Mind Your Inflections!
Improving NLP for Non-Standard English with Base-Inflection Encoding

Samson Tan[§♮], **Shafiq Joty**[‡§], **Lav R. Varshney**[§], **Min-Yen Kan**[♮]
[§]Salesforce Research
[♮]National University of Singapore
[‡]Nanyang Technological University
[§]{samson.tan,sjoty,lvarshney}@salesforce.com
[♮]kanmy@comp.nus.edu.sg

Abstract

Morphological inflection is a process of word formation where base words are modified to express different grammatical categories such as tense, case, voice, person, or number. World Englishes, such as Colloquial Singapore English (CSE) and African American Vernacular English (AAVE), differ from Standard English dialects in inflection use. Although comprehension by human readers is usually unimpaired by non-standard inflection use, NLP systems are not so robust. We introduce a new Base-Inflection Encoding of English text that is achieved by combining linguistic and statistical techniques. Fine-tuning pre-trained NLP models for downstream tasks under this novel encoding achieves robustness to non-standard inflection use while maintaining performance on Standard English examples. Models using this encoding also generalize better to non-standard dialects without explicit training. We suggest metrics to evaluate tokenizers and extensive model-independent analyses demonstrate the efficacy of the encoding when used together with data-driven subword tokenizers.

1 Introduction

Large-scale neural models have proven successful at a wide range of natural language processing (NLP) tasks but are susceptible to amplifying discrimination against minority linguistic communities (Hovy and Spruit, 2016; Tan et al., 2020) due to selection bias in the training data and model overamplification (Shah et al., 2019).

Most datasets implicitly assume a distribution of perfect Standard English speakers, but this does not accurately reflect the majority of the global English speaking population that are either second language or non-standard dialect speakers (Crystal, 2003; Eberhard et al., 2019). This is particularly concerning because these World Englishes differ at the lexical, morphological, and syntactic levels (Kachru et al., 2009); over-sensitivity to these local variations predisposes English NLP systems to discriminate against speakers of World Englishes by either misunderstanding or misinterpreting them (Hern, 2017; Tatman, 2017).

In particular, Tan et al. (2020) recently show that current question-answering and machine translation systems are overly sensitive to non-standard inflection use, which is a common feature of dialects like Colloquial Singapore English (CSE) and African American Vernacular English (AAVE).[1] Since humans are able to internally correct for or ignore non-standard inflection use (Foster and Wigglesworth, 2016), we should expect NLP systems to be equally robust.

Existing work in adversarial robustness for NLP primarily focuses on adversarial training methods (Belinkov and Bisk, 2018; Ribeiro et al., 2018; Tan et al., 2020) or classifying and correcting adversarial examples (Zhou et al., 2019a). However, this effectively increases the size of the training dataset by including adversarial examples or training a new model to identify and correct perturbations, thereby significantly increasing the overall computational cost for creating hardened models.

These approaches also only operate on either raw text or the model and ignore tokenization, which transforms raw text into a form that the neural network can learn from. Here we introduce a new representation for word tokens that separates base and inflection to improve NLP robustness, in a sense making grammatical parsing explicit (Pāṇini, c. 500 BCE) and potentially changing the statistical properties of tokens (Zipf, 1965, p. 255).

Many NLP systems presently use a combination of a whitespace and punctuation tokenizer followed by a data-driven subword tokenizer like byte pair

[1]Examples in Appendix A of Tan et al. (2020).

encoding (BPE) (Sennrich et al., 2016).[2] However, a purely data-driven approach may fail to find the optimal encoding, both in terms of vocabulary efficiency and cross-dialectal generalization, thereby making the neural model more vulnerable to inflectional perturbations. As such, we:

- Propose Base-Inflection Encoding (BITE), which uses morphological information to help the data-driven tokenizer use its vocabulary efficiently and generate robust token sequences. In contrast to morphological and subword segmentation techniques like BPE and Morfessor (Creutz and Lagus, 2002), we reduce inflected forms to their base forms before re-injecting the inflection information into the encoded sequence via special inflection symbols. This approach gracefully handles the canonicalization of words with ablaut while allowing the original sentence to be easily reconstructed.

- Demonstrate BITE's effectiveness at hardening the NLP system to non-standard inflection use while preserving performance on Standard English examples. Crucially, simply fine-tuning the pre-trained model for the downstream task after adding BITE is sufficient. Unlike adversarial fine-tuning, our method does not increase the dataset size and is more environment-friendly.

- Show that BERT (Devlin et al., 2019) is less perplexed by non-standard dialectal data when equipped with BITE, which implies BITE helps the model generalize better across dialects.

- Conduct extensive model-independent analyses to demonstrate BITE's efficacy when used in tandem with data-driven subword tokenizers like BPE, WordPiece (Schuster and Nakajima, 2012), and unigram language model (LM) (Kudo, 2018). Since there is little prior work on quantitatively evaluating the efficacy of subword encoding schemes, we propose a number of metrics to operationalize and evaluate the vocabulary efficiency and semantic capacity of an encoding scheme. Our evaluation measures are generic and can be used to evaluate any tokenizer.

2 Related Work

Subword tokenization. Before neural models can learn, raw text must first be encoded into symbols with the help of a fixed-size vocabulary. Early models represented each word as a single symbol in the vocabulary (Bengio et al., 2001; Collobert et al., 2011) and uncommon words were represented by an `unknown` symbol. However, such a representation is unable to adequately deal with words absent in the training vocabulary. Therefore, subword representations like WordPiece (Schuster and Nakajima, 2012) and byte pair encoding (Sennrich et al., 2016) were proposed to encode out-of-vocabulary (OOV) words by segmenting them into subwords and encoding each subword as a separate symbol. This way, less information is lost in the encoding process since OOV words are approximated as a combination of subwords in the vocabulary. Wang et al. (2019) propose to reduce vocabulary sizes by operating on bytes instead of characters.

To make subword regularization more tractable, Kudo (2018) proposed an alternative method of building a subword vocabulary by reducing an initially oversized vocabulary down to the required size with the aid of a unigram language model, as opposed to incrementally building a vocabulary like in WordPiece and BPE variants.

Adversarial robustness in NLP. To harden NLP systems against adversarial examples, existing work largely uses adversarial training (Goodfellow et al., 2015; Jia and Liang, 2017; Ebrahimi et al., 2018; Belinkov and Bisk, 2018; Ribeiro et al., 2018; Iyyer et al., 2018; Cheng et al., 2019). However, this generally involves *retraining* the model with the adversarial data, which is computationally expensive and time-consuming. Tan et al. (2020) showed that simply fine-tuning the trained model for a single epoch on appropriately generated adversarial training data is sufficient to harden it against inflectional adversaries. Instead of adversarial training, Zhou et al. (2019b) propose using a BERT-based model to detect adversarial examples and recover the original examples. Jia et al. (2019) and Huang et al. (2019) use Interval Bound Propagation to train provably robust models.

3 Linguistically-Grounded Tokenization

In data-driven subword encoding schemes like BPE, the goal is to improve the model's ability to approximate the semantics of an unknown word by encoding words as subwords.

Although the fully data-driven nature of such methods make them language-independent, this forces them to rely only on the statistics of the surface forms when transforming words into subwords

[2] SentencePiece treats whitespace as just another character.

Dataset	Condition	WordPiece (WP)	BITE + WP		WP + Adv. FT.
SQuAD 2.0 Ans. Qns (F_1)	Clean	**74.58**	74.50	75.46	**79.07**
	Adv.	61.37	**71.33**	**72.56**	72.21
SQuAD 2.0 All Qns (F_1)	Clean	**72.75**	72.71	73.69	**74.45**
	Adv.	59.32	**69.23**	**70.66**	68.23
MultiNLI (Acc)	Clean	**83.44**	83.01	82.21	**83.86**
	Adv.	58.70	**76.11**	81.05	**83.87**
MultiNLI-MM (Acc)	Clean	**83.59**	83.50	83.36	**83.86**
	Adv.	59.75	**76.64**	**81.04**	75.77

Table 1: $\text{BERT}_{\text{base}}$ results on the clean and adversarial MultiNLI and SQuAD 2.0 datasets. We compare WordPiece+BITE to both WordPiece alone and with one epoch of adversarial fine-tuning. To ensure a fair comparison with adversarial fine-tuning, we trained the BITE-equipped model for an extra epoch (right column) on clean data.

Algorithm 1 Base-Inflection Encoding (BITE)

Require: Input sentence $S = [w_1, \ldots, w_N]$
Ensure: Tokenized sequence T
  $T \leftarrow [\emptyset]$
  for all $i = 1, \ldots, |N|$ **do**
    if POS$(w_i) \in$ {NOUN, VERB, ADJ} **then**
      $base \leftarrow$ GETLEMMA(w_i)
      $inflection \leftarrow$ GETINFLECTION(w_i)
      $T \leftarrow T + [base, inflection]$
    else
      $T \leftarrow T + [w_i]$
    end if
  end for
  return T

since they do not make any language-specific morphological assumptions. To illustrate, the past tense of `go`, `take`, and `keep` have the inflected forms `went`, `took`, and `kept`, respectively, which have little to no overlap with their base forms and each other even though they share the same tense. These six surface forms would likely have no subwords in common in the vocabulary, thereby putting the burden of learning both the relation between base forms and inflected forms and the relation between inflections for the same tense on the model. Additionally, since vocabularies are fixed before model training, such an encoding does not optimally use a limited vocabulary.

Even when inflections do not exhibit ablaut and there is a significant overlap between the base and inflected forms, e.g., the `-ed` and `-d` suffixes, there is no guarantee that the suffix will be encoded as a separate subword and that the base forms and suffixes will be consistently represented. To illustrate, encoding `danced` as {`dance`, `d`} and `dancing` as {`danc`, `ing`} results in two different "base forms" for the same word, `dance`. This again places the burden of learning that the two "base forms" mean the same thing on the model and makes inefficient use of a limited vocabulary.

When encoded in conjunction with another inflected form like `entered`, which should be encoded as {`enter`, `ed`}, this encoding scheme also produces two different subwords for the same type of inflection `-ed` vs `-d`. Like the first example, the burden of learning that the two suffixes correspond to the same tense is transferred to the learning model.

A possible solution is to instead encode `danced` as {`danc`, `ed`} and `dancing` as {`danc`, `ing`}, but there is no guarantee that a data-driven encoding scheme will learn this pattern without some language-specific linguistic supervision. In addition, this unnecessarily splits up the base form into two subwords `danc` and `e`; the latter contains no extra semantic or grammatical information yet increases the tokenized sequence length. Although individually minor, encoding many base words in this manner increases the computational cost for any encoder or decoder network.

Finally, although it is theoretically possible to force a data-driven tokenizer to segment inflected forms into morphologically logical subwords by limiting the vocabulary size, many inflected forms are represented as individual symbols at common vocabulary sizes (30–40k). We found that the BERT_{base} WordPiece tokenizer and BPE[3] encoded each of the above examples as single symbols.

To address these issues, we propose the Base-Inflection Encoding framework (or BITE), which encodes the base form and inflection of content words separately. Similar to how existing subword encoding schemes improve the model's ability to approximate the semantics of out-of-vocabulary words with in-vocabulary subwords, BITE helps

[3]Trained on Wikipedia+BookCorpus (1M) with a vocabulary size of 30k symbols.

the model handle out-of-distribution inflection usage better by keeping a content word's base form consistent even when its inflected form drastically changes. This distributional deviation could manifest as *adversarial* examples, such as those generated by MORPHEUS (Tan et al., 2020), or sentences produced by non-standard English (L2 or dialect) speakers. As a result, BITE provides adversarial robustness to the model.

3.1 Base-Inflection Encoding

Given an input sentence $S = [w_1, \ldots, w_N]$ where w_i is the i^{th} word, BITE generates a sequence of tokens $S' = [w'_1, \ldots, w'_N]$ such that $w'_i = [\text{BASE}(w_i), \text{INFLECT}(w_i)]$ where $\text{BASE}(w_i)$ is the base form of the word and $\text{INFLECT}(w_i)$ is the inflection (grammatical category) of the word (Algorithm 1). If w_i is not inflected, $\text{INFLECT}(w_i)$ is NULL and excluded from the final sequence of tokens to reduce the neural network's computational cost. In our implementation, we use Penn Treebank tags to represent inflections. For example, "Jack jumped the hurdles" would be encoded as

```
[Jack, jump, VBD, the, hurdle, NNS].
```

By lemmatizing each word to obtain the base form instead of segmenting it like in most data-driven encoding schemes, BITE ensures this base form is consistent for all inflected forms of a word, unlike a subword produced by segmentation, which can only contain characters available in the original word. For example, $\text{BASE}(took)$, $\text{BASE}(taking)$, and $\text{BASE}(taken)$ all correspond to the same base form, `take`, even though it is significantly orthographically different from `took`.

Similarly, encoding all inflections of the same grammatical category (e.g., *verb-past-tense*) in a canonical form should help the model to learn each inflection's grammatical role more quickly. This is because the model does not need to first learn that the same grammatical category can manifest in orthographically different forms.

Finally and most importantly, this encoding process is informationally lossless and we can easily reconstruct the original sentence using the base forms and grammatical information preserved by the inflection tokens.

3.2 Compatibility with Data-Driven Methods

Although BITE has the numerous advantages outlined above, it suffers from the same weakness as regular word-level tokenization schemes when used alone: a limited ability to handle out-of-vocabulary words. Hence, we designed BITE to be a general framework that seamlessly incorporates existing data-driven schemes to take advantage of their proven ability to handle OOV words.

To achieve this, a whitespace/punctuation-based pre-tokenizer is first used to transform the input into a sequence of words and punctuation characters, as is common in neural machine translation. Next, BITE is applied and the resulting sequence is converted into a sequence of integers by a data-driven encoding scheme (e.g., BPE). In our experiments, we use BITE in this manner and refer to the combined tokenizer as "BITE+D", where D refers to the data-driven encoding scheme.

4 Model-based Experiments

In this section, we demonstrate the effectiveness of BITE using the pre-trained cased $\text{BERT}_{\text{base}}$ model (Devlin et al., 2019) implemented by Wolf et al. (2019). We do not replace WordPiece but instead extend incorporate it into the BITE framework as described in §3.2. There are both advantages and disadvantages to this approach, which we will discuss in the next section.

4.1 Adversarial Robustness

We evaluate BITE on question answering and natural language understanding datasets, SQuAD 2.0 (Rajpurkar et al., 2018) and MultiNLI (Williams et al., 2018), respectively. Following Tan et al. (2020), we report F_1 scores on both the full SQuAD 2.0 dataset and only the answerable questions. In addition, we also report scores for the out-of-domain development set (MultiNLI-MM).

For the experiments in this subsection, we use MORPHEUS (Tan et al., 2020) to generate adversarial examples that resemble second language English speaker (L2) sentences. These adversarial examples, while synthetic, can be thought of as "worst-case" examples of L2 sentences for the model under test. This is done separately for each $\text{BERT}_{\text{base}}$ model.

BITE vs. BITE-less. First, we demonstrate the effectiveness of BITE at making the model robust to inflectional adversaries. After fine-tuning two separate $\text{BERT}_{\text{base}}$ models (one with BITE, the other with standard WordPiece) on SQuAD 2.0 and MultiNLI with Wolf et al. (2019)'s default hyperparameters, we generate adversarial examples for them using MORPHEUS. From Table 1, we observe

that the BITE-equipped model not only achieves similar performance ($\pm$0.5) on clean data, but is significantly more robust to inflectional adversaries (10 points for SQuAD, 17 points for MultiNLI).

BITE vs. Adversarial Fine-tuning. Next, we compare the BITE to adversarial fine-tuning (Tan et al., 2020), an economical variation of adversarial training (Goodfellow et al., 2015) for making models robust to inflectional perturbations. In adversarial fine-tuning, an adversarial training set is generated by randomly sampling inflectional adversaries k times from the adversarial distribution found by MORPHEUS and adding them to the original training set. Rather than retraining the model on this adversarial training set, the previously trained model is simply trained for one extra epoch. In this experiment, we follow the above methodology and adversarially fine-tune the WordPiece-only $\text{BERT}_{\text{base}}$ for one epoch with k set to 4. To ensure a fair comparison, we also train the BITE-equipped $\text{BERT}_{\text{base}}$ on the same training set for an extra epoch.

From Table 1, we observe that BITE is often more effective than adversarial fine-tuning at making the model more robust against inflectional adversaries and in some cases (SQuAD 2.0 All and MNLI-MM) even without needing the additional epoch of training.

However, the adversarially fine-tuned model consistently achieves better performance on clean data. This is likely due to the fact that even though adversarial fine-tuning requires only a single epoch of extra training, the process of generating the training set increases its size by a factor of k and therefore the computational cost. In contrast, BITE requires no extra training and is much more economical.

Adversarial fine-tuning's poorer performance on the out-of-domain adversarial data (MultiNLI-MM) compared to the in-domain data hints at a possible weakness: it is less effective at inducing model robustness when the adversarial example is from an out-of-domain distribution.

BITE, on the other hand, performs equally well on both in- and out-of-domain data, demonstrating its applicability to practical scenarios where the training and test domain may not match.

4.2 Dialectal Variation

Apart from second languages, dialects are another common source of non-standard inflection use. However, there is a dearth of task-specific datasets in English dialects like AAVE and CSE. Therefore, in this section's experiments, we use the model's perplexity on monodialectal corpora as a proxy for its performance on downstream tasks in the corresponding dialect. The perplexity reflects the pre-trained model's generalization ability on the dialectal datasets.

4.2.1 Corpora

For AAVE, we use the Corpus of Regional African American Language (CORAAL) (Kendall and Farrington, 2018), which comprises transcriptions of interviews with African Americans born between 1891 and 2005. For our evaluation, only the interviewee's speech was used. In addition, we strip all in-line glosses and annotations from the transcriptions before dropping all lines with less than three words. After pre-processing, this corpus consists of slightly under 50k lines of text (1,144,803 tokens, 17,324 word types).

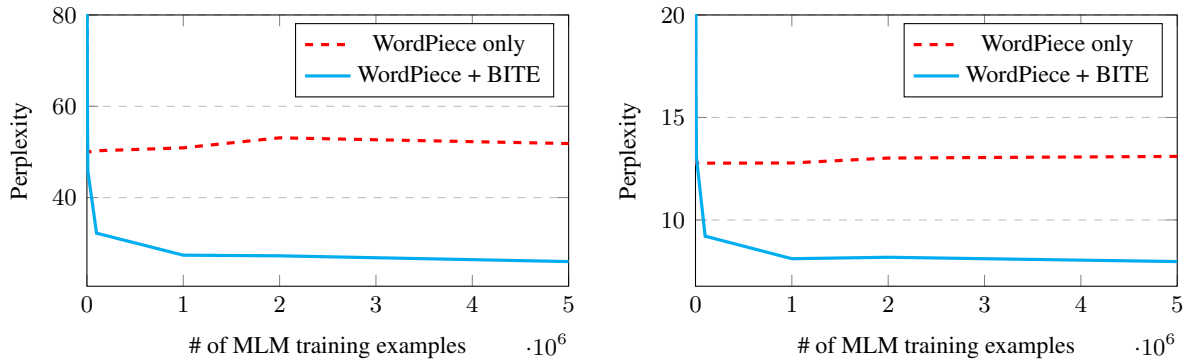
To obtain a CSE corpus, we scrape the Infotech Clinics section of the Hardware Zone Forums[4], a forum frequented mainly by Singaporeans and where CSE is commonly used. Similar pre-processing to the AAVE data yields a 2.2 million line corpus (45,803,898 tokens, 253,326 word types).

4.2.2 Experimental Setup

In this experiment, we take the same pre-trained $\text{BERT}_{\text{base}}$ model and fine-tune two separate variants (with and without BITE) on Wikipedia and BookCorpus (Zhu et al., 2015) using the masked language modeling (MLM) loss without the next sentence prediction (NSP) loss. We fine-tune for one epoch on increasingly large subsets of the dataset, since this has been shown to be more effective than doing the same number of gradient updates on a fixed subset (Raffel et al., 2019).

Next, we evaluate model perplexities on the AAVE and CSE corpora, which we consider to be from dialectal distributions that differ from the training data which is considered to be Standard English. Since calculating the "masked perplexity" requires randomly masking a certain percentage of tokens for prediction, we also experiment with doing this for each sentence multiple times before averaging the perplexity. However, we find no significant difference between doing the calculation once or five times; the random effects likely cancel out due to the large size of our corpora.

[4]https://forums.hardwarezone.com.sg/



(a) Colloquial Singapore English (forum threads)

(b) African American Vernacular English (CORAAL)

Figure 1: Perplexity of $\text{BERT}_{\text{base}}$ with and without BITE on CSE and AAVE corpora. To allow the model to adapt to the new inflection tokens added to the vocabulary, we perform masked language model (MLM) training for one epoch on increasingly large subsets of Wikipedia+BookCorpus. We observe a large initial perplexity for the WordPiece+BITE model as it has not fully adapted to the new inflection tokens; this stabilizes to around half of the WordPiece-only model's perplexity as we increase the training dataset size.

4.2.3 Results

From Figure 1, we observe that the BITE-equipped model initially has a much higher perplexity, before converging to around 50% of the standard model's perplexity as the model adapts to the presence of the new inflection tokens (e.g., `VBD`, `NNS`, etc.). Crucially, the models are not trained on dialectal corpora, which demonstrates the effectiveness of BITE at helping models better generalize to dialectal distributions after a short adaptation phase.

CSE vs. AAVE. Astute readers might notice that there is a large difference in perplexity between the two dialectal corpora, even for the same tokenizer combination. There are two possible explanations. The first is that CSE, being an amalgam of English, Chinese dialects, Malay, etc., differs significantly from Standard English not only morphologically, but also in word ordering (e.g., topic prominence) (Tongue, 1974). In addition, numerous loan words and discourse particles not found in Standard English like *lah*, *lor* and *hor* are commonplace in CSE (Leimgruber, 2009). AAVE, however, generally shares the same word ordering as Standard English due to its largely English origins (Poplack, 2000) and is less different linguistically (compared to CSE vs. Standard English). These differences between AAVE and CSE are likely explanations for the significant differences in perplexity.

Another possible explanation is that the BookCorpus may contain examples of AAVE since the BookCorpus' source, Smashwords, also publishes African American fiction. We believe the reason for the difference is a mixture of these two factors.

5 Model-Independent Analyses

Although wrapping an existing pretrained subword tokenizer and model like WordPiece and $\text{BERT}_{\text{base}}$ allows us to quickly reap the benefits of BITE without the computational overhead of pretraining them from scratch, such an approach likely does not make use of BITE's full potential. This is due to the compressing effect that BITE offers on the lexical space. Therefore, in this section, we analyze WordPiece, BPE, and unigram language model (Kudo, 2018) subword tokenizers that are trained from scratch with and without BITE using the `tokenizers`[5] and `sentencepiece`[6] libraries. Through our experiments we aim to answer the following two questions:

- Does BITE help the data-driven tokenizer use its vocabulary more efficiently (§5.1)?
- How does BITE improve adversarial robustness (§5.2)?

We draw one million examples from English Wikipedia and BookCorpus for use as our training set. Unless otherwise mentioned, we use another 5k examples as our test set. Although SentencePiece can directly process raw text, we pretokenize the raw text with the BertPreTokenizer from the `tokenizers` library before encoding them for ease of comparison across the three encoding schemes. For practical applications, users should be able to apply SentencePiece's method

[5] `github.com/huggingface/tokenizers`
[6] `github.com/google/sentencepiece`

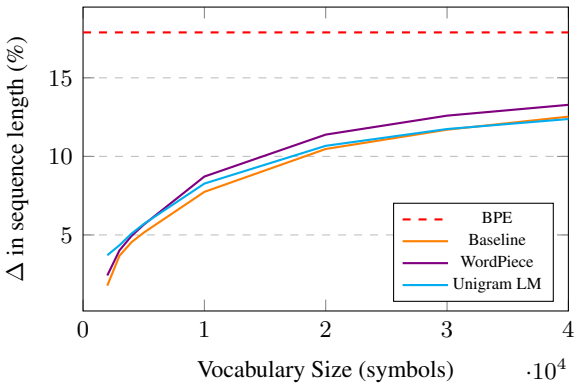


Figure 2: Relative increase in mean tokenized sequence lengths (%) between BITE-less and BITE-equipped tokenizers after training the data-driven subword tokenizers with varying vocabulary sizes; lower is better. Baseline (dotted red line) denotes the percentage of inflected forms in an average sequence; this is equivalent to the increase in sequence length if BITE had no effect on the data-driven tokenizers' encoding efficiency.

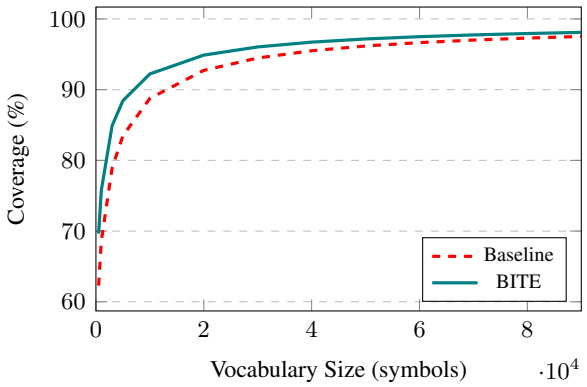


Figure 3: Comparison of coverage of the Wikipedia+BookCorpus (1M examples) dataset between BITE and a trivial baseline (word counts). Since BITE has no fixed vocabulary, we determine the "vocabulary" by taking the N most common tokens.

of handling whitespace characters instead of the BertPreTokenizer with no issues.

5.1 Vocabulary Efficiency

We may operationalize the question of whether BITE improves vocabulary efficiency in at least three ways. The most straightforward is to measure the sequence-level change: the difference in the tokenized sequence length with and without BITE. Efficient use of a fixed-size vocabulary should result in it comprising symbols that minimize the average tokenized sequence length. An alternative vocabulary-level perspective is to observe how well a limited vocabulary represents a corpus. Finally, since the base form of a word is sufficient to semantically represent all its inflected forms (Jackson, 2014), the number of unique linguistic lexemes[7] that can be represented by a vocabulary should also be a good measure how much "semantic knowledge" it contains.

Sequence lengths. A possible concern with BITE is that it may significantly increase the length of the tokenized sequence, and hence the computational cost for sequence modeling, since it splits all inflected content words (nouns, verbs, and adjectives) into two tokens. We calculate the percentage of inflected tokens to be 17.89%.[8] Therefore, if BITE did not enhance WordPiece's and BPE's encoding efficiency, we should expect a 17.89% increase (i.e., upper bound) in their mean tokenized sequence length. However, from Figure 2, we see this is not the case as the relative increase (with and without BITE) in mean sequence length generally stays below 13%, 5% less than the baseline. This demonstrates that BITE helps the data-driven tokenizer make better use of its limited vocabulary.

In addition, we see that the gains are inversely proportional to the vocabulary size. This is likely due to the following reasons. For a given sentence, the corresponding tokenized sequence's length usually decreases as the data-driven tokenizer's vocabulary size increases as it allows merging of more smaller subwords into longer subwords. On the other hand, BITE is vocabulary-independent, which means that the tokenized sequence length is always the same for a given sentence. Therefore, we can expect the same absolute difference to contribute to a larger relative increase as the vocabulary size increases. Additionally, more inflected forms are memorized as the vocabulary size increases, resulting in an average absolute increase of 0.4 tokens per sequence for every additional 10k vocabulary tokens. These two factors together should explain the above phenomenon.

Vocabulary coverage. Next, we look at this question from a vocabulary-level perspective and operationalize vocabulary efficiency as the coverage of a representative corpus by a vocabulary's tokens. We measure coverage by computing the total number of tokens (words and punctuation) in the corpus that are represented in the vocabulary divided by the total number of tokens in the corpus.

[7] A lexeme is the set of a word's base and inflected forms.
[8] Note that only content words are subject to inflection.

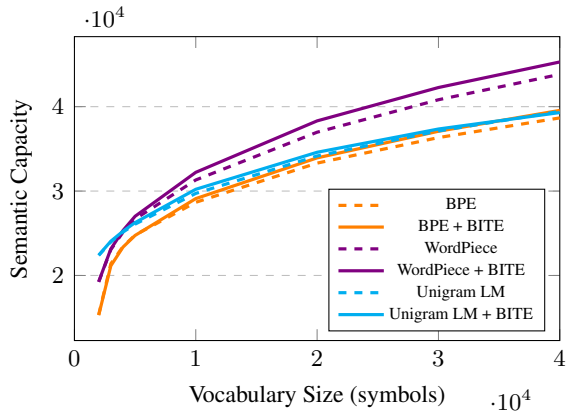


Figure 4: Semantic capacity of each tokenizer combination's vocabulary, as computed in Equations (1) and (2). Higher is better.

Since BITE does not require a vocabulary to be fixed at training time, we simply set the N most frequent tokens (base forms and inflections) to be our vocabulary. We use the N most frequent tokens in the unencoded text as our baseline vocabulary.

From Figure 3, we observe that the BITE vocabulary achieves a higher coverage of the corpus than the baseline, hence demonstrating the efficacy of BITE at improving vocabulary efficiency. Additionally, we note that this advantage is most significant when the vocabulary is between 5–10k tokens. This implies that inflected word forms comprise a large portion of frequently occurring words, which comports with intuition.

Semantic capacity. Since a word's base form is able to semantically represent all its inflected forms (Jackson, 2014), we posit that the number of base forms contained in a vocabulary can be a good proxy for the amount of semantic knowledge it contains. This may be termed the *semantic capacity* of a vocabulary. In whole-word (non-subword) vocabularies, semantic capacity may be measured by the number of unique lexemes in the vocabulary.

For subword vocabularies like BPE's, however, such a measure is slightly less useful since every English word can be trivially represented given the alphabet and a hyphen. Thus, we propose a generalization of the above definition which penalizes the use of multiple (and unknown) tokens to represent a single base form. Formally,

$$f(T_i) = \begin{cases} \frac{1}{|T_i|+u_i} & |T_i| - u_i > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

$$\text{SemCapacity}(T_1, \ldots, T_N) = \sum_{i=1}^{N} f(T_i), \quad (2)$$

where N is the total number of unique base forms in the evaluation corpus, T_i is the sequence of (subword) tokens obtained from encoding the ith base form, and u_i is the number of unknown tokens[9] in T_i. While not strictly necessary when comparing vocabularies on the same corpus, normalizing Equation (2) by the number of unique base forms in the corpus may be helpful for cross-corpus comparisons. Further normalizing the resulting quantity by the tokenizer's vocabulary size yields a measure of semantic *efficiency*.

Although it is possible to extend Equation (1) to cover cases where there are only multiple unknown tokens, this would unnecessarily complicate the equation. Hence, we define $f(T_i) = 0$ when there are *only* unknown tokens in the encoded sequence. We also implicitly define the penalty of each extra unknown token to be double[10] that of a token in the vocabulary. If necessary, it is trivial to alter the weight of this penalty by introducing a constant λ:

$$f(T_i, \lambda) = \begin{cases} \frac{1}{(|T_i|-u_i)+\lambda u_i} & |T_i| - u_i > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

To evaluate the semantic capacity of our vocabularies, we use WordNet (Miller, 1995) as our representative "corpus". Specifically, we only consider its single-word lemmas ($N = 83118$). From Figure 4, we see that data-driven tokenizers trained with BITE tend to produce vocabularies with higher semantic capacities.

Additionally, we observe that tokenizer combinations incorporating WordPiece or unigram LM generally outperform the BPE ones. We believe this to be the result of using a language model to inform vocabulary generation. It is logical that a symbol that maximizes a language model's likelihood on the training data is also semantically "denser", hence prioritizing such symbols produces semantically efficient vocabularies. We leave the in-depth investigation of this relationship to future work.

[9] Usually represented as `[UNK]` or `<unk>`.
[10] $|T|$ contributes the extra count.

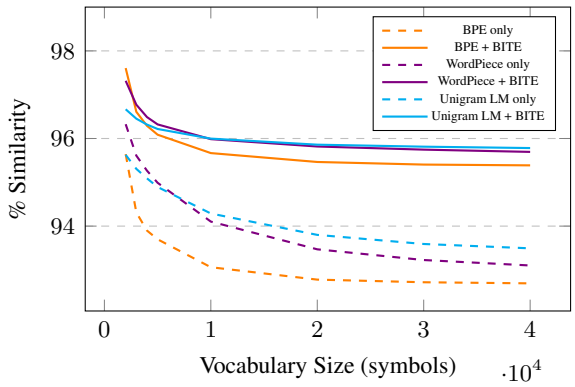


Figure 5: Mean percentage of tokens that are the same in the clean and adversarial tokenized sequences.

5.2 Adversarial Robustness.

BITE's ability to make models more robust to inflectional perturbations can be directly attributed to its preservation of a consistent, inflection-independent base form. We demonstrate this by measuring the similarity between the encoded clean and adversarial sentences, using the Ratcliff/Obershelp algorithm (Ratcliff and Metzener, 1988) as implemented by Python's `difflib`. We use the MultiNLI in-domain development dataset and the MORPHEUS adversaries generated in §4.1 for this experiment.

We find that clean and adversarial sequences tokenized by the BITE-equipped tokenizers were more similar (1–2% for WordPiece, 2–2.5% for BPE) than those tokenized by the ones without BITE (Figure 5). The decrease in similarity for all conditions as the vocabulary size increases is unsurprising; a larger vocabulary will generally result in shorter sequences and the same number of differing tokens will be a larger relative change.

The above results demonstrate that the improved robustness shown in §4.1 can be directly attributed to the separation of each content word's base forms from its inflection and keeping it consistent as the inflection varies, hence mitigating any significant token-level changes.

6 Conclusion

The tokenization stage of the modern deep learning NLP pipeline has not received as much attention as the modeling stage, with researchers often defaulting to a commonly used subword tokenizer like BPE. Adversarial robustness techniques in NLP also largely focus on augmenting the training data with adversarial examples. However, we posit that the process of encoding raw text into network-operable symbols may have more impact on a neural network's generalization and adversarial robustness than previously assumed.

Hence, we propose to improve the tokenization pipeline by incorporating linguistic information to guide the data-driven tokenizer in learning a more efficient vocabulary and generating token sequences that increase the neural network's robustness to non-standard inflection use. We show that this improves its generalization to second language English and World Englishes without requiring explicit training on such data. Since dialectal data is often scarce or even nonexistent (in the case of task-specific labeled datasets), an NLP system's ability to generalize across dialects in a zero-shot manner is crucial for it to work well for diverse linguistic communities.

Finally, given the effectiveness of the common task framework for spurring progress in NLP (Varshney et al., 2019), we hope to do the same for tokenization. As a first step, we propose to evaluate an encoding scheme's efficacy by measuring its vocabulary efficiency and semantic capacity (which may have interesting connections to information-theoretic limits (Ziv and Lempel, 1978)). We have already shown that Base-Inflection Encoding helps a data-driven tokenizer use its limited vocabulary more efficiently by increasing its semantic capacity when the combination is trained from scratch.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations*, Vancouver, BC, Canada.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2001. A neural probabilistic language model. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 932–938. MIT Press.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333, Florence, Italy. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(76):2493–2537.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

David Crystal. 2003. *English as a Global Language*. Cambridge University Press.

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.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2019. *Ethnologue: Languages of the World*, 22 edition. SIL International.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Pauline Foster and Gillian Wigglesworth. 2016. Capturing accuracy in second language performance: The case for a weighted clause ratio. *Annual Review of Applied Linguistics*, 36:98–116.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*, San Diego, California.

Alex Hern. 2017. Facebook translates 'good morning' into 'attack them', leading to arrest. *The Guardian*.

Dirk Hovy and Shannon L. Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, Berlin, Germany. Association for Computational Linguistics.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4083–4093, Hong Kong, China. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Howard Jackson. 2014. *Words and their Meaning*. Routledge.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4129–4142, Hong Kong, China. Association for Computational Linguistics.

Braj B. Kachru, Yamuna Kachru, and Cecil Nelson, editors. 2009. *The Handbook of World Englishes*. Wiley-Blackwell.

Tyler Kendall and Charlie Farrington. 2018. The corpus of regional african american language.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Jacob RE Leimgruber. 2009. *Modelling variation in Singapore English*. Ph.D. thesis, Oxford University.

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

Pāṇini. c. 500 BCE. *Aṣṭadhyāyī*.

Shana Poplack. 2000. *The English History of African American English*. Blackwell Malden, MA.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, arXiv:1910.10683.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

John W Ratcliff and David E Metzener. 1988. Pattern-matching: The gestalt approach. *Dr Dobbs Journal*, 13(7):46.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Deven Shah, H. Andrew Schwartz, and Dirk Hovy. 2019. Predictive biases in natural language processing models: A conceptual framework and overview. *arXiv e-prints*, arXiv:1912.11078.

Samson Tan, Shafiq Joty, Min-Yen Kan, and Richard Socher. 2020. Its Morphin Time! Combating linguistic discrimination with inflectional perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Seattle, Washington. Association for Computational Linguistics.

Rachael Tatman. 2017. Gender and dialect bias in YouTube's automatic captions. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 53–59, Valencia, Spain. Association for Computational Linguistics.

Ray K Tongue. 1974. *The English of Singapore and Malaysia*. Eastern Universities Press.

Lav R. Varshney, Nitish Shirish Keskar, and Richard Socher. 2019. Pretrained AI models: Performativity, mobility, and change. *arXiv e-prints*, arXiv:1909.03290.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2019. Neural machine translation with byte-level subwords. *arXiv e-prints*, arXiv:1909.03341.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv e-prints*, arXiv:1910.03771.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019a. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019b. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4906–4915, Hong Kong, China. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

George K. Zipf. 1965. *The Psycho-Biology of Language: An Introdution to Dynamic Philology*. MIT Press, Cambridge, MA, USA.

Jacob Ziv and Abraham Lempel. 1978. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536.