Product Review Translation using Phrase Replacement and Attention Guided Noise Augmentation

Kamal Kumar Gupta, Soumya Chennabasavraj,† Nikesh Garera,† and Asif Ekbal

Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India
†Flipkart, India
kamal.pcs17,asif@iitp.ac.in
†soumya.cb,nikesh.garera@flipkart.com

Abstract

Product reviews provide valuable feedback of the customers, however, they are available today only in English on most of the e-commerce platforms. The nature of reviews provided by customers in any multilingual country poses unique challenges for machine translation such as code-mixing, ungrammatical sentences, presence of colloquial terms, lack of e-commerce parallel corpus etc. Given that 44% of Indian population speaks and operates in Hindi language, we address the above challenges by presenting an English–to–Hindi neural machine translation (NMT) system to translate the product reviews available on e-commerce websites by creating an in-domain parallel corpora and handling various types of noise in reviews via two data augmentation techniques, *viz.* (i). a novel phrase augmentation technique (PhrRep) where the syntactic noun phrases in the sentences are replaced by the other noun phrases carrying different meanings but in the similar context; and (ii). a novel attention guided noise augmentation (AttnNoise) technique to make our NMT model robust towards various noise. Evaluation shows that using the proposed augmentation techniques we achieve a 6.67 BLEU score improvement over the baseline model. In order to show that our proposed approach is not language-specific, we also perform experiments for two other language pairs, *viz.* En-Fr (MTNT18 corpus) and En-De (IWSLT17) that yield the improvements of 2.55 and 0.91 BLEU points, respectively, over the baselines.

1 Introduction

Product reviews written by the users on e-commerce websites are useful to get the feedback about the products and provide valuable insights to the user for making the buying decision. The product reviews available on different e-commerce websites are mainly in English language. India is a multilingual country with great linguistic and cultural diversities. There are 22 officially spoken languages, and many of them such as Hindi, Bengali, etc. come into the top 10 most spoken languages all over in the world. Since English is not a first language in India and most of the population (approximately, 68.9%)[1] from the rural areas do not have the proper understanding of English language,

[1]http://mohua.gov.in/cms/urban-growth.php

Source (A)	osm product.i really love it. osm camera quality...nice one
Reference	बहुत बढ़िया प्रॉडक्ट. मुझे यह पसंद है. बहुत बढ़िया कैमरा क्वालिटी... अच्छा है
(Transliteration)	bahut badhiya prodakt. mujhe yah pasand hai. bahut badhiya kaim kvaalitee... achchha hai
Gen-NMT	ओसम उत्पाद. मैं वास्तव में इसे प्यार करता हूँ. ओसम कैमरा गुणवत्ता... अच्छा एक
(Transliteration)	osam utpaad. main vaastav mein ise pyaar karata hoon. osam kaimara kvaalitee... achchha hai
Source (B)	NYC product,and cloth quilty is too good
Reference	अच्छा प्रॉडक्ट, और कपड़े की क्वालिटी बहुत बढ़िया है
(Transliteration)	achchha prodakt, aur kapade kee kvaalitee bahut badhiya hai
Gen-NMT	NYC उत्पाद, और कपड़ा रजाई बहुत अच्छा है
(Transliteration)	nyc utpaad, aur kapada rajaee bahut achchha hai
Source (C)	Nice Mobile and value for money 😇😇
Refernce	अच्छा मोबाइल और पैसा वसूल 😇😇
(Transliteration)	achchha mobail aur paisa vasool 😇😇
Gen-NMT	अच्छा मोबाइल और पैसे के लिए मूल्य money
(Transliteration)	achchha mobail aur paise ke lie mooly money

Table 1: Sample outputs for En→Hi translation from sources with various inconsistencies. Here, **Gen-NMT:** Generic NMT **(A)** Abbreviations and colloquial terms, **(B)** Spelling mistake and **(C)** Emojis

it becomes difficult for them to read a review or write a review in English with proper vocabulary and grammar. This makes the availability of product reviews in vernacular languages essential for the vast majority of Indian e-commerce customers. However, building an automated translation system for the large amount of reviews poses unique challenges to the machine translation community.

We illustrate some of the challenges with examples as shown in Table 1. In example A, the word *osm* appears as a short form of the word *awesome*; also there is no *space* between the words *product* and *i*. The model is not able to translate these correctly. Similarly, in example B, *NYC* and *quilty* are the short forms and misspelled versions of the words *nice* and *quality*, respectively. Presence of emojis in example-C also causes translation difficulty.

We address the above challenges with the main contributions or attributes of our work as follows:

- We build an NMT system for product reviews in low-resource scenarios. To the best of our knowledge, this is the very first attempt towards building a machine translation system for English to Indian language review translation.
- We build data resources by crawling reviews from an e-commerce portal, translate them into Hindi using our in-house open domain English-Hindi MT system, and perform manual verification for the correctness (c.f. Section 3.1).
- We introduce novel data augmentation techniques to handle the noise and the scarcity of in-domain training data as follows:

    1. We introduce a novel similar phrase replacement technique (PhrRep) which generates more diverse synthetic parallel samples compared to the word augmentation techniques (c.f. Section 4.3).

2. We use Part-of-Speech (PoS) guided word embedding based and context aware word augmentation techniques for synthetic data creation (c.f. Section 4.1 and Section 4.2), and show that our proposed PhrRep approach significantly outperforms the word based augmentation methods.
3. We introduce a novel attention guided noise augmentation (AttnNoise) technique to make the NMT model robust towards noisy inputs (c.f. Section 5.1). We show that AttnNoise method significantly outperforms the random noise injection (RndNoise) techniques.

2 Related Work

There are two main challenges for translating the product reviews, *viz.* (i). non-availability of parallel corpus; and (ii). noisy sentences in product and/or service reviews. Machine translation with noisy text is, itself, a very challenging task. The typical noises that pose challenges for machine translation include improper grammatical structures, misspellings, punctuation, emojis etc (c.f. Section 3.1) (Michel and Neubig, 2018). In the literature, there are a few works concerning the noise in the text and to increase the robustness of the translation model. Michel and Neubig (2018) presented a noisy dataset and discussed the challenges of noisy contents.

Belinkov and Bisk (2018) and Karpukhin et al. (2019) showed that small noise in the input text can reduce the quality of translation. To improve the robustness of the translation model they introduced synthetic errors like character swapping, deletion and insertion in the corpus. Vaibhav et al. (2019) also inserted synthetic noises and back-translated noise in the original corpus. Apart from the spelling distortion, to make the model immune to the grammatical errors, Anastasopoulos et al. (2019) augmented training data with the grammatical errors. They focused on articles, prepositions, subject-verb agreements etc. Considering the challenges, Berard et al. (2019) analyzed the performance of NMT model over a small French-English corpus of restaurant reviews. Unlike this, we do not inject any random noise, rather we introduce an attention guided noise augmentation (AttnNoise) technique to insert the synthetic noise at the source (English) side.

To address the second challenge related to the availability of training data, we make use of the data augmentation techniques to increase the training samples and noise handling techniques to increase the robustness of the model. Fadaee et al. (2017) replaced the common words by rare words to provide better evidence and contexts for the rare words. Gao et al. (2019) introduced a soft contextual augmentation method where a word's embedding is replaced by a weighted average of its similar words. Kobayashi (2018) used a bi-directional language model to predict the replacement by using the sentence context. Wu et al. (2019) used the BERT (Bidirectional Encoder Representations from Transformers) Devlin et al. (2019) model to predict the randomly masked word. Inspired by Wu et al. (2019), we mask the noun and adjective words in the source sentence and predict the appropriate nouns and adjectives as substitutes based on the sentence context. We introduce a phrase replacement based data augmentation technique (PhrRep) to replace the whole syntactic noun phrase (multiple words in a single attempt) with other diverse but contextually similar noun phrases.

3 Parallel Corpus Creation

In this section we describe the steps followed for parallel corpus creation and the necessary statistics.

3.1 Crawling reviews and challenges in pre-processing

We crawl English product reviews from the e-commerce portal, Flipkart. Product reviews are user generated contents and contain various noises (inconsistencies) as shown in Table 1.

A. Systems	Sentences	%Increase
Baseline (Human translated)	19,457	
Base+BT	122,570	
Base+BT+WDA	297,392	142.6%
Base+BT+CDA	369,765	201.7%
PhrRep	306,475	150%
Development Set (Human trans.)	599	
Testset (Human trans.)	2,539	
B. Systems	**Sentences**	
Base	1,561,840	
PhrRep	1,701,704	8.9%
Development Set	520	
Testset (newstest2014)	2,507	
C. Systems	**Sentences**	
Base	300,000	
PhrRep	488,501	62.83%
Development Set	1,500	
Testset (newstest2015)	1,500	
D. Systems	**Sentences**	
Base	223,021	
PhrRep	312,504	40.12%
Development Set	885	
Testset (IWSLT2017)	1,138	

Table 2: Parallel corpus size. Here, **A:** Product review dataset, **B:** IIT-Bombay English-Hindi dataset Kunchukuttan et al. (2018), **C:** UN-Corpus English-French dataset Ziemski et al. (2016) and **D:** IWSLT2017 English-German dataset.

3.2 Pre-processing

We remove the emojis from the English sentence by providing their unicode range using regular expressions. Any character having repetition of more than two times is trimmed and then checked for its compatible correct word using spell-checker[2] and a list provided by Facebook[3] Edizel et al. (2019). Writing the complete sentence in upper case is also very common in user generated content (i.e. *NICE PHONE IN LOW BUDGET*). Normalization is done to convert all such instances into the lower case. Since we focus on the product reviews data, we make the first character of brand's name[4] (Google, Moto, Nokia etc.) as capital. After the pre-processing steps as mentioned above (emoji removal, character repetition, casing etc.), we found that approximately 62.3% sentences from the total crawled sentences are correct.

3.3 Gold Corpus Creation by Human Post-editing

After pre-processing, we obtain 22,595 standard English sentences as mentioned in Table 2. Instead of translating sentences from scratch, we use our in-house judicial domain system to generate the initial target sentences and post-edit. It is trained for English-Hindi translation using 0.45 million parallel judicial domain samples and additional English-Hindi corpus Kunchukuttan et al. (2018) having 1.6 million parallel samples. It achieves 55.67 BLEU (En-to-Hi) points on our in-house judicial domain testset. After translation into Hindi, manual verification for the correctness of the translation is done by three language experts. The experts are post-graduates in linguistics and have good command in Hindi and English both. The experts read the English sentences and their Hindi translation. They were instructed to make the correction in the sentences, if required. The human post-edited parallel corpus as shown in Table 2 is divided into training, development and test set consisting of 19,457, 599 and 2,539 parallel sentences, respec-

[2] `https://pypi.org/project/pyspellchecker/`

[3] `https://github.com/facebookresearch/moe/tree\\/master/data`

[4] `https://en.wikipedia.org/wiki/List\_of\_mobile\_phone\_brands\_by\_country`

Sentence	There are many **offers** for this smartphone
WDA	There are many **provides** for this smartphone
CDA	There are many **applications/designs/models** for this smartphone
PhrRep	There are **multiple features in my new smartphone**

Table 3: Samples generated using WDA, CDA and PhrRep approaches.

tively. The gold standard corpus, and the parallel corpus created synthetically is made available[5]. We also crawl the Hindi sentences and back-translate them into English. We build a Hindi–to–English NMT model to back-translate the crawled Hindi sentences. We use the IIT Bombay Hindi-English general domain parallel corpus Kunchukuttan et al. (2018) to train a Hindi–to–English NMT model, and then fine-tune it over the human post-edited review domain parallel corpus. The fine-tuned Hindi–to–English NMT model is used to back-translate the crawled monolingual Hindi sentences into English. These back-translated (BT) English-Hindi synthetic parallel sentences are augmented with the human post-edited parallel sentences and referred to as 'Base+BT', shown in Table 2.

4 Data Augmentation

We further enrich the training corpus (in low-resource language) following the data augmentation techniques as discussed below.

4.1 Word Embedding based Data Augmentation (WDA)

Let us take one example: **Original sample:** This ***phone*** is not ***good***. and **New sample:** This ***handset*** is not ***nice***.
In the original sample, the words *'phone'* and *'good'* are replaced by their most semantically close words *'handset'* and *'nice'*, respectively, based on the cosine similarity between their word embeddings. To reduce the alignment complexity, we choose noun and adjective words as the replacement candidates because:

- Hindi is morphologically richer than English. One English verb token may be aligned to more than one Hindi tokens. But nouns and adjectives are most likely to generate only one Hindi token. For example: translation of word '*started* (verb)' (1 token) can be 'शुरू कर दिया' 'shuroo kar diya' (3 tokens) or 'शुरू किया' 'shuroo kiya' (2 tokens). Here, we see that for the word '*started*', more than one translations possible with different token lengths.

To select the noun and adjectives for replacement, we use NLTKLoper and Bird (2002) Part-of-Speech (PoS) tagger for the English sentences. A word2vec skip-gram model[6] Mikolov et al. (2013) is trained using the WMT14 monolingual English dataset and English sentences from the gold corpus. Now for all the noun and adjective words, we find the most similar words using our trained word2vec model. The words having the cosine similarity more than 0.75 will be considered as the substitutes. A mapping dictionary is created with the triplet consisting of the 'original English word', 'its replacement English word' and 'Hindi translation of the replacement word'. Now using the mapping dictionary, the tokens in the original corpus are replaced. Source-target word alignment information using GIZA++ tool (Och and Ney, 2003) is used to replace the aligned Hindi tokens in the Hindi side. But WDA does not guarantee to replace the original word with a similar context word as shown by an example in Table 3.

[5] https://www.iitp.ac.in/~ai-nlp-ml/resources/data/review-corpus.zip
[6] https://code.google.com/archive/p/word2vec/

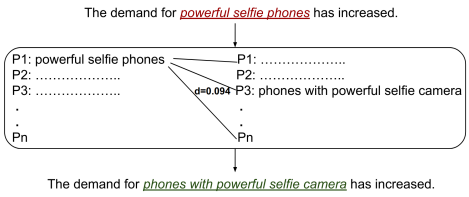

Figure 1: Synthetic sample generation using phrase replacement (PhrRep)

4.2 Context Aware Data Augmentation (CDA)

Wu et al. (2019) used a BERT based method which predicts the substitution for the randomly masked word. Here, we mask only nouns and adjective words. Similar to 4.1, *noun* and *adjective* words in the source sentence are masked, and their appropriate substitutes are predicted based on the sentence context. We use the 'bert-base-uncased' pre-trained model for the prediction which is trained using the default hyper parameters: 12 layers, 728 hidden units and 12 attention heads. Here, we also find the replacements for nouns and adjectives only. A list of noun and adjective tokens is created and in each English sentence, we mask the tokens by replacing the tokens with '[MASK]' which are in the list.

Now, the masked sequence is passed through the trained BERT model. Since BERT contains the bidirectional sequence information, it can predict the most appropriate token for position *'i'* by considering the previous and next context words within the sentence. For generating more augmented samples, we take the top 3 predicted words for position *'i'* and generate different samples. We use Giza++ alignment information to obtain the aligned positions between English and Hindi sentences, and the translated Hindi word of the newly predicted English word is placed at the Hindi side too. A mapping dictionary similar to WDA is needed here to obtain the parallel counterpart of an augmented word. Using CDA, multiple replacements can be found for a single masked token based on the context (because here no fixed mapping dictionary is used). Also, the substitute token suits the syntactic and semantic structure of the sentence. In Table 3, we can see in the example, "There are many **offers** for this smartphone", 'applications', 'designs' and 'models' are predicted at the place of original hidden word 'offers'.

4.3 Data Augmentation using Phrase Replacement (PhrRep)

Here, we introduce a novel approach for data augmentation using similar phrase replacement strategy. The method generates more diverse samples (a phrase of multiple tokens is replaced with similar phrases of different token lengths) in a single attempt. Unlike the previous word augmentation techniques Fadaee et al. (2017); Gao et al. (2019), here we replace a noun phrase (NP) with its semantically similar noun phrase (NP). To extract NP from the English sentences, we use the Stanford parser[7] and obtain the corresponding constituency trees. To reduce the complexity in alignment mapping and trivial replacements, we filter out very large (>8 tokens) and very short (<3 tokens) NPs. Here, we refer to the replacements of very small NPs as trivial replacements since most likely they are already part of larger NPs, and get replaced when larger NPs are replaced. To find the similarity among phrase embeddings, we use a BERT based sentence-transformer[8] Reimers and Gurevych (2019).

For an original phrase P_{oi}, its similar phrase P_{si} is:

$$P_{si} = P_j, [i = (1,....,n) \text{ and } j = (1,....,n)] \quad (1)$$

[7] https://nlp.stanford.edu/software/lex-parser.shtml

[8] https://github.com/UKPLab/sentence-transformers

$$P_j = \arg\min_j d(h_i, h_j) \quad (2)$$

n is the number of NPs. h is the hidden representation of the phrases. d represents the Euclidean distance between the two vectors. Equation 2 returns the index j of a phrase having minimum Euclidean distance d with the phrase at index i. As shown in equation 1, the respective phrase P_j at index j is the most similar phrase to the original phrase P_{oi}. Figure 1 shows the mapping of the original phrase *'powerful selfie phones'* with phrase *'phones with powerful selfie camera'* having the Euclidean distance d = 0.094, minimum in the distances with all the other phrases. Further, Hindi counterparts of the English NPs are extracted from the original parallel data itself using the alignment information.

5 Noise Augmentation

We create a noisy copy of the original corpus. To deal with character missing, article missing, punctuation missing and the dropping offs of starting noun-pronouns, we introduce various noise in the original training corpus. In similar ways to the prior works Vaibhav et al. (2019); Anastasopoulos et al. (2019), we also drop the characters randomly from the source (English) side, but with some additional rules.

- It is observed in the reviews that 'vowels' are most likely to be dropped by the users. For example, for a word 'phone', *'phne'* and *"phon'* are most likely to occur compared to the *"pone'* and *"phoe'*. So in each English sentence, along with dropping the random characters we make sure that vowels are also dropped in a few words.
- We randomly drop the articles 'the', 'a' and 'an' from the English side because we observe that in reviews users often drop the articles.
- Users often write reviews without mentioning the starting nouns or pronouns. We drop the starting nouns and pronouns randomly from the sentences. The PoS tagger was used to mark the words to be dropped. For example, "was planning to buy this" or "am happy with the phone".

Here, when we pick the tokens randomly for noise injection (char drop) we call it *random noise* (RndNoise) insertion. All these noises are introduced into a copy of the original corpus. It is then augmented with the original corpus. This provides noisy and correct source versions for a target sentence.

5.1 Attention Guided Noise Augmentation (AttnNoise)

	x_1	x_2	x_{n-1}	x_n
y_1	W11	W12	.	W1n
y_2	W21	W22	.	W2n
y_{m-1}	.	.	.	Wm-1.n
y_m	Wm1	Wm2	.	Wmn
Sum=(W11+..+Wn)	**W1**	**W2**	**Wn-1**	**Wn**
AvgAttn=Sum/m	**AvgW1**	**AvgW2**	**AvgWn-1**	**AvgWn**

Table 4: Attention weight matrix during source-to-target inference. Here, Wij: attention weight between i^{th} target token and j^{th} source token

Most of the existing literature Vaibhav et al. (2019); Anastasopoulos et al. (2019) introduced noise in the training data by randomly dropping characters from the source words. To make our model robust towards misspellings, article missing, punctuation and word missing, we also drop the words or introduce the character inconsistencies

in words. Instead of executing these randomly, we follow a guided approach to drop a word or character(s) from these words. To do this, we take the help of attention weights between the source-target pairs. We call this technique as *attention guided noise augmentation* (AttnNoise).

Algorithm 1 Attention guided noise augmentation (AttnNoise)

Notations: $\boldsymbol{s_i}$=$\{x_1, x_2, ..., x_n\}$, i^{th} sequence.
$\boldsymbol{AvgAttn_i}$: list of avg. attention weights of tokens in s_i
$\boldsymbol{lProb_i}$: list of probability (occurrence frequency) of tokens in s_i
$\boldsymbol{sN_i}$: i^{th} noisy source sequence
lMinAttn: indexes of bottom 10% min values in $AvgAttn_i$.
lMaxAttn: indexes of top 25% max values in $AvgAttn_i$.
lMaxProb: indexes of top max 50% values in $lProb_i$.
ind: index of a token in s_i.
$\boldsymbol{x_j}$: token at j^{th} position in s_i.

```
procedure NOISE(s_i, AvgAttn_i, lProb_i)
    for j ∈ 0, ..., len(s_i) do                          ▷ for each token
        if ind[x_j] ∉ lMinAttn then
            if ind[x_j] ∈ lMaxAttn then
                sN_i.append(dropChar(x_j))
            else
                sN_i.append(x_j)
        else if ind[x_j] ∉ lMaxProb then
            dropWord(x_j)
        else
            sN_i.append(dropChar(x_j))
    return (sN_i)
procedure WORD-PROB(s_i, S)
    for k ∈ 0, ..., len(s_i) do                          ▷ for each token
        p = (#x_k in S / #all tokens in S)
        lProb_i.append(p)
    return (lProb_i)
```

We have a corpus D with parallel pairs *[S,T]*, where S and T are the collection of source and target sentences, respectively. s_k and t_k represent a pair of k^{th} source and target sequences in S and T, respectively. Each $s_k = \{x_1,x_2,...,x_n\}$ is a sequence of n source tokens and $t_k = \{y_1,y_2,...,y_m\}$ is a sequence of m target tokens. We calculate the average attention for each source token as shown in Table 4. All the attention heads are considered here. We drop a fraction of tokens from the source sequence having low average attention weight, and introduce noise in a fraction of tokens having high average attention weight. Method *NOISE* in Algorithm 1 describes the steps involved in the AttnNoise. To decide if a token comes under the low or high attention weight category, we choose some *percentage* value as the threshold. For example, we have a list $AvgAttn_i$ of source sequence s_i which has 15 tokens. For our experiments, we empirically decide to drop the bottom 10% of total tokens in s_i having minimum average attention weight (i.e. 10% of 15 = 2 tokens, so we drop 2 tokens having the lowest weights). Similarly, top 25% of tokens in s_i having high weights are made noisy by dropping the characters from them.

We also calculate the occurrence probability of the source tokens of s_i using the method *WORD-PROB* in Algorithm 1 to know whether any token is frequent or rare in the vocabulary. A token with less occurrence probability is said to be rare and we do not drop any rare token even if it has the low average attention weight. The rare

	Systems	BLEU	TER
En→Hi	Base	34.36	46.23
(Review)	Base+BT	35.19	45.10
	+Fadaee et al. (2017)	38.54	41.69
	+WDA	38.67	40.28
	+CDA	39.66	39.65
	+CDA+RndNoise	40.14	40.36
	+PhrRep+RndNoise	**40.61**	**38.79**
	+PhrRep+AttnNoise	**41.03**	**37.92**
En→Fr	Base	20.83	66.74
(MTNT18)	PhrRep	**22.75**	**64.16**
	+AttnNoise	**23.38**	**63.37**

	System	BLEU	TER
1.A	Base	15.42	71.46
	PhrRep	**16.56**	69.62
1.B	Base	22.47	62.84
	PhrRep	**22.69**	61.92
1.C	Base	4.49	89.14
	PhrRep	**6.24**	86.44
En→Fr	Base	19.36	67.15
(newstest2015)	PhrRep	**20.91**	65.83
En→De	Base	18.83	65.91
(IWSLT 2017)	PhrRep	**19.74**	64.38
En→Fr	Base	21.77	63.83
(IWSLT 17)	PhrRep	**22.52**	61.87

Table 5: BLEU and TER scores of different systems for different datasets of English-Hindi, English-French and English-German language pairs. Also for En→Hi translation: **(1.A)** Trained on IITB-Hin-Eng corpus and tested over newstest2014, **(1.B)** Trained on IITB-Hin-Eng corpus and tested over product review testset, **(1.C)** Trained on product review corpus and tested over newstest2014.

tokens correspond to those having high attention weights, and instead of dropping these from the source sequence, we insert noise into it. To prevent the dropping of any rare word having low attention weight, we increase the *percentage* value for the threshold. Here, the top 50% tokens in s_i having low occurrence probabilities are considered as the rare tokens. Since our target is to avoid the rare words to be dropped due to low attention weight, the threshold of 50% is taken with an assumption that the rare tokens would fall in this range only otherwise that token is not rare. After inserting the noise in all the source sentences, we make their pairing with their respective target sentences. Finally, this noisy parallel corpus is augmented to the original parallel corpus for final source–to–target training.

6 Experiment Setup

Our translation model is based on the Transformer architecture Vaswani et al. (2017). We use the Sockeye toolkit[9] Hieber et al. (2018) for our experiments. Table 2 gives the size of the training samples for different systems. We also experiment our proposed method on the IIT Bombay English-Hindi parallel corpus Kunchukuttan et al. (2018). To perform experiments for the English–to–French translation, we use a part (for true resource-poor setting) of the UN-corpus Ziemski et al. (2016) for training and newstest2015 Bojar et al. (2015) as the test set. We also perform experiment for English-German translation and test over the IWSLT 2017 testset[10].

The tokens of the training, test and validation sets are segmented into subword units Sennrich et al. (2016) by applying 4,000 BPE merge operations at the source and target sides. Our training set-up details are given below: No. of layers at the encoder and decoder sides: 6 each; 8-head attention; Hidden layer size: 512; Embedding vector size: 512; Learning rate: 0.0002; Minimum batch size: 4800 tokens; early stopping is used to terminate the training.

7 Results and Analysis

From Table 5, we can see significant BLEU score improvement over the baseline using various data and noise augmentation techniques. Using human translated and back-translated corpus, we train the Base+BT model which yields the BLEU improvement of 0.83. Further, with data augmentation techniques, WDA and CDA, we obtain additional 3.48 and 4.47 BLEU score improvement, respectively. The random noise augmentation

[9] `https://github.com/awslabs/sockeye`

[10] `https://wit3.fbk.eu/2017-01`

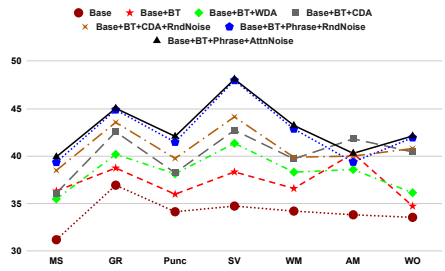

Figure 2: BLEU scores of models in the presence of various kinds of noises in input sentence.

(RndNoise) in CDA model also shows additional improvement of 0.48 BLEU point. In total, with noisy word augmentation methods, we achieve 5.78 BLEU improvement over the base model. After using our proposed phrase replacement (PhrRep) technique, we outperform the word augmentation techniques 'Base+BT+CDA+RndNoise' with 0.47 BLEU score. As mentioned in Table 2, 'PhrRep+RndNoise' model outperforms all the models with comparatively less parallel data. Further adding AttnNoise with 'PhrRep' the model 'Base+BT+PhrRep+AttnNoise' gives 0.42 additional BLEU improvement. In total, with 'Base+BT+PhrRep+AttnNoise' method, we achieve a total of 6.67 BLEU improvement over the 'Base' model. We also perform experiment over the MTNT testset which is a user generated English-French corpus. 'PhrRep' method yields 1.92 BLEU over the baseline score. Further, sing 'AttnNoise' method with 'PhrRep' gives additional 0.63 BLEU improvement.

We also apply our proposed PhrRep technique over the benchmark English-Hindi testset newstest2014 Bojar et al. (2014). As shown in Table 5, we achieve a 1.14 BLEU score improvement over the baseline. We perform statistical significance tests[11] Koehn (2004), and found that the proposed model attains significant performance gain with 95% confidence level (with p=0.013 which is < 0.05). We also apply the PhrRep technique for English–to–French translation. To test the performance in a low-resource scenario, we perform our experiment over a small part of data i.e. 300k parallel sentences. We achieve a gain of 1.55 BLEU (statistically significant) over the baseline. For English–to–German translation task, it also yields significant improvement[12] of 0.91 BLEU over the baseline.

7.1 Analyzing the Robustness

To analyze the models' performance on the product domain testset, we manually tag the test sentences on the basis of major inconsistencies. We divide the testset into the following 7 categories: misspell (MS): 10.09%, wrong Grammar (GR): 6.94%, punctuation mistake (Punc): 7.83%, sub-verb disagreement (SV): 2.56%, word missing (WM): 5.99%, article missing (AM): 1.94% and word order (WO): 3.67%. The distribution in percentage shows how much of the test sentences lie in which noise category. Figure 2 depicts the performance of all the models in presence of different noises. Augmented techniques outperform the 'Base' and 'Base+BT' models in all the major categories. Evaluation results show that 'PhrRep+RndNoise' model outperforms all the other word augmentation models. Further, introducing 'AttnNoise' in 'PhrRep+AttnNoise' improves the performance over 'PhrRep+RndNoise'. It shows that the guided noise augmentation is better than the random noise augmentation based technique. For AM

[11] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/analysis/bootstrap-hypothesis-difference-significance.pl`

[12] $p<0.005$

	Adequacy	Fluency
Base+BT	2.06	2.74
Base+BT+CDA+RndNoise	**2.49** (+0.43)	**3.28** (+0.54)
Base+BT+PhrRep+RndNoise	**2.63** (+0.57)	**3.35** (+0.61)
Base+BT+PhrRep+AttnNoise	**2.87** (+0.81)	**3.52** (+0.78)

Table 6: Average adequacy and fluency score

error, 'PhrRep+AttnNoise' lags behind the 'CDA'.

7.2 Human Evaluation

We also analyze the translation quality from human perception. Each hypothesis is assigned with adequacy and fluency score from 0–to–4 in the following scale:
0- *Incorrect*, **1-** *Almost incorrect*, **2-** *Moderately incorrect*, **3-** *Almost correct*, **4-** *Correct*. We select 500 random test samples and ask 3 language experts to read and assign the fluency and adequacy scores. Table 6 shows the average rating for different data augmentation models assisted with random noise (RndNoise) and Attention guided noise (AttnNoise). We calculate the inter-annotator-agreement scores (IAA) using Fleiss's Kappa. The scores for "Base+BT" model are found to be 0.874 and 0.891 for adequacy and fluency rating, respectively. The proposed model "Base+BT+PhrRep+AttnNoise" shows the scores of 0.867 and 0.913 for adequacy and fluency, respectively. The 'Choice of output tokens', 'translation of noisy source tokens', 'missing source tokens to translate', 'word order', 'tense preservation', 'punctuation', and 'subject-verb agreement' are some important factors while assigning adequacy and fluency scores. PhrRep and AttnNoise techniques provide incremental improvements as shown in Table 6.

8 Conclusion

In this paper, we have presented an effective NMT model for English–to–Hindi product review translation. As there was no parallel corpus in this domain, we, therefore, crawled English reviews, pre-processed, filtered, translated into Hindi and corrected using professional human translators. Hindi descriptions of electronic gadgets are crawled and back-translated into English using human translated corpus and again augmented with human translated corpus. We make the parallel corpus freely available.

We have introduced a novel phrase replacement based augmentation technique (PhrRep) which replaces the whole noun phrase (multiple tokens at a time) with an alternative noun phrase to generate the new training sample in fewer attempts. For robustness in our model, we use a novel attention guided noise augmentation technique (AttnNoise) which drops the words or makes them noisy on the basis of attention weights. Using phraseRep and AttnNoise, for En→Hi review translation, we achieve an improvement of 6.67 BLEU over the baseline. In order to show the generic behavior of our model, we also evaluate it on the English-French and English-German benchmark datasets, demonstrating the effectiveness of our proposed approach.

In future, we shall focus on the spelling variations and code-mixed challenges in the input and output sentences. A bigger English–to–Indic multilingual product review translation system will be investigated.

9 Acknowledgement

Authors gratefully acknowledge the unrestricted research grant received from the Flipkart Internet Private Limited to carry out the research. Authors thank Muthusamy Chelliah for his continuous feedbacks and suggestions to improve the quality of work; and to Anubhav Tripathee for gold standard parallel corpus creation and translation quality evaluation.

References

Anastasopoulos, A., Lui, A., Nguyen, T. Q., and Chiang, D. (2019). Neural machine translation of text from non-native speakers. 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 3070–3080, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Berard, A., Calapodescu, I., Dymetman, M., Roux, C., Meunier, J.-L., and Nikoulina, V. (2019). Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 168–176, Hong Kong. Association for Computational Linguistics.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the ninth workshop on statistical machine translation (WMT 2014)*, pages 12–58.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 workshop on statistical machine translation. pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Edizel, B., Piktus, A., Bojanowski, P., Ferreira, R., Grave, E., and Silvestri, F. (2019). Misspelling oblivious word embeddings. *ArXiv*, abs/1905.09755.

Fadaee, M., Bisazza, A., and Monz, C. (2017). Data augmentation for low-resource neural machine translation. In *Proceedings of ACL*.

Gao, F., Zhu, J., Wu, L., Xia, Y., Qin, T., Cheng, X., Zhou, W., and Liu, T.-Y. (2019). Soft contextual data augmentation for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5539–5544, Florence, Italy. Association for Computational Linguistics.

Hieber, F., Domhan, T., Denkowski, M., Vilar, D., Sokolov, A., Clifton, A., and Post, M. (2018). The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 200–207, Boston, MA.

Karpukhin, V., Levy, O., Eisenstein, J., and Ghazvininejad, M. (2019). Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Kobayashi, S. (2018). Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Kunchukuttan, A., Mehta, P., and Bhattacharyya, P. (2018). The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Michel, P. and Neubig, G. (2018). MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting Similarities among Languages for Machine Translation. *arXiv preprint arXiv:1309.4168*.

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

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (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, Berlin, Germany.

Vaibhav, V., Singh, S., Stewart, C., and Neubig, G. (2019). Improving robustness of machine translation with synthetic noise. 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 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wu, X., Lv, S., Zang, L., Han, J., and Hu, S. (2019). Conditional bert contextual augmentation. In *ICCS*.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. (2016). The united nations parallel corpus v1. 0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia.