Product Review Translation: Parallel Corpus Creation and Robustness towards User-generated Noisy Text

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

Reviews written by the users for a particular product or service play an influencing role for the customers to make an informative decision. Although online e-commerce portals have immensely impacted our lives, available contents predominantly are in English language- often limiting its widespread usage. There is an exponential growth in the number of e-commerce users who are not proficient in English. Hence, there is a necessity to make these services available in non-English languages, especially in a multilingual country like India. This can be achieved by an in-domain robust machine translation (MT) system. However, the reviews written by the users pose unique challenges to MT, such as misspelled words, ungrammatical constructions, presence of colloquial terms, lack of resources such as in-domain parallel corpus etc. We address the above challenges by presenting an English–Hindi review domain parallel corpus. We train an English–to–Hindi neural machine translation (NMT) system to translate the product reviews available on e-commerce websites. By training the Transformer based NMT model over the generated data, we achieve a score of 33.26 BLEU points for English–to–Hindi translation. In order to make our NMT model robust enough to handle the noisy tokens in the reviews, we integrate a character based language model to generate word vectors and map the noisy tokens with their correct forms. Experiments on four language pairs, *viz.* English-Hindi, English-German, English-French, and English-Czech show the BLUE scores of 35.09, 28.91, 34.68 and 14.52 which are the improvements of 1.61, 1.05, 1.63 and 1.94, respectively, over the baseline.

1 Introduction

In the era of exponentially rising internet users, contents over social media, e-commerce portals are increasing rapidly. In recent times, there has been a phenomenal growth in the number of e-commerce users, especially during this COVID pandemic situation. However, the contents in such e-commerce portals are mostly in English, limiting the scope of these services to only a section of the society who can read and/or write in English. India is a multilingual country with 22 officially spoken languages. The number of internet users in India has increased dramatically in the last few years with the widespread usage of low-cost android phones. Users find it very difficult to understand the English contents written in these service portals. Hence, there is a great demand to translate these contents from English to Indian languages. As the manual translation is both time-consuming and cost-sensitive, building an automated machine translation (MT) system that could translate these enormous amounts of reviews will be of great interest. However, there are several challenges for this, such as the non-availability of in-domain parallel training corpus, noisy nature of the text, ungrammatical constructions, and the mixing of more than one language (i.e. code-mixed contents) (ref. Table 1). Product reviews are user generated content where writing inconsistencies are common as shown in Table 1. It is possible to make mistakes in writing words in a sentence due to various reasons, for example, weak grasp on the language, fast writing, writing just to convey the message without concerning more about the sentence formation etc.

Type of noise	Example
Emoji	Face unlock works well. even in dim light
Char Repetiton	full package bestttttt phone
Capital letter	NICE PHONE IN LOW BUDGET.
Misspell	Awsome prodct....loved it
Punctuation Irregularity	phone gives best photos!! awesome feeling
Article missing	It is best earphone I got with phone
Starting noun pronoun missing	was a nice product i got
Code Mixed	Good product. lekin price bahot high hai

Table 1: Various noise present in product review sentences

In our current work, we take up English–to–Hindi translation as there are 57.09% Hindi

Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), pages 174–183
August 5, 2021. ©2021 Association for Computational Linguistics

1. Source	the perfomence of the phone is bad.
Reference	फोन की परफॉर्मेंस खराब है। (phone of performance bad is.)
Output	फोन का परफ्यूम खराब है। *(phone of perfume bad is.)*
2. Source	The content is a disgrce to the page.
Reference	Der Inhalt ist eine Schande für die Seite.
Output	Der Inhalt ist eine Abneigung gegen die Seite.
3. Source	current procedure is more transpatent
Reference	la procédure courante est plus transparente .
Output	la procédure courante est plus transcriptive .

Table 2: Sample outputs for **1** En→Hi, **2.** En→De and **3.** En→Fr translation in presence of noisy input tokens.

speakers in India[1]. These two languages are morphologically and syntactically distant to each other, posing challenges to build a robust NMT system. We crawl the English review sentences (electronic gadgets) from the e-commerce websites. After pre-processing (ref. Section 3.2) and filtering (ref. Section 3.3), we translate the English sentences into Hindi language using our in-house English-Hindi translation system[2]. The generated Hindi output sentences are given to the professionals who are experts in Hindi and English languages. The language experts post-edit the Hindi output as per the guidelines (ref. Section 3.5) provided to them. In addition, we also crawl monolingual Hindi sentences (ref. Section 3.6) from electronics gadgets' description websites. These sentences are back-translated[3] (Sennrich et al., 2016a) using the Hindi-to-English translation model trained over the post-edited parallel corpus.

Neural machine translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017) is the dominant translation technology nowadays, and adapting this to the noisy text is very crucial due to the phenomenal growth in social media. Since NMT models learn from a fixed number of source and target vocabulary during training, any noisy word during the inference becomes an out-of-vocabulary (OOV) token because it does not belong to the NMT model's training vocabulary. It is not possible to train an NMT model with all the noisy versions of a correct token. In this case, models treat noisy tokens as OOV tokens and either miss their translation in the output sentence or translate them incorrectly. Incorrect translation of noisy tokens affects the translation quality of the whole output sequence. It affects the translation output and degrades the output quality. For example, English-to-Hindi (En–to–Hi) NMT model has one token *performance* as a part of its source vocabulary during training. As shown in example 1 in Table 2, a noisy version *performence* appears in the input sentence which is incorrectly translated as 'परफ्यूम' *perfume* instead of 'परफॉर्मेंस' *performance*. Similarly, in examples 2 and 3, we can see that *disgrce* and *transpatent* are the noisy tokens in the English to German (En–to–De) and English to French (En–to–Fr) models, respectively, and both of these noisy tokens are incorrectly translated by their respective translation models.

To handle the noisy tokens as source input, we integrate a similarity based token replacement model before word segmentation at inference time where the word vectors of noisy input tokens are matched with the correct and seen tokens in the source vocabulary, and replaced with the highest similar token. We use a character based language model to generate the word vectors and map the noisy and correct version of tokens in vector space. The generation of the word vectors depends on the characters present in that word.

The remainder of the paper is organized as follows. In Section 2, we discuss the related work. Section 5 presents the approaches of training the character language model, word vector generation model, and handling of noisy source input tokens at inference time. Section 6 presents the details regarding the dataset used and the experimental setup. Results and analysis of our approach are discussed in Section 7. Finally, Section 8 concludes the work with future research directions.

2 Related Work

Machine translation with noisy text is, itself, a very challenging task. Noisy tokens (misspelled words) pose great challenges to develop the Neural Machine Translation (NMT) models (c.f. Table 2) (Michel and Neubig, 2018). In the literature, there are a few existing works that focus on handling the noisy text by increasing the robustness of the translation model. An MTNT (machine translation of noisy text) test-bed was introduced in (Michel and Neubig, 2018) that discussed the challenges of noisy contents. It has been also observed that even small noise in the input sentence can degrade the translation quality of the NMT model significantly (Belinkov and Bisk, 2018; Karpukhin et al., 2019). To improve the robustness of the translation model, they introduced synthetic errors like character swapping, replacement and drop in the corpus. Synthetic noise using back-translated corpus was also inserted in the original corpus to introduce the NMT model with noise at train-

[1]https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India

[2]This system has BLEU of 55.67 for judicial domain

[3]Translating monolingual target data using target→source NMT model.

Sr.	English Sentence (crawled)	Hindi Sentence (corrected)
1.	rounded corners make griping the phone very well .	राउंडेड कोर्नर फोन को बहुत ही अच्छी पकड़ देते हैं । (raunded kornar phon ko bahut hee achchhee pakad dete hain)
2.	one of the best phone ever at this price .	इस कीमत में अब तक के सबसे अच्छे फोन में से एक । (is keemat mein ab tak ke sabase achchhe phon mein se ek .)
3.	but this is Apple and Apple is like that only	लेकिन यह ऐप्पल है और ऐप्पल ऐसा ही होता है (lekin yah aippal hai aur aippal aisa hee hota hai)
4.	At first I want to say Thank u flipkart.	सबसे पहले मैं थैंक यू फ्लिपकार्ट कहना चाहता हूं। (sabase pahale main thaink yoo phlipakaart kahana chaahata hoon.)
5.	Rear camera image quality is very good.	रियर कैमरा इमेज क्वालिटी बहुत अच्छी है। (riyar kaimara imej kvaalitee bahut achchhee hai)

Table 3: Samples from the generated English-Hindi parallel corpus

ing time (Vaibhav et al., 2019; Anastasopoulos et al., 2019).

Since it is difficult for the NMT model to see all the noisy variants of a correct token at training time, the model hence treats the noisy tokens as the unseen tokens. Word segmentation is a popular method that deals with the unseen tokens. Byte-pair-encoding (BPE) (Sennrich et al., 2016b) segments the words based on the rare character combinations. In BPE, a word is converted into the subword units based on the fixed learned list of less frequent character combinations. Subword regularization (SR) (Kudo, 2018) was introduced as a more diverse word segmentation method which segments the words based on a unigram language model. For these segmentation models, it is difficult to capture all the noisy versions at training time. So before segmentation, we use a character based language model that maps the noisy and correct versions of tokens together in a vector space as shown in Figure 1. It helps to replace the noisy token with its correct form before inference.

There has not been any significant attempt to translate the product reviews, *except* the one proposed in (Berard et al., 2019) that addressed the translation of English to French. In contrast, we develop product review translation system for English-Hindi. English and Hindi are morphologically and syntactically distant languages, which pose more challenges for machine translation. Further, Hindi is a resource-poor language for which we do not have sufficient resources and tools, even for the generic domain.

3 Parallel Corpus Creation

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.

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 2 times is trimmed and then checked for its compatible correct word using spell-checker[4], and a list provided by Facebook[5] (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[6] (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 corrected.

3.3 Filtering Standard vs. Non-standard Sentences

We prepare the translation model to deal with the noises as mentioned in Section 3.1. Some sentences in reviews are written in Roman script[7]. We consider these sentences as non-standard sentences. Before generating the target counterpart of the source sentences, we filter out the non-standard sentences using an autoencoder based NMT model. We use Sockeye toolkit (Hieber et al., 2018) to train our model, and the hyperparameters used are mentioned in Section 6.2. Steps involved in the filtering process are as follows:

- Apply 30,000 BPE merge operations using subword technique(Sennrich et al., 2016b) over 21.2 million English monolingual data (Bojar et al., 2014).

[4] https://pypi.org/project/pyspellchecker/

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

[6] https://en.wikipedia.org/wiki/List_of_mobile_phone_brands_by_country

[7] We do not focus on the sentences written in the Roman script (Hindi words written in Roman script (English letters)).

System	Parallel samples	BLEU	TER
Base	13,000	33.26	46.49
Base+BT	48,000	37.79	41.35

Table 4: BLEU and TER scores for English–to–Hindi NMT system over review domain corpus

- Train an English–to–English system. Here, the source and target are identical.
- After training, infer the English sentence from the crawled product reviews and generate an English hypothesis.
- Calculate the similarity between the input sentence and the inferred hypothesis using BLEU score.
- If BLEU $<$ 40 then the sentence will be filtered out. We consider 40 BLEU point as a threshold because BLEU in the range 30-40 is considered as "understandable to good translations"[8].

The objective of training the autoencoder is to generate an output sequence very similar to the input sequence. Model would not be able to regenerate a source input properly if it is not trained on the similar kind of samples.

On an average from the total crawled reviews, 15 to 20 % reviews were filtered out as the non-standard sentences which were dropped and not considered further. After this filtering, there were still some sentences left, having grammar and spelling inconsistencies. These sentences have been considered as noisy sentences. Noise handling techniques as discussed in Section 5 are used to train the model to translate the noisy sentences.

3.4 Gold Corpus Creation by Human Post-editing

After pre-processing and filtering, we obtain 16,138 standard English sentences. Instead of translating sentences from scratch, we use an in-house English-Hindi machine translation system developed for the judicial domain. The model is trained for English-Hindi translation using judicial data (English-Hindi), and additional English-Hindi corpus (Kunchukuttan et al., 2018)[9]. The sentences generated from this automatic translation are post-edited by human experts. The experts are post-graduates in linguistics and have good command in Hindi and English both. The experts read the English sentences and its Hindi translation. They were instructed to make the correction in the sentences, if required. Some guidelines for making the corrections in the data are mentioned in Section 3.5. The human corrected parallel corpus is divided into training, development and test set consisting of 13000, 599 and 2,539 parallel sentences, respectively. Vocabulary size of English and Hindi training data is 9,331 and 8,367 tokens respectively. We also crawl Hindi sentences and back-translate them into English. In Table 4, 'Base+BT' shows the size of those samples. Section 3.6 describes the generation process of that synthetic (back-translated) data.

3.5 Guidelines for the Gold Corpus Creation

Guidelines for making the corrections (ref. Section 3.4) to generate the review domain parallel corpus are as follows:

- Source and target sentence should carry the same semantic structure.
- Product name should be transliterated.
- User friendly vocabulary selection at Hindi (target) side. Too many complicated Hindi words which are not in much use should be avoided. Transliteration of an English word can also be used in the Hindi side because in India, people generally use Hinglish (mix of Hindi and English words) vocabulary, e.g. 'time', 'face recognition', 'premium' etc.
- If hyphen, slash, dot etc. symbols occur in the source side then the same pattern should be preserved at the translated side too.
- Literal translation can be avoided sometimes. For example, adjectives and nouns like terrible, great etc. which carry extreme intensity should be translated into understandable simple words as घटिया (ghatiya), शानदार (shaanadaar) respectively which preserve the sense and intensity.

A few samples from the generated parallel English-Hindi corpus are shown in Table 3.

3.6 Crawling Hindi Reviews and Back-translating into English

We crawl 35,000 monolingual Hindi sentences from the various websites[10][11][12] which provide Hindi descriptions of electronic gadgets. Since these are commercial websites, we randomly gave 3,000 sentences out of all the

[8] https://cloud.google.com/translate/automl/docs/evaluate

[9] It achieves a 55.67 BLEU (En-to-Hi) on our in-house judicial domain test set

[10] https://www.digit.in/hi/reviews/

[11] https://hindi.gadgets360.com/reviews

[12] https://www.91mobiles.com/hi/tech/

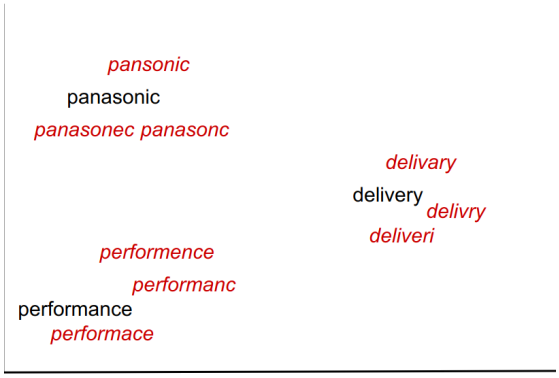

Figure 1: Mapping the noisy and correct forms of tokens close to each other in a vector space

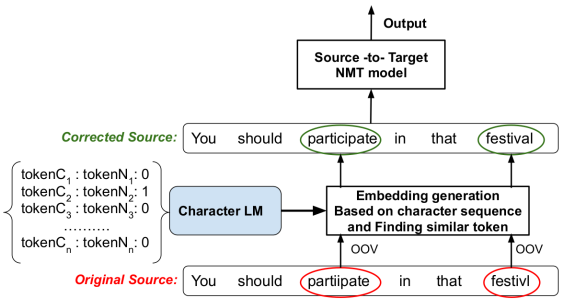

Figure 2: Token correction at inference time using character sequence based word embedding. *tokenC* and *tokenN* are the correct and noisy tokens respectively paired together for training. *0* and *1* denotes the similar and non-similar token pairs respectively.

crawled Hindi sentences as a sample to our language experts to read, and they found them to be in-domain, relevant, correct in the sense of syntax and semantics, and hence useful for our use-case. We build a Hindi–to–English NMT model to back-translate the crawled Hindi sentences. We use 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 corrected review domain parallel corpus. The fine-tuned Hindi–to–English NMT model is used to back-translate the crawled 35,000 monolingual Hindi sentences into English. This back-translated (BT) English-Hindi synthetic parallel corpus is augmented with the original 13,000 parallel sentences. Table 4 gives the statistics about the dataset. A new system Base+BT model from English–to–Hindi is trained using the human corrected+back-translated corpus. We will make the human corrected and back-translated parallel corpus available on request for the research purpose [13].

4 Training NMT over Human Corrected Corpus

We train an English–to–Hindi baseline model using the human corrected corpus as mentioned in Table 4. We use the Sockeye framework (Hieber et al., 2018) for training the Transformer neural network based NMT. We splitted the words into subwords (Sennrich et al., 2016b) using BPE technique. We perform 4,000 BPE merge operations. Our model contains 6-6 encoder-decoder layers, 512 hidden size and word embedding size, learning rate as 0.0002 and min batch size as 3800 tokens. We used early stopping over the validation set.

After training over the human corrected corpus, we perform testing over the review domain test set and achieves 33.26 BLEU points and name it as 'Base'- the baseline model. In addition to it, we also add the back-translated synthetic corpus into human corrected corpus, and train the NMT model over it. We call it as the 'Base+BT' model that yields 37.79 BLEU points.

5 Handling Noisy Tokens

In this section, we describe the methodology used in our work. Figure 2 presents the overall process of our proposed method. It consists of various steps like character language model (LM) training, word vector (embedding) generation, and finally noisy token replacement at inference time. Section 5.1 and Section 5.2 describe the steps in details.

5.1 Training Character LM and Word Vector Generation

A word consists of a sequence of characters. Each character is represented as a one-hot vector and a sequence of such vectors is passed through two different Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) layers. It generates the embedding vector of that particular word. As a training model, chars2vec[14] is utilized for embedding generation and character sequence learning. To be more specific, we deal with a neural network taking two sequences of one-hot vectors representing two different words as an input, creating their embeddings with one chars2vec model, calculating the norm of the difference between these embedding vectors and feeding it into the last layer of the network with the sigmoid activation function. The output of the neural network is a number that ranges

[13] https://www.iitp.ac.in/~ai-nlp-ml/resources/data/review-corpus.zip

[14] https://github.com/IntuitionEngineeringTeam/chars2vec

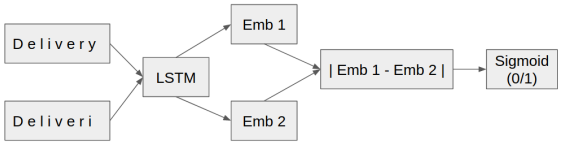

Figure 3: Training the character based language model

from 0 to 1 because of the sigmoid as an output function. The network is trained to capture the similarity between the noisy and its non-noisy version. For similar word pairs, i.e. for noisy and its equivalent version, we use *0* as a class label. On the other hand, we use *1* to denote the non-similar word pairs. For example, *panasonic* and *pansonic* are similar pairs while *panasonic* and *panorama* are non-similar. As shown in Figure 3, we are trying to reduce the distance of two embeddings *Emb1* and *Emb2* of two similar tokens so that they can be mapped as close as possible in vector space. This is why the label of similar pairs is *0* and the training objective is to reduce the distance between *Emb1* and *Emb2* close to *0*.

As shown in the Figure 1, our objective is to map the correct and noisy versions of a token as close as possible in vector space. To train the character LM, we prepare the training data by creating noisy versions of tokens in the source vocabulary set *trainX*. The labelled training data can be generated by taking a multitude of words and then performing various changes (e.g. character drop and replacement) upon them to obtain the noisy versions of those words. These new noisy words, so produced by injecting character errors in one original word, would naturally be similar to this original word, and such pairs of words would have the label 0. As an example, two noisy versions *performnce* and *performence* are generated using character drop and character replacement, respectively, from the original source vocabulary word *performance*. So [(performance, performnce) : 0] and [(performance, performence) : 0] are two similar training pairs with label *0*. It is to be noted that on a source token we apply at most two character operations to generate the similar pairs. To generate non-similar pairs with label *1*, with token from the source vocabulary, we randomly pair the shuffled tokens, for example: [(performance, product) : 1] and [(performance, smartphone) : 1]. These training pairs are used to train and save the character LM model which learns the parameter in the process of mapping the similar word embeddings closer. Now this model is used to generate the vector representation of the source vocabulary tokens and tokens in the input sentence at inference time.

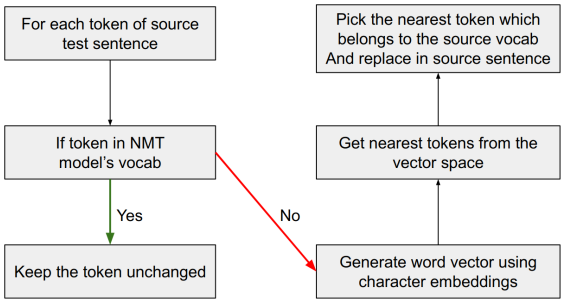

Figure 4: Flowchart of the noisy token replacement

5.2 Noisy Token Replacement

As discussed in Section 5.1, a trained model is saved which is used to generate the vector representation (embedding) of each word in the training source vocabulary. The vector representation is generated based on the characters in those words. Let us have a vector space S which contains the vector representation $trainV_i$ of training source vocabulary token $trainX_i$. Here, $trainV_i$ is generated using the trained chars2vec model based on the characters appearing in the token $trainX_i$.

Now at the time of inference, each test input sentence $input_i$ consists of len tokens and $j = 1, ..., len$. Here, we assume that if a noisy token or say a noisy version of a word appears in the test input sentence then it will not be a part of the training source vocabulary $trainX$. As shown in Figure 4, for each token $inferX_{ij}$ in the test input sentence $input_i$, we find if $inferX_{ij}$ belongs to the source train vocabulary $trainX$ then we keep that token as it is in source input sentence. If $inferX_{ij}$ does not belong to the source train vocabulary $trainX$, we find the most similar token from the vocabulary list $trainX$ using the cosine similarity. Now $inferX_{ij}$ will be replaced with the most similar token from $trainX$. Finally, the corrected (replaced) source sequence segmented by the subword model will be fed to the NMT model for the translation.

6 Dataset and Experimental Setup

In this section, we present the details of the datasets used in our experiments and the various setups.

6.1 Dataset

We perform experiments with four different translation directions which are English–to–Hindi (En–to–Hi), English-German (En–to–De), English–to–Czech (En–to–Cs) and English–to–French (En–to–Fr). Among these language pairs, English-Hindi is a low-resource and less-explored, and distant language pair.

	Train	Dev	Test
En-Hi (Reviews)	13,000	599	2,539
En-Hi (WMT14)	1,561,840	520	2,507
En-De (WMT14)	1,264,825	1,057	2,000
En-Cs (IWSLT17)	105,924	483	1,080
En-Fr (IWSLT17)	230,912	883	1,466
En-Fr (MTNT18)	36,014	852	1,020

Table 5: Size of train, dev and test sets for different language pairs

	Proposed	**Synthetic Noise** (Vaibhav et al., 2019)	**SR** (Kudo, 2018)	**BPE** (Sennrich et al., 2016b)
En–to–Hi (Reviews)	35.09	34.27	33.48	33.26
En–to–Hi (newstest2014)	14.64	14.08	13.68	13.35
En–to–De (newstest2014)	28.91	28.22	27.86	27.84
En–to–Cs (IWSLT17)	14.52	13.65	12.58	12.04
En–to–Fr (MTNT18)	23.01	22.87	21.46	20.83
En–to–Fr (IWSLT17)	34.68	33.62	33.05	33.11

Table 6: Evaluation results of the proposed method in terms of BLEU score for different translation pairs. Here, **SR:** Subword regularization, **BPE:** Byte pair encoding

For English–to–Hindi translation, we use the IIT Bombay English-Hindi parallel corpus[15]. For English–to–Hindi, we also perform experiments over the generated review domain parallel corpus. For English–to–German, we use Europarl corpus from WMT 2014[16] (Bojar et al., 2014). We use the IWSLT17 dataset for English–to–Czech and English–to–French[17]. We also use the MTNT[18] dataset for English–to–French translation. Table 5 presents the statistics of training, development and test sets.

6.2 Experimental Setup

In order to build our machine translation systems, we use the Sockeye[19](Hieber et al., 2018) toolkit. Our training set-up is described below. The tokens of training, evaluation and validation sets are segmented into the subword units using the BPE technique (Gage, 1994) proposed by (Sennrich et al., 2016b). We perform 20,000 join operations. We use 6 layers at encoder and decoder sides each, 8-head attention, hidden layer of size 512, embedding vector of size 512, learning rate of 0.0002, and the minimum batch size of 3800 tokens.

6.3 Noise Injection in the Test Sets

For the experiment, we introduce noise in the En-Hi, En-De, En-Fr and En-Cs test sets to make them noisy and suitable for testing the models' performance in the noisy environment. We introduce two kinds of noise in the source test sequence: **1.** character drop and **2.** character replacement. In character drop, we randomly drop any character from a source token and in character replacement, we replace the characters randomly with some other characters.

7 Result and Analysis

We evaluate the models using BLEU, and these results are shown in Table 6. We also perform experiments using the word segmentation approaches, viz. BPE (Sennrich et al., 2016b) and subword regularization (Kudo, 2018). For English–to–Hindi review domain translation, proposed method yields 35.09 BLEU points which significantly outperforms synthetic noise, SR and BPE with a difference of 0.82, 1.61 and 1.83 BLEU points, respectively. We also perform experiments for En–to–Hi translation using benchmark newstest2014 as the test set. We achieve 0.96 and 1.29 BLEU improvements over subword regularization (SR) (Kudo, 2018) and byte-pair-encoding BPE (Sennrich et al., 2016b), respectively. We also evaluate the performance for En–to–De translation and achieve 1.05 and 1.07 BLEU improvements over SR and BPE, respectively. For En–to–Cs, we use the IWSLT17 testset, and the evaluation yields 1.94 and 2.48 BLEU improvements over SR and BPE, respectively. For En–to–Fr, we evaluate over two datasets, IWSLT17 and MTNT18. The MTNT is a noisy testset for En-Fr translation. For the MTNT testset, our model yields 1.55 and 2.18 BLEU improvements over SR and BPE, respectively. For IWSLT testset, En–to–Fr translation using our approach achieves the 1.63 and 1.57 BLEU improvements over SR and BPE, respectively.

We also perform experiments by adding synthetic noise in the training corpus (Vaibhav et al., 2019) which is a noise handling technique. For En–to–Hi, En–to–De, En–to–Fr and En–to–Cs, our proposed method achieves 0.96, 1.05, 1.63 and 1.94 BLEU improvement, respectively, over the synthetic noise model (Vaibhav et al., 2019). We perform statistical significance tests[20] (Koehn, 2004), and found that the proposed model attains significant performance gain with 95% confidence level (with p=0.013 which is < 0.05). For En–to–Fr over MTNT18 testset, we achieve only 0.14 BLEU improvement over the synthetic noise model (Vaibhav et al., 2019) which is not a

[15]http://www.cfilt.iitb.ac.in/iitb_parallel/

[16]http://www.statmt.org/wmt14/translation-task.html

[17]https://wit3.fbk.eu/

[18]https://www.cs.cmu.edu/~pmichel1/mtnt/

[19]https://github.com/awslabs/sockeye

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

En–to–Hi (newstest2014)	20%	25%	30%	40%
Subword Regularization (SR)	14.37	13.68	10.31	9.81
Proposed	14.84	14.54	12.86	11.27
En–to–De (newstest2014)				
Subword Regularization (SR)	27.24	26.18	24.83	23.48
Proposed	28.53	27.34	26.08	25.22
En–to–Fr (IWSLT17)				–
Subword Regularization (SR)	33.14	32.26	29.24	28.37
Proposed	34.45	33.29	31.07	30.35

Table 7: Performance evaluation in terms of BLEU scores by increasing the % of noisy tokens

significant improvement.

7.1 Quantitative Analysis

We evaluate the performance of our approach in the presence of varying amount of noisy tokens. We inject the noise (character drop and character replacement) in 20%, 25%, 30% and 40% tokens in source input sentences for En-Hi, En-De and En-Fr testsets. Table 7 shows the change in the BLEU scores by increasing the count of noisy tokens. As we increase the number of noisy source tokens for three translation tasks, *viz.* En–to–Hi, En–to–De and En–to–Fr, we observe a decrease in BLEU score in both the models (proposed and SR). But our proposed method preserves the robustness significantly as compared to the SR model.

7.2 Human Evaluation

We perform the qualitative analysis of outputs using human evaluation. We took 250 random samples from English-Hindi review test set. It is given to 3 language experts (postgraduate in linguistics and have experiences for the translation task) to rate the outputs on the basis of adequacy and fluency and assign the scores in the range of 0 to 4 *(0: incorrect, 1: almost incorrect, 2: moderately correct, 3: almost correct and 4: correct)*. Table 8 shows the average ratings for the En-Hi translation.

We also calculate the inter-annotator-agreement scores (IAA) using Fleiss's Kappa. The scores for "En–to–Hi (proposed)" translation are found to be 87.2 and 86.8 for adequacy and fluency rating, respectively. The "En–to–Hi(SR)" translation shows the scores of 89.5 and 84.0 for adequacy and fluency, respectively. We also present a few output samples and error analysis in the appendix **A**.

8 Conclusion

In this paper, we have developed a robust NMT model for product review translation that can handle noisy input text. Because of the absence of an in-domain parallel corpus, we introduce a parallel English-Hindi corpus for product review domain. We crawl the user reviews of electronic gadgets from e-commerce sites written into English language. These are

	Adequacy Range: 0-4	Fluency Range: 0-4
En–to–Hi (Proposed)	2.65	2.81
En–to–Hi (SR)	2.47	2.68
En–to–Hi (BPE)	2.37	2.61

Table 8: Human evaluation for English–to–Hindi translation

pre-processed; passed through an in-house judicial domain NMT system; and a part of this dataset is post-edited by the language experts. It is also observed that product reviews which are user generated content contain noisy tokens which are a challenge to handle in any MT system. Due to the limitation of fixed vocabulary size at training time, it is not possible for the NMT models to see all the noisy variants of input tokens. We have integrated a token replacement approach during the inference time. We trained a character based language model which generates the vector representation of the tokens present in the source vocabulary based on the characters present in that word. The token replacement approach finds the most similar token from the source vocabulary for each noisy input token at inference time to replace it with the correct token.

We perform experiments over a variety of language pairs, such as En–to–Hi, En–to–De, En–to–Fr and En–to–Cs. and using the proposed approach, we achieve 35.09, 28.91, 34.68 and 14.52 BLEU points respectively. We also observe the behaviour of the proposed method by varying the % (20, 25, 30 and 40%) of noisy tokens at the input side. The proposed method significantly outperforms the baseline in the presence of different quantities of noisy tokens. Human evaluation shows that our model achieves good fluency and adequacy levels.

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

Antonios Anastasopoulos, Alison Lui, Toan Q. Nguyen, and David Chiang. 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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representation (ICLR 2015)*.

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

Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. 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.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, 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.

Bora Edizel, Aleksandra Piktus, Piotr Bojanowski, Rui Ferreira, Edouard Grave, and Fabrizio Silvestri. 2019. Misspelling oblivious word embeddings. *ArXiv*, abs/1905.09755.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 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.

Philipp Koehn. 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.

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.

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

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*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016) (Volume 1: Long Papers)*, pages 86–96.

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)*, pages 1715–1725, Berlin, Germany.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A Comparing Output Samples and Error Analysis

Table 9 shows some examples to illustrate the performance of the proposed model. In example A, we translate an English sentence

A. En→Hi Source	Names of other partiipatng lawmkers were to be released in coming days .
Reference	सहभागिता करने वाले अन्य विधि - निर्माताओं के नाम आने वाले दिनों में जारी किए जाएँगे । (sahabhaagita karane vaale any vidhi - nirmaataon ke naam aane vaale dinon mein jaaree kie jaenge)
Output (Subword Regularization)	आने वाले दिनों में अन्य दर्लो के नाम जारी किए जाने थे। (aane vaale dinon mein any dalon ke naam jaaree kie jaane the)
Corrected Source	names of other participating lawmakers were to be released in coming days .
Output (Proposed)	आगामी दिनों में अन्य भाग लेने वाले विधिनिर्माता के नाम जारी किए जाने थे। (aagaamee dinon mein any bhaag lene vaale vidhinirmaata ke naam jaaree kie jaane the)
B. En→De Source	the European Commission's sixth report prsents very valuable conclusions .
Reference	der Sechste Bericht der Europäischen Kommission bietet sehr wertvolle Schlußfolgerungen .
Output (Subword Regularization)	der Sechste Bericht der Europäischen Kommission ist sehr wertvoll .
Corrected Source	the European Commission ' s sixth report presents very valuable conclusions .
Output (Proposed)	der Sechste Bericht der Europäischen Kommission enthält sehr wertvolle Schlußfolgerungen .
C. En→Hi Source	The new featurs of the phone lok nice.
Reference	फोन के नए फीचर अच्छे लगते हैं। (phon ke nae pheechar achchhe lagate hain.)
Output (Subword Regularization)	फोन के नए सौदे बहुत ही अच्छे थे। (phon ke nae saude bahut hee achchhe the)
Corrected Source	The new features of the phone lock nice.
Output (Proposed)	फोन के नए फीचर अच्छे से लॉक होते हैं। (phon ke nae pheechar achchhe se lok hote hain.)

Table 9: Output samples for English→Hindi and English→German translation

into Hindi. The two tokens, *partiipating* and *lawmkers* are noisy and appear as OOV candidates for the trained NMT model. The SR model is not able to recognize those tokens and misses their translations in the output sentence. Our proposed method of replacing the tokens using character LM finds the two most similar tokens *participating* and *lawmakers* as the correct tokens and update the source English sentence which results in the correct Hindi sentence as the output. Similarly, in example B, for English–to–German translation, *prsents* appears as noisy as well as OOV token, which is eventually replaced by its correct version *presents* in the proposed method.

Example C shows one limitation of the spell correction method. There are two misspelled tokens *featurs* and *lok* in the source sentence. Using the proposed method, *features* and *lock* tokens appear as the replaced correct tokens respectively. *features* is the correct replacement for *featurs* but *lock* is not the correct replacement for *lok*. It should be *look* as the correct token. But, *lok*, *lock* and *look* tokens contain almost similar character combinations which make them appear closer to each other in the vector space. So our method may struggle in case of very small length (character count) noisy tokens.