Semantic Diversity by Phonetics for Accurate and Robust Machine Translation

Abstract

Neural Machine Translation (NMT) learns from examples, and thus often lacks robustness against noise. Previous work has shown that integrating noise into the training process is effective at improving such robustness, but this solution can be inefficient due to the exponential number of string perturbations, i.e., exponential in the number of words or characters. To robustify the translation input, we treat human phonetic interaction throughout history as a pre-compiled computational device. This device implements a many-to-one function that converts text into phonetics. To the best of our knowledge, we are the first in Machine Translation, to apply the phonetic algorithms Soundex, NYSIIS, and MetaPhone to foreign word/character sequences. We also apply another linguistic representation, the logogram inference, Wubi, for Chinese. To explain why phonetic encodings improve NMT, we introduce, quantify, and empirically verify our hypothesis: "one phonetic representation usually corresponds to words that are semantically diverse." Driven by our hypothesis, we simulate this "natural" phonetic device and introduce an artificial method called random clustering. We achieved significant and consistent improvements overall language pairs and datasets we experimented with: French-English, German-English, and Chinese-English in IWSLT'17, with up to nearly 2 BLEU points over the state-of-the-art. Moreover, our approaches are more robust than baselines when evaluated on unknown noisy or out-of-domain test sets, with up to about 5 BLEU point increase.

1 Introduction

Machine translation (MT) has achieved remarkable success with milestone contributions including, but not limited to, (Koehn et al., 2007; Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). State-of-the-art MT systems are trained with massive human labeled samples and have achieved high accuracy in evaluations (Bojar et al., 2018). However, in real life, an out-of-domain test set can easily make MT fail. Moreover, noisy sets due to string distortions like typos, slang, dialect, idiolect and informal use of languages such as acronym, abbreviation, and emoji can have adverse effects on MT and lower translation quality (Belinkov and Bisk, 2018; Khayrallah and Koehn, 2018; Wang et al., 2018).

Important past works have branched in two main directions: domain adaptation (Jiang, 2008; Carpuat et al., 2013; Freitag and Al-Onaizan, 2016; Wang et al., 2017; van der Wees et al., 2017; Zhang et al., 2019) and noisy data augmentation (Liu et al., 2018; Karpukhin et al., 2019; Vaibhav et al., 2019). Both leverage the datasets between training and testing but one with aspects of the domain and the other with text expression, respectively. Though experiments have demonstrated that these methods are effective, they can be inefficient for NMT training due to a large number of string perturbations and possible domains. Importantly, they cannot be generalized to an arbitrary distorted test set if we know nothing about its distribution in advance.

We aim to improve MT robustness and accuracy in general. We introduce a novel framework for NMT using phonetic information "computed" by the human interaction throughout the evolution of spoken language. We view social interaction as a computational device that generates pre-computed knowledge. Phonology has been shown to preserve semantic meanings (Tyler et al., 1996; Beaver et al., 2007), which coincides with neurological discoveries about the correlation between phonology and semantics in the human brain (Wang et al., 2016; Amenta et al., 2017; Paz-Alonso et al., 2018). Table 1 shows examples of Pinyin and Soundex. We view these encodings as many-to-one functions, where multiple words are mapped to one.

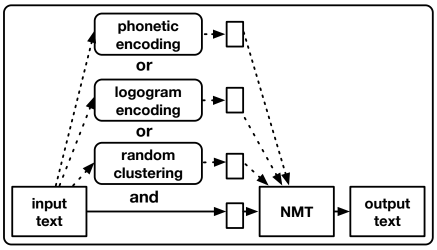


Figure 1: Workflow: all these bring NO new information (nothing new than the input we already have): BPE and embedding are in the empty box

function	input	output
Pinyin	xiao4	笑(smile), 校(school), 孝(filial), 效(imitate)
	shi4	氏(surname),事(matter),市(city),视(vision)
Soundex	B300	body,but,bad
	S120	speak,space,suppose,speech
	C600	car,care,chair,cherry,choir,cry,crow,core

Table 1: phonetics is a many-to-one function

Thus, we decided to give, by using phonetic algorithms, this new form of phonetic sentence representation together with its written form as an input when training and decoding the neural networks. It is a challenging task to purely rely on neural networks to extract all hidden features in NMT. Therefore, adding auxiliary information will potentially allow a simpler network structure. As shown in Figure 1, we first apply phonetic encoding, logogram, or random clustering to the foreign input-sentences, then use Byte-Pair-Encoding to learn a word embedding (marked as empty boxes) on each individual coding representation. Finally, we concatenate them with the embedded original text to feed into the NMT model.

We achieved significant and consistent improvements over the state-of-the-art on *all* language pairs that we experimented with. Our approaches robustify the baseline NMT. In particular, they lead to a higher accuracy even on an arbitrary test set whose distribution is oblivious at training time. This is a general approach that is applicable to any language that can be compiled with phonetics, and it generically benefits *any* NMT system, which it treats as a black-box.

Importantly, we performed a systematic empirical analysis to explain why phonetic encoding helps in NMT. Our hypothesis of semantic diversity by phonetics is stated as

"One phonetic representation usually corresponds to characters/words that are semantically diverse."

We performed three quantitative analyses to verify this hypothesis. Then, driven by our hypothesis, we introduced a new random clustering algorithm that casts words or characters into classes, which also improved translation accuracy and robustness.

This work contains two areas of study: phonetic encoding and random clustering, where the former inspires the latter in accordance with our stated hypothesis. Our contributions mainly include the hypothesis of semantic diversity by phonetics and below models:

1. *Phonetic and logogram encodings* We convert the source input text using various algorithms such as Soundex, NYSIIS, and MetaPhone for western languages: English, French, and German. For Chinese, we apply a Logogram encoding, Wubi. Both phonetic and logogram encodings as auxiliary inputs significantly improve translation results.

2. *Random clustering* Word/character clustering significantly improves NMT. This empirical finding aligns with the empirical justification of why phonetic encoding improves translation accuracy.

We conducted extensive experiments and achieved up to nearly 2 BLEU points on IWSLT'17 tasks over the state-of-the-art in translation directions of English-German, German-English, English-French, French-English, and Chinese-English. We verified that our approaches are more robust on French-English experiments with about 5 BLEU point improvement on a foreign test set whose distribution is oblivious during training.

Below, we will first introduce the *phonetic and logogram encodings*. Then, we will study why using these as auxiliary inputs improves NMT and propose our hypothesis. Consequently, we will introduce one artificial method, *random clustering*, as a generalization to *text encoding*. Finally, we will demonstrate that all of these approaches significantly boost our NMT accuracy and robustness.

2 Background

NMT is an approach to MT using neural networks, which takes as an input a source sentence $(x_1, .., x_t, .., x_I)$ and generates its translation $(y_1, .., y_{t'}, .., y_{I'})$, where x_t and $y_{t'}$ are source and target words respectively (Bahdanau et al., 2015; Sutskever et al., 2014; Cho et al., 2014). NMT models with attention have three components, namely, an encoder, a decoder, and an attention mechanism. The encoder summarizes the meaning of the input sequence by encoding it with a bidirectional recurrent neural network (RNN). We apply the sequence-to-sequence learning architecture by Gehring et al. (2017), where the intermediate encoder and decoder states are calculated using convolutional neural networks (CNNs).

3 Phonetic Encodings

A phonetic algorithm is used to index words by their pronunciation. Taking as the input a sequence of words, we apply the phonetic algorithm to each word and output a sequence of encodings.

3.1 Soundex

Soundex is the most widely known phonetic algorithm for indexing names by sound, as pronounced in English, and avoids misspelling and alternative spelling problems. It maps homophones to the same representation so that they can be matched despite minor differences in spelling (Russel, 1918). It clusters the letter with exceptions. For example, the Soundex key letter codes ‘b, f, p, v’ to ‘1’, and ‘c, g, j, k, q, s, x, z’ to ‘2’, and ‘d, t’ to ‘3’.

3.2 NYSIIS

The New York State Identification and Intelligence System Phonetic Code, commonly known as NYSIIS, is a phonetic algorithm devised in 1970 as part of the New York State Identification and Intelligence System (Rajkovic and Jankovic, 2007). It produces better results than Soundex because it takes special care to handle phonemes that occur in European and Hispanic surnames.

3.3 MetaPhone

Metaphone (Philips, 1990) is another algorithm that improves on earlier systems such as Soundex and NYSIIS. The Metaphone algorithm is significantly more complicated than the others because it includes special rules for handling spelling inconsistencies and for looking at combinations of consonants in addition to some vowels.

3.4 Hanyu Pinyin

We also studied Hanyu Pinyin (Pinyin), the official romanization system for Standard Chinese in mainland China. Pinyin means ‘spelled sound’ and is usually used for the purpose of teaching Mandarin.

Algorithm 1 Random Clustering

Input: translation units
Parameter: baseline encoding
Output: mapping of units to clusters

```
perform a phonetic or logogram encoding as baseline
for each unique code in the baseline encoding vocabulary do
    $Z$ = “how many units are mapped”
    uniformly random sample $Z$ units to form a new cluster
end for
return
```

One Pinyin corresponds to multiple Chinese characters. One Chinese word is usually composed of one, two, or three Chinese characters.

3.5 Logogram Encoding: Chinese Wubi

The Wubizingxing (Wubi or Wubi Xing) is a Chinese character input method primarily for efficiently inputting Chinese text with a keyboard. The Wubi method is based on the structure, namely the decomposition of characters rather than their pronunciation. Every character can be written with at most 4 keystrokes including -, |, 丿, hook, and 丶.

4 Random Clustering

Driven by our hypothesis, which will be elaborated in Section 5, we further introduce an artificial way to encode the text in order to simulate “natural” encoding, i.e. phonetics and logogram. We call this random clustering as described in Algorithm 1. We cluster words (or characters) uniformly at random. The cluster size follows the distribution of how many words/characters are associated with each phonetic, here MetaPhone. For example, in Chinese, each Pinyin is a cluster, and, the number of clusters equals the number of unique Pinyins. Furthermore, each cluster’s size is the same as the number of characters mapped to each Pinyin.

5 Hypothesis

Hypothesis: *One phonetic representation (for example, Pinyin in Chinese) usually corresponds to characters/words that are semantically diverse.*

At first, this hypothesis may seem counterintuitive. However, it is made because, otherwise, humans would not be able to communicate effectively due to confusion. For example, red (Pinyin: ‘hong’) and green (Pinyin: ‘lv’) in Chinese appear in similar contexts. To reduce ambiguity in oral communication, it seems plausible to think that part of the development of phonetics is that one re-uses the same sound when context can be used to distinguish among multiple interpretations. For example,

Algorithm 2 c: Smoothed Convex Hull of Points

Input: points (embedded R^2 vectors) in a cluster
Parameters: β: threshold; r: radius
Output: The convex hull's vertices

```
for for each point do
    draw a circle with $r$
    if the total number of points in the circle is less than $\beta$ then
        remove this point
    end if
end for
return the convex hull
```

"fair" (county fair) versus "fair" (equitable).

How do we set up experiments to verify this?

We test our hypothesis using geometric interpretations of semantics, precisely, word embeddings (Bengio and Heigold, 2014). Intuitively, an embedding (Mikolov et al., 2013; Arora et al., 2016) preserves pairwise semantic distances, where two words/characters are close if they are semantically similar and far away otherwise. For instance, see the work of Zouzias (2010); Molitor (2017) about volume preserving embeddings, which formalizes the concept of this term. That is, if we have a set of words, for example, and all the words correspond to a Pinyin, then the points themselves may mean nothing, but the distances among the points are our focus. Typically in geometry, three points in space are sufficient to quantify a volume. We embed each word or characters from Chinese-English translation data (in Section 6) into 100 dimensions and then project this embedding into two dimensions using PCA. Algorithm 2 describes how we compute a smooth convex hull of points. The convex hull of a Pinyin is the convex hull of the embedding of all words or characters that are pronounced with this Pinyin.

Observations. Figure 2 shows all embedded Chinese characters in red dots, and black dots are the Chinese character(s) of one random Pinyin in each plot. We can see that characters with the same pronunciation tend to have distributed meaning - that is, well-distributed over the Euclidean plane.

In Figure 3, we measure the convex hull (the smallest convex set that contains all points - implemented in Matlab) of all characters. We exclude the outliers (blue dots) by removing all points that are encircled along with less than β other points in a ball of the radius of r. The first plot shows the hull enclosing characters of one random group (either cluster or Pinyin). The second plot shows adding characters of a second random group to the first group, and so on. The convex hull volume (here, 2D volume) increases as more groups are added. We can see that Soundex, Pinyin, and random grouping covers the space faster than the K-Means clustering when increasing the number of groups, namely the convex hull volume is greater for one or two clusters or Pinyins.

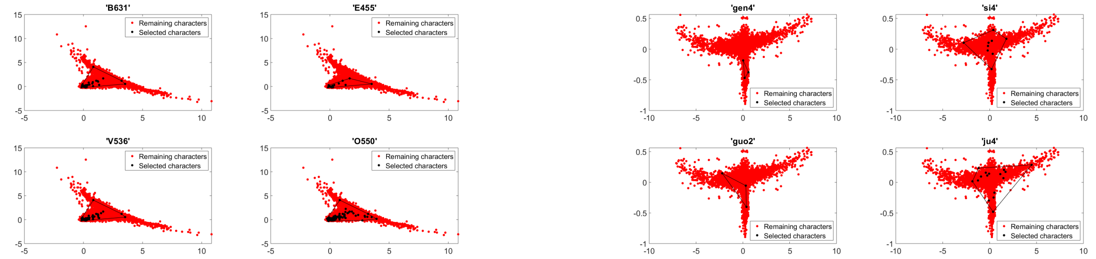
Quantitative verifications. These are carried out with three experiments. First, Figure 4 shows the empirical **CDF** of the convex hull volume of characters of each Pinyin, random clustering, and K-Means clustering, where the x-axes indicate the volume, and y-axes indicate the frequency. Random clustering and Pinyin grouping have a larger volume than K-Means, respectively. This means that for each group, Pinyin is slightly better distributed (more widespread) than uniformly random clustering, and both of these are better distributed than K-Means. This is quite interesting and is probably due to the isoperimetry of the uniform random sampling for these data points.

Second, we define the **concentration factor** as

$$\Gamma(\mathbf{p}_{1}^{K}{}_{1}^{I_k}) = \frac{\sum_{k=1}^{K}\|\mathbf{C}_k - \frac{\sum_{k=1}^{K}\mathbf{C}_k}{K}\|_2}{\sum_{k=1}^{K}\sum_{i=1}^{I_k}\|\mathbf{p}_{ki} - \mathbf{C}_k\|_2},$$

where $\mathbf{C}_k = \frac{\sum_{i=1}^{I_k}\mathbf{p}_{ki}}{I_k}$. $\mathbf{p}_{ki}$ is the $i-$th point in group k (either cluster or Pinyin). The smaller the value, the better distributed the points located in each cluster are over the whole space. The concentration factor Γ is 9350 for K-Means, 3.783 for Pinyin, 1.476 for the random clustering in Chinese; 3543K for K-Means, 0.3674 for Soundex, and 0.0191 for random clustering.

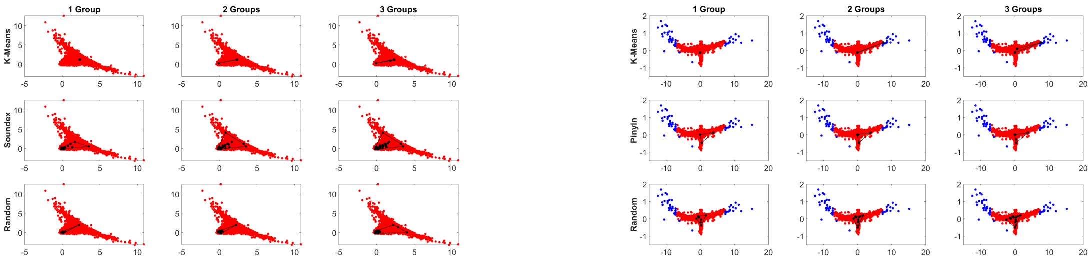
Finally, we define the **density measure** as in Algorithm 3, which intuitively seems to be a more robust test. For each point x in the smoothed convex hull of all words/characters, let $D_i(x)$ be the distance between x and the i-th nearest neighbor of x in the space X. We then look at either the maximum of $D_i(x)$ over all x, or the average. Choosing a larger i captures the "density" of the point-set at larger scales, which is a parameter that can be tuned to be more robust against noise. We numerically integrate the convex hull surface by randomly sampling the points, which are a linear combination of the convex hull corner weighted uniformly at random. The density result is shown in Table 2. This is consistent with the CDF in Figure 4. The Pinyin is most well-distributed, then the random clustering, after that the K-Means.



(a) Plot 1-4: words (black dots) of Soundex ‘B631’, ‘E455’, ‘V536’, ‘O550’respectively.

(b) Plot 1-4: all Chinese characters (black dots) of Pinyin ‘gen4’, ‘si4’, ‘guo2’, ‘ju4’respectively.

Figure 2: Same pronounced words/Chinese characters have distributed meaning in semantic space (red dots).



(a) K-Means, Soundex, and random clustering coverage speed by adding words (black dots) of group: The convex hull volume (black lines) of Soundex and random clustering cover the space (red dots) faster than K-Means by increasing the number of groups .

(b) K-Means, Pinyin, and random clustering coverage speed by adding characters (black dots) of each cluster or Pinyin: The convex hull volume (black lines) of Pinyin and random clustering cover the space (red dots) faster than K-Means by increasing the number of groups.

Figure 3: Coverage speed when adding groups one by one

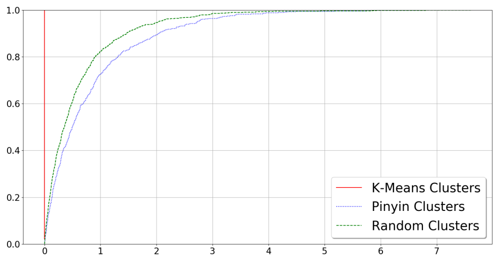


Figure 4: CDF of the convex hull volume of characters in each group (cluster or Pinyin) using three methods. K-Means has a very small convex hull volume in each group. The volume of Pinyin, and random clustering are close, but Pinyin is even larger.

	Max			Sum		
Method	1	2	3	1	2	3
K-Means	0.26	0.29	0.35	19.3	26.2	31.6
Random	0.18	0.21	0.22	15.5	19.3	21.2
Pinyin	0.08	0.21	0.22	7.19	19.4	20.9

Table 2: Density result (Converge threshold: 0.001; 1, 2, 3 nearest neighbour). The less the value, the more well-distributed in the entire space.

6 Experiments

Adding auxiliary information: First, we convert all the words or characters in the source sentences (in the training, development, and test set) into phonetic or other encodings. Then, those sequences of encodings are segmented with the Byte Pair Encoding compression algorithm (Sennrich et al., 2016b; Gage, 1994) (BPE). We learn a new embedding on this encoded training data only, for example, MetaPhone encodings. Afterward, the embedded vectors are either concatenated with the original sentences’ embedding (after BPE) or used alone to be fed into the encoder of the CNN neural translator. This decomposition results in a significant improvement over the baseline.

6.1 Datasets and Vocabularies

We carried out experiments for five translation directions from the IWSLT 2017 bilingual tasks: Chinese to English (ZH-EN), English to French (EN-FR), French to English (FR-EN), English to German (EN-DE), and German to English (DE-EN).

Algorithm 3 Density Measure

```
Input: Set of points $P$, clusters $Q$, nearest neighbor $i$
Output: Density $density$
$chosen = \{\}$
for $i = 1$ to $5$ do
    $cluster =$ get_random_cluster($Q$)
    for $word$ in $cluster$ do
        $chosen = chosen \cup$wordvector$(word)$
    end for
end for
Remove outliers from $P$ using $\beta = 0.3$ & $r = 10$
$X =$ convex_hull($P$)
$C$ = corner_points($X$)
for $i = 1$ to $m - 1$ do
    for $i = 1$ to $|C|$ do
        $q_i$ = random number between 0 & 1
    end for
    for $i = 1$ to $|C|$ do
        $p_i = \frac{q_i}{\sum_{k=1}^{|C|} q_k}$
        $C'_i = C_i * p_i$
    end for
    $hullpt = sum(C')$
    $density = 0$
    repeat
        $dist =$ KNN$(hullpt, chosen, i)$
        $density+ = dist$
    until $noChange$ is $true$
end for
```

Source	EN	EN	FR	DE
Target	FR	DE	EN	EN
Source(Words)	54k	51k	73k	119k
Target	73k	119k	54k	51k
Soundex	10k	10k	-	16k
NYSIIS	38k	36k	43k	99k
MetaPhone	36k	34k	37k	94k
W+Soundex	58k	55k	-	124k
W+NYSIIS	84k	80k	108k	206k
W+MetaPhone	83k	79k	104k	203k

Table 3: Vocabulary sizes before/after encodings.

ZH(W)/EN	Pinyin	Wubi	W+Pinyin	W+Wubi
94k/54k	1k	4k	95k	97k

Table 4: Vocabulary sizes in Chinese to English system.

We used the IWSLT 2017 training data (IWSLT, 2017), the development data combines test sets in 2013, 2014, and 2015, and the evaluation data is the 2017 test set.

Figure 3 and Figure 4 show vocabulary statistics on source/target tokenized text (Cettolo, 2015) before and after applying encodings (Turk and Stephens, 2010). We apply a BPE with 89K and 16K (Denkowski and Neubig, 2017) operations for FR and 89K for DE, and 18K operations for ZH, then we train an individual embedding on source/target jointly for each encoding.

6.2 Translation Results

For each encoding scheme, we carried out two experiments, one with only the encoded sentences and another one with the *source sentence concatenated with the encoded sentence*. For example, W+Soundex means the source sentence in words concatenated with all words converted into Soundex as the input to the Neural Networks. As the Soundex algorithm does not support French text, we do not have results for Soundex and W+Soundex for EN-FR. The translation results are evaluated with Bojar (2006).

Table 5 shows that encoding as an auxiliary input (concatenated with the original sentence) significantly improves the translation quality in all language directions. W+MetaPhone indicates adding MetaPhone to the word-based NMT baseline, which gives the best results for EN-FR and DE-EN, with an improvement of 1.71 and 1.2 in BLEU points, respectively. In our experiments, random clustering consistently improves over baselines on all languages. The non-uniform random clustering method in algorithm 1 achieves a higher BLEU score of 37.95% than a uniform random clustering after tuning on the cluster size. For EN-FR ($16k$ BPE operations) data in Table 5, we uniform randomly sample words for each cluster. We get the BLEU score of 37.74%, 37.77%, 37.38%, and 37.63% when setting the number of clusters to be 20%, 40%, 60%, and 80% of the vocabulary size (63615 words), i.e. the average cluster size to be 5, 2.5, 1.6, 1.25, respectively.

However, for most languages, the best codings are phonetic ones. Phonetic linguistic knowledge is helpful in MT, and we explained the underlying reason with our hypothesis of semantic diversity by phonetics. Linguistic information is typically language dependent, thus different phonetic algorithms serve better for certain languages. NYSIIS handles phonemes that occur in European and Hispanic surnames. Thus, it performs best in French. MetaPhone is a more advanced algorithm with spelling variations and inconsistencies, hence, it works best for English and German (both Germanic languages).

Table 7 shows the results of the ZH-EN translation system (BPE $18k$ operations). We apply Pinyin (Yu, 2016a), Pinyin segmented into letters, and Wubi encoding (Yu, 2016b). We achieve significant improvement over the baseline by adding auxiliary information: 0.87 BLEU points with Pinyin, 1.68 BLEU points with Pinyin in letters, and 1.11 BLEU points with Wubi, respectively. The randomly clustering on Chinese characters and on words both improve the baseline with 1.49 and 1.47 BLEU points, respectively. This is a larger improvement than that of the K-Means clustering.

Coding	FR-EN$_{(89k)}$	FR-EN$_{(16k)}$	EN-FR$_{(89k)}$	EN-FR$_{(16k)}$	DE-EN$_{(89k)}$	EN-DE$_{(89k)}$
Baseline: Words	35.01	36.21	34.37	36.78	27.79	25.12
Soundex	-	-	27.44	27.41	20.89	21.19
NYSIIS	30.87	31.22	31.36	31.06	25.76	18.90
MetaPhone	29.83	30.43	31.10	30.77	23.61	21.92
W+Soundex	-	-	35.88	36.80	27.54	24.97
W+NYSIIS	**35.44**	**37.33**	35.10	37.23	28.40	25.37
W+MetaPhone	35.09	37.04	**36.08**	**37.95**	**28.99**	25.00
W+random clustering	35.02	36.84	35.47	37.07	28.21	**25.58**

Table 5: Translation results in BLEU[%] using various codings. Training IWSLT 2017 data, development data is combined test 2013, 2014, and 2015 data and evaluated on test 2017 data. BPE operations: $89k$, $16k$.

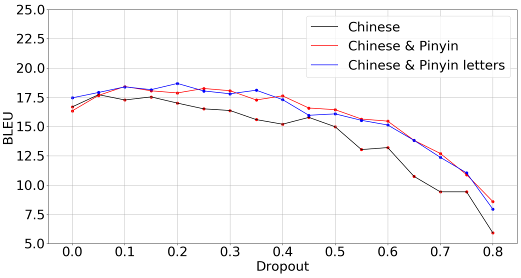


Figure 5: Tuning for dropout. The x axis indicate the drop out paramter value.

6.3 Model Complexity

We tune the drop out parameter for three experiments: Words, W+Pinyin, and W+Pinyin letters on ZH-EN. The drop out is set by default to 0.2, and the beam-size to 12. Figure 5 shows how translation accuracy changes. Our approach consistently performs better than the baseline systems. The peak BLEU score is achieved at drop out 0.05 for the baseline, but between 0.2 and 0.3 for our approach. This implies that adding auxiliary inputs will reduce the model complexity, indicated by a higher drop out parameter value opt for the best translation performance.

6.4 Training Speed

Table 6 shows the system training time (with BPE $89k$ operations and for ZH-EN $18k$). The total time (in thousands of seconds) is in the first column, and the time per epoch is in the second. Given the auxiliary information reduces the model complexity as in Section 6.3, the training becomes more efficient and needs a smaller number of epochs to converge. In most systems, the total training time with the auxiliary information is comparable to those without it, sometimes even less.

6.5 Robustness

To test the system robustness, we evaluated English-French systems ($89k$) in Table 5 that are trained on IWSLT'17. The tests are the out-of-domain and informal language data, the WMT'15 (Ondřej Bojar and Turchi, 2015) News and MTNT test sets. MTNT'18 is a dataset in the informal domain, recently released by Michel and Neubig (2018) for the robustness task in WMT'19 (WMT, 2019), which includes MTNT'18 test data and MTNT'19 test data. Note that unlike the robustness task itself, we did not use MTNT training data when building systems. We aim to verify how the system would behave in a new domain that was entirely unknown during the system building process - aligning with a real-life scenario. As in Table 8, all of our approaches achieved higher accuracy, showing more robustness in this experiment. +MetaPhone outperforms all other encoding methods for all the three out-of-domain test sets and improves over the baseline by up to 1 BLEU points.

Table 5 also has results for system trained on the MTNT'18 data and tested on both MTNT'18 and MTNT'19 test sets as well as the out-of-domain WMT'15 test data. +MetaPhone outperforms all other systems and improves over the baseline by about 5 BLEU points.

7 Related Work

Phonological rules or constraints have been previously applied to tasks such as word segmentation (Hayes, 1996; Johnson et al., 2015). Phonetics involves gradient and variable phenomena, whereas phonology is characteristically categorial and far less variable. Instead of optimizing towards phonological constraints, we directly learn from phonetic data and discover hidden phonetic features to optimize NMT performance.

Coding	FR-EN	EN-FR	DE-EN	EN-DE
Words (W)	2.92/112	2.84/123	2.57/88.6	2.91/112
Soundex	-	3.99/133	2.27/83.9	3.02/121
NYSIIS	2.02/101	3.25/125	2.22/79.4	3.11/111
Metaphone	2.72/109	2.95/123	1.75/83.2	3.09/115
W+Soundex	-	3.25/155	2.00/111	3.66/141
W+NYSIIS	1.48/148	3.12/149	2.07/94.3	3.27/131
W+Metaphone	**1.12**/140	3.92/151	**1.98**/98.8	3.29/132

Coding	ZH-EN
Words (W)	1.82/79.0
Pinyin	2.74/85.8
Wubi	2.60/81.2
W+Pinyin	2.95/114
W+Wubi	3.06/110

Table 6: Training time for each system. It shows total time [K]/ average epoch time in seconds.

Coding	ZH-EN
Words	17.00
Wubi	14.43
Pinyin	15.57
Pinyin in letters	12.51
W+Wubi	18.11
W+Pinyin	17.87
W+Pinyin in letters	**18.68**
K-Means characters	14.57
random clustering words	17.35
random clustering characters	15.84
W+K-Means words	17.86
W+random clustering words	18.47
W+random clustering characters	**18.49**

Table 7: Translation results in BLEU[%] for ZH-EN.

Training Data	Coding	MTNT'18	MTNT'19	WMT'15
IWSLT'17	Words (W)	13.94	10.59	12.05
	W+Soundex	13.46	10.53	12.35
	W+Metaphone	**14.44**	**11.60**	**12.65**
	W+NYSIIS	13.81	11.21	12.14
MTNT'18	Words (W)	10.36	7.10	8.64
	W+Soundex	10.40	11.59	12.73
	W+Metaphone	**10.58**	**10.67**	**13.39**
	W+NYSIIS	10.98	12.65	14.53

Table 8: Robustness results in BLEU[%]: test on MTNT, WMT; train on IWSLT for EN-FR.

Discriminatively learning phonetic features has demonstrated success in various Language technology applications. Huang et al. (2004) used phonetic information to improve the named entity recognition task. Bengio and Heigold (2014); Zhu et al. (2018) integrate speech information into word embedding and subword unit models, respectively. Du and Way (2017) converted Chinese characters to subword units using Pinyin to alleviate the unknown words. Our work aims to improve NMT overall rather than to only translate unknown Chinese words. We are the first to introduce several phonetic algorithms: Soundex, NYSIIS, MetaPhone; and Logogram, Wubi to improve NMT. We also develop new algorithms driven by an empirically verified observation, which works for all languages in any NMT framework.

Leading research has investigated auxiliary information to NLP tasks, such as polysemous word embedding structures by Arora et al. (2016), factored models by García-Martínez et al. (2016); Sennrich and Haddow (2016), as well as compiling various features as in Kobus et al. (2017) and Sennrich et al. (2016a). In this paper, we focus on the introduction of phonetic encoding and random clustering and demonstrate that our approaches are effective even when applied in a simple way (namely, concatenation without the help of a factor model). Treating NMT as a black-box can be beneficial when experimenting with different NMT models such as CNN, seq2seq, and attention-based one.

Closely related, but independent to this work, is the approach of word segmentation or character based NMT (Chung et al., 2016), which focuses on the decomposition of the translation unit. Smaller text granularity helps in unseen word forms and tokenization challenges (Ling et al., 2015), while more extended translation units reduce model complexity and input lengths (Lee et al., 2017). Finding the optimal granularity when feeding information to an NMT is undoubtedly impressive, but stratifying the translation unit does not necessarily only take place at the next level (in the form of character or word sequences). We take a different angle and view MT input as an information source encoded in various forms. We study the source sentence representations other than text such as phonetic encodings, which works surprisingly well when combined with word segmentation methods.

8 Conclusions

We introduce phonetic and logogram encodings to convert foreign text into phonetic and logogram forms. We deploy them into NMT systems and significantly improve NMT translation quality and robustness. When analyzing this improvement, we introduce and verify our hypothesis of semantic diversity by phonetics. Driven by this hypothesis, we further introduce the random clustering which also enhance the NMT accuracy and robustness.

References

Simona Amenta, Marco Marelli, and Simone Sulpizio. 2017. From sound to meaning: Phonology-to-semantics mapping in visual word recognition. *Psychonomic Bulletin & Review*, 24(3).

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6.

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*.

David I Beaver, Brady Clark, Edward Stanton Flemming, T Florian Jaeger, and Maria Wolters. 2007. When semantics meets phonetics: Acoustical studies of second-occurrence focus. *Language*, 83.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Samy Bengio and Georg Heigold. 2014. Word embeddings for speech recognition. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Ondřej Bojar. 2006. Multibleu script. https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl. Online; accessed 29 January 2019.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors. 2018. *Proceedings of the Third Conference on Machine Translation*.

Marine Carpuat, Hal Daumé, Alexander Fraser, Chris Quirk, Fabienne Braune, Ann Clifton, Ann Irvine, Jagadeesh Jagarlamudi, John Stanley Morgan, Majid Razmara, Ales Tamchyna, Katharine Henry, and Rachel Rudinger. 2013. Domain adaptation in machine translation : Final report. *Technical report*.

Mauro Cettolo. 2015. Chinese char segmenter for iwslt evaluation campaigns. http://hltshare.fbk.eu/IWSLT2015/chineseText2Chars.pl. Online; accessed 29 January 2019.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Jinhua Du and Andy Way. 2017. Pinyin as subword unit for chinese-sourced neural machine translation. In *Irish Conference on Artificial Intelligence and Cognitive Science*.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2).

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *CoRR*, abs/1609.04621.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *Computing Research Repository*, abs/1705.03122.

Bruce Hayes. 1996. Phonetically driven phonology: The role of optimality theory and inductive grounding. rutgers optimality archive.

Fei Huang, Stephan Vogel, and Alex Waibel. 2004. Improving named entity translation combining phonetic and semantic similarities. In *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

IWSLT. 2017. Homepage of iwslt evaluation 2017.

Jing Jiang. 2008. A literature survey on domain adaptation of statistical classifiers. *Technical report, University of Illinois at Urbana-Champaign*.

Mark Johnson, Joe Pater, Robert Staubs, and Emmanuel Dupoux. 2015. Sign constraints on feature weights improve a joint model of word segmentation and phonology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. *CoRR*, abs/1902.01509.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2018. Robust neural machine translation with joint textual and phonetic embedding. *arXiv preprint arXiv:1810.06729*.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

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

Mathieu Molitor. 2017. Remarks on the space of volume preserving embeddings. *Differential Geometry and its Applications*, 52.

Christian Federmann Barry Haddow Matthias Huck Chris Hokamp Philipp Koehn Varvara Logacheva Christof Monz Matteo Negri Matt Post Carolina Scarton Lucia Specia Ondřej Bojar, Rajen Chatterjee and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of EMNLP Tenth Workshop on Statistical Machine Translation, Shared Task Papers*.

Pedro M Paz-Alonso, Myriam Oliver, Garikoitz Lerma-Usabiaga, Cesar Caballero-Gaudes, Ileana Quiñones, Paz Suárez-Coalla, Jon Andoni Duñabeitia, Fernando Cuetos, and Manuel Carreiras. 2018. Neural correlates of phonological, orthographic and semantic reading processing in dyslexia. *NeuroImage: Clinical*, 20.

Lawrence Philips. 1990. Hanging on the metaphone. *Computer Language*, 7(12 (December)).

P Rajkovic and D Jankovic. 2007. Adaptation and application of daitch-mokotoff soundex algorithm on serbian names. In *XVII Conference on Applied Mathematics*.

Robert C. Russel. 1918. A method of phonetic indexing. *Patent no. 1,261,167*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*.

James Turk and Michael Stephens. 2010. Phonetic encoding python toolkit. https://pypi.org/project/jellyfish/. Online; accessed 29 January 2019.

Lorraine K Tyler, J Kate Voice, and Heien E Moss. 1996. The interaction of semantic and phonological processing. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*.

Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. *CoRR*, abs/1902.09508.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Computing Research Repository*, abs/1706.03762.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Xiaojuan Wang, Rong Zhao, Jason D Zevin, and Jianfeng Yang. 2016. The neural correlates of the interaction between semantic and phonological processing for chinese character reading. *Frontiers in psychology*, 7.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

WMT. 2019. Shared task: Machine translation robustness. http://www.statmt.org/wmt19/robustness.html. Online; accessed 18 May 2019.

Lx Yu. 2016a. Pinyin python toolkit. https://pypi.org/project/Pinyin/. Online; accessed 29 January 2019.

Lx Yu. 2016b. Wubi python toolkit. https://pypi.org/project/jellyfish/. Online; accessed 29 January 2019.

Xuan Zhang, Gaurav Kumar Pamela Shapiro, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Wenhao Zhu, Xin Jin, Jianyue Ni, Baogang Wei, and Zhiguo Lu. 2018. Improve word embedding using both writing and pronunciation. *PloS one*, 13(12).

Anastasios Zouzias. 2010. Low dimensional euclidean volume preserving embeddings. *arXiv preprint arXiv:1003.0511*.