypochondriacs american uncles is gold                                        |
| 80%              | hypochondriacs hypochondriacs discuss donald trump but hypochondriacs american hypochondriacs is soccer                                 |

**News Headlines - Phrase RE**

| Input: | kindergarten teacher allegedly drinks beer at school |
|--------|------------------------------------------------------|
| RE:    | donald trump                                        |

| Method          | Generated Output                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| **SMERTI-Transformer** |                                                                                                                                             |
| 20%              | donald trump allegedly drinks beer at mexico                                                          |
| 40%              | donald trump allegedly drinks beer president                                                          |
| 60%              | donald trump makes beer cans                                                                           |
| 80%              | donald trump will not apologize                                                                        |
| **SMERTI-RNN**   |                                                                                                                                             |
| 20%              | donald trump allegedly drinks beer at now                                                             |
| 40%              | donald trump allegedly drinks beer in big race                                                        |
| 60%              | donald trump takes a beer debate                                                                        |
| 80%              | donald trump cabinet challenge                                                                         |
| **W2V-STEM**     |                                                                                                                                             |
| 20%              | kindergarten teacher donald trump drinks beer at school                                               |
| 40%              | kindergarten teacher donald trump drinks beer at school                                               |
| 60%              | kindergarten teacher donald trump drinks beer bigger                                                  |
| 80%              | kindergarten teacher donald trump drinks beer bigger                                                  |

**Table 21: Example outputs for a news headlines evaluation line with phrase RE**