Identifying Complaints from Product Reviews in Low-resource Scenarios via Neural Machine Translation

Raghvendra P. Singh[‡], **Rejwanul Haque**[⋆], **Mohammed Hasanuzzaman**[†] **and Andy Way**
The ADAPT Centre
[‡]School of Computing, Dublin City University, Dublin, Ireland
[⋆]School of Computing, National College of Ireland, Dublin, Ireland
[†]School of Computing, Cork institute of Technology, Cork, Ireland
raghvendra.singh6@mail.dcu.ie
rejwanul.haque,mohammed.hasanuzzaman,andy.way@adaptcentre.ie

Abstract

Automatic recognition of customer complaints on products or services that they purchase can be crucial for the organisations, multinationals and online retailers since they can exploit this information to fulfil their customers' expectations including managing and resolving the complaints. Recently, researchers have applied supervised learning strategies to automatically identify users' complaints expressed in English on Twitter. The downside of these approaches is that they require labeled training data for learning, which is expensive to create. This poses a barrier for them being applied to low-resource languages and domains for which task-specific data is not available. Machine translation (MT) can be used as an alternative to the tools that require such task-specific data. In this work, we use state-of-the-art neural MT (NMT) models for translating Hindi reviews into English and investigate performance of the downstream classification task (complaints identification) on their English translations.

1 Introduction

Almost all online retailers allow users to freely express their opinions and thoughts on products via their websites and relevant social media platforms. Customers who intend to purchase a product may take purchasing decisions based on the reviews of the product. Accordingly, commercial and retail companies consider product reviews as an important source of information, and could exploit this information to build their marketing tools and strategy, and to resolve any issues in relation to the product. This could also benefit users with suggestions on the quality of the products or services that they want to purchase. As for the number of reviews of a product posted by the users, they could range from several hundreds to tens of thousands. E-commerce companies and online retailers want to identify complaints given the reviews of a product for their own benefit. Likewise, customers who want to buy a product or service may need such information while avoiding having to consult thousands of reviews about the product.

In this context, Gupta et al. (2014) identified the relationship between users' purchase intent from their social media forums such as Quora[1] and Yahoo! Answers,[2] and Wang et al. (2015) investigated the problem of identifying purchase intent with using a list of seed intent-indicators (e.g. 'want to'). Haque et al. (2019b) extend the work of Wang et al. (2015) while increasing the coverage of the purchase intent indicators with the distributed vector representation of words using the continuous skip-gram model (Mikolov et al., 2013).

Recently, Preotiuc-Pietro et al. (2019) automatically identified complaints from tweets posted by social media users and potential customers. In Singh et al. (2020), we conducted a similar study in an attempt to identify complaints from opinionated texts (reviews) about products posted in a low-resource language, Hindi, from the the websites of the retail giant Amazon India[3] and the popular social media platform YouTube.[4] For investigating this problem in Hindi (Singh et al., 2020), as in Gupta et al. (2014); Wang et al. (2015); Haque et al. (2019b); Preotiuc-Pietro et al. (2019), we had to manually create labeled training data[5]

[1] www.quora.com

[2] www.answers.yahoo.com

[3] https://www.amazon.in/

[4] https://www.YouTube.com/

[5] https://github.com/MrRaghav/

by employing a number of human annotators. The process of creating such a data set is not easy; it is not only time-consuming and laborious but also a very expensive task.

In this context, Tebbifakhr et al. (2019) investigated possibility of exploiting MT in a specific NLP task in a language for which dedicated tools are not available due to the scarcity of task-specific training data. As in Tebbifakhr et al. (2019), in this work, we considered Hindi, an under-resourced Indic language, and investigate whether MT can play a role in complaint identification and eliminate the requirement for complaint identification tools for Hindi, which require labeled data for training, which is expensive to create. Accordingly, we study the following two scenarios while considering reviews about a variety of products from the the websites of Amazon India and YouTube expressed in Hindi as the test examples in our experiments, namely performance of the classifiers (complaints identifiers) built for English on the English translations of the Hindi reviews by (i) our MT systems and (ii) human translators. Note that as a part of our investigation, we created a labeled training dataset of English reviews about products posted in Amazon, and detail the data creation process and statistics in Section 2.2.

Unlike Tebbifakhr et al. (2019) who focus on improving a downstream task (i.e. sentiment classification) by controlling translations of an MT system but at the expense of translation quality, we customise our neural MT systems using the standard and commonly-used data augmentation and terminology-aware domain adaptation techniques (Jooste et al., 2020; Haque et al., 2020c; Nayak et al., 2020b; Parthasarathy et al., 2020) so that the translations produced by the MT systems can retain source-side stylistics property and semantics as much as possible. In other words, in this study, we aim to observe the performance of the English classifiers (complaint identifiers) on the translations of the Hindi reviews by the baseline, adapted/customised neural MT systems, and human translators.

The remainder of the paper is organised as follows. In Section 2, we detail how we created training data for our experiments. Section 3 describes our MT system building and setups. In Section 4, we present our experimental methodology for complaint classification. Section 5 presents our evaluation results, with some discussion. Section 6 concludes and provides avenues for further work.

2 Dataset Creation

This section details the creation of training data that has been used in this task.

2.1 The Hindi Review test data

In attempt to create an evaluation test data of reviews, we first collected reviews written in Hindi posted online. The reviews were taken from two different sources: (i) websites of Amazon India, and (ii) YouTube. Amazon India has around 180 million listed products and YouTube has 265+ million active users. In order to collect the reviews from the Amazon India websites, we used the *amazon-reviews-scraper Python library*[6] which takes a product name as input and provides reviews about the product across the different languages. Similarly, in order to collect the reviews from YouTube, we used the *YouTube-comment-downloader Python library.*[7] This script provided us with reviews on the products across the different languages. In order to remove noise (e.g. HTML tags, special characters) from reviews, we applied a number cleaning scripts including a language identifier.[8]

Each of the collected clean reviews is manually tagged with a particular category, namely complaint or non-complaint. For this, we followed the annotation scheme described in Singh et al. (2020). A sample of annotated test set is presented in Table 1. The statistics about the test set reviews are shown in Table 2. We can see from Table 2 that the test set contains 400 examples, with 200 complaints and 200 non-complaints reviews. The numbers of positive and negative examples are equal because we wanted to use a balanced test set in our experiments. Note that the Hindi re-

Complaints-mining-from-Hindi-product-reviews

[6] https://github.com/philipperemy/amazon-reviews-scraper. Accessed on August 2020

[7] https://github.com/egbertbouman/YouTube-comment-downloader. Accessed on August 2020.

[8] https://pypi.org/project/pycld2/

	Review	**Label**
Hi:	वो ज़िन्दगी जो हम जीना चाहते हैं	0
En:	The life we want to live	
Hi:	पर फेस अनलॉक चल नहीं रहा	1
En:	But face unlock is not working	
Hi:	हिन्दी माध्यम के लिए एक वरदान	0
En:	A boon for Hindi medium	
Hi:	पृष्ठों की क्वालिटी व छपाई बहुत ही खराब हैं	1
En:	The quality and printing of pages are very poor	
Hi:	समान की डिलवरी ही नहीं हुई	1
En:	The product was not delivered	

Table 1: Sample Hindi reviews from test set and their manual English translations.

	count	words (HI)	words (EN)
Reviews	400	5,141	4,762
Complaints	200	2,932	2,738
Non-Complaints	200	2,209	2,024

Table 2: Statistics of the test set reviews.

views have been manually translated into English and the statistics about the English translations of the Hindi reviews are shown in the third column of Table 2. In addition to the sample Hindi reviews, Table 1 shows the corresponding English translations of the Hindi reviews.

2.2 The English Review data

As discussed above, for complaint identification in English we required labeled training data for building classifiers (complaint identifiers). Accordingly, we created labeled training data for English. For this, we followed the data creation and annotation methods described in Singh et al. (2020). First, we took English reviews from Amazon review dump.[9] We sampled reviews from four different categories, namely Books, Cell_Phones_and_Accessories, Electronics, and Movies_and_TV. The Hindi reviews which we collected from the websites of Amazon India and YouTube were mainly on books and electronic goods. This is the reason why we considered English reviews on those four (related) product categories. As for data cleaning and preprocessing, we adopted the same steps as applied for Hindi (cf. Section 2.1). Table 3 presents the statistics of the English dataset (the training and development sets).

Table 3: Statistics of the train and development sets (English reviews).

	Reviews	Words	Complaints
Train set	8,026	3,84,467	4,013
Dev. set	400	17,873	200

3 The Hindi-to-English MT Systems

Our MT systems are Transformer models (Vaswani et al., 2017) which were trained using the Marian-NMT toolkit.[10] The tokens of the training, evaluation and validation sets are segmented into sub-word units using Byte-Pair Encoding (BPE) (Sennrich et al., 2016), and BPE is applied individually on the source and target languages. From our experiences (Jooste et al., 2020; Haque et al., 2020b,c; Nayak et al., 2020b,a; Parthasarathy et al., 2020) in the participation in the recent shared translation tasks (Barrault et al., 2020; Mayhew et al., 2020; Nakazawa et al., 2020) involving low-resource language pairs and domains, we found that the following configuration usually leads to the best results in the low-resource translation settings: (i) the BPE vocabulary size: 6,000, (ii) the sizes of the encoder and decoder layers: 4 and 6, respectively, and (iii) learning-rate: 0.0003. As for the remaining hyperparameters, we followed the recommended best setup from Vaswani et al. (2017). The early stopping criterion is based on cross-entropy; however, the final NMT system is selected as per the highest BLEU score on the validation set. The beam size for search is set to 6. We make our final NMT model with ensembles of 8 models that are sampled from the training run.

For building our baseline models (forward

[9] https://jmcauley.ucsd.edu/data/amazon/

[10] https://github.com/marian-nmt/marian

and backward), we used the IIT Bombay English-Hindi parallel corpus[11] (Kunchukuttan et al., 2017) that is compiled from a variety of existing sources, e.g. OPUS[12] (Tiedemann, 2012). After applying standard cleaning procedures including applying a language identifier[13] we are left with just over 1.1 million parallel sentence pairs. As for Hindi and English monolingual sentences for forward-translation and back-translation, respectively, we sampled them from the AI4Bharat-IndicNLP Corpus (Kunchukuttan et al., 2020) and Amazon review dump (cf. Section 2.2), respectively. Table 4 presents the corpus statistics. As above (cf. Section 2.1), for our development set we used 385 reviews from Amazon India and YouTube, which were then manually translated into English (cf. last row of Table 4).

	sentences	words (EN)	words (HI)
Train	1,102,511	22.4M	23.4M
Monolingual			
English	6.86M	121.3M	–
Hindi	7.82M	–	142.9M
Dev. set	385	6,952	7,209

Table 4: The Corpus statistics.

We present the performance of our MT systems in terms of the automatic evaluation metric BLEU (Papineni et al., 2002). Additionally, we performed statistical significance tests using bootstrap resampling methods (Koehn, 2004). We obtained the BLEU scores of our MT systems on the test set, and the scores are reported in Table 5. The first row of Table 5 represents our baseline Hindi-to-English MT system. The English-to-Hindi MT system which has been used to translate the English monolingual sentences (reviews) into Hindi produced 20.52 BLEU points on the development set. The BLEU scores of the MT systems (Base+BT and Base+BT+FT) trained on training data that consists of both authentic and (target- or/and source-original) synthetic parallel data are shown in the next two rows of Table 5. As in Caswell et al. (2019), in order to let the NMT model know that the given source is synthetic, we tag the source sentences of the synthetic data with the extra tokens.

	BLEU	
	devset	test set
Base	25.92	23.03
Base+BT	30.84	26.51
Base+BT+FT	30.89	26.85
Base+BT+FT+DA	31.52	27.49

Table 5: The BLEU scores of the English-to-Hindi NMT systems.

We observed that the review texts generally contain terms or product names, and terminology translation is a challenging task in MT (Haque et al., 2019a, 2020a). In order to adapt our best MT system, Base+BT+FT, to the task, we adopted the terminology-aware on-the-fly adaption method Jooste et al. (2020); Haque et al. (2020c); Nayak et al. (2020b); Parthasarathy et al. (2020), and mine those sentences from large monolingual datasets that could be beneficial for fine-tuning the original NMT model. As in Jooste et al. (2020); Haque et al. (2020c); Nayak et al. (2020b); Parthasarathy et al. (2020), we first identified terms in the review test set (cf. Table 2) to be translated,[14] and given the list of extracted terms, Hindi sentences which were mined from large monolingual data are similar in style to the test set sentences. We mined Hindi sentences (a total of 129,800 sentences) from a large monolingual corpus given the list of terms (a total of 2,953 terms) appearing in the test set. Then, a source-original synthetic corpus was created by translating these mined Hindi sentences into English using the best MT system, Base+BT+FT. The monolingual corpus that we used for this purpose contains 62,679,936 sentences from the AI4Bharat-IndicNLP Corpus. Additionally, we mined 36,397 sentences from the source side of the parallel training corpus and took their target counterparts, which gives us an authentic parallel corpus for adaptation. Finally, Base+BT+FT was fine-tuned on the resultant training corpus (166,197 training instances which contains 129,800 synthetic and 36,397 authentic sentence-pairs). As for translating the development set sentences, we fol-

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

[12] http://opus.lingfil.uu.se/

[13] https://pypi.org/project/pycld2/

[14] We followed Haque et al. (2014, 2018) in order to automatically identify terms in the in-domain texts.

lowed the same strategy.

The BLEU scores of the adapted MT system (Base+BT+FT+DA) on the test set are shown in the last row of Table 5. When we compare the original MT system with the adapted MT system, we see that the adapted version of Base+BT+FT, Base+BT+FT+DA, produces a 0.64 BLEU point (corresponding to 2.38% relative) improvement over Base+BT+FT, and the improvement is statistically significant.

4 The Complaint Identification Models

4.1 LSTM Network

Nowadays, recurrent neural networks (RNN), in particular long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) hidden units, have proven to be an effective model for many classification tasks in NLP, e.g. sentiment analysis (Wang et al., 2016), text classification (Joulin et al., 2016; Zhou et al., 2016). RNN is an extension of the feed-forward neural network (NN), which has the gradient vanishing or exploding problems. LSTM deals with the gradient vanishing and exploding problems of RNN. An RNN composed of LSTM hidden units is often called an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. More formally, each cell in LSTM can be computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \tag{1}$$

$$f_t = \sigma(W_f \cdot X + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{3}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh\ (W_c \cdot X + b_c) \tag{5}$$

$$h_t = o_t \odot tanh\ (c_t) \tag{6}$$

where $W_i, W_f, W_o \in \mathbb{R}^{d \times 2d}$ are the weighted matrices and $b_i, b_f, b_o \in \mathbb{R}^d$ are the biases of LSTM, which need to be learned during training, parameterising the transformations of the input, forget and output gates, respectively. σ is the sigmoid function, and $\odot$ stands for element-wise multiplication. x_t includes the inputs of LSTM cell unit. The vector of hidden layer is h_t. The final hidden vector h_N represents the whole input review, which is passed to the *softmax* layer after linearising it into a vector whose length is equal to the number of class labels. In our work, the set of class labels includes complaint and non-complaint categories.

4.2 Classical Supervised Classification Models

Furthermore, we compare the LSTM network with classical supervised classification models. We employ the following classical supervised classification techniques in our experiments:

- Logistic Regression (LR)
- Decision Tree (DT)
- Random Forest (RF)
- Naïve Bayes (NB)
- Support Vector Machine (SVM)

These classical learning models (LR, DT, RF, NB and SVM) can be viewed as the baseline in this task. Thus, we obtain a comparative overview on the performance of different supervised classification models including the LSTM network.

4.3 Training Setup

In order to build LR, DT, RF and NB classification models, we use the well-known scikit-learn machine learning library,[15] and performed all the experiments with default parameters set by scikit-learn. As for the representation space, each review was represented as a vector of word unigrams weighted by their frequency in the reviews.

For the classifiers based on the neural networks, we use a 300-Dimensional word embeddings from *fastText*. We use the *sigmoid* activation function with the Adam optimizer (Kingma and Ba, 2014) and binary cross entropy loss function. The size of the input layer of the NN is 300. We employ layer normalisation (Ba et al., 2016) in the model. Dropout

[15] https://scikit-learn.org/stable/

(Gal and Ghahramani, 2016) between layers is set to 0.10. The size of the embedding and hidden layers are 300. The models are trained with learning-rate set to 0.0003 and reshuffling the training examples for each epoch.

5 Results and Discussion

We evaluate the performance our classifiers on the gold-standard test set (cf. Table 2) and report the evaluation results in this section. In order to measure a classifier's accuracy on the test set, we use three widely-used evaluation metrics: precision, recall and F_1 measures. The results obtained are reported in Table 6. The first five columns of Table 6 represent our baseline classifiers (i.e. the classical supervised classification models). We see from the table that these classifiers perform moderately to excellently and LR is the best-performing method among them (LR: a 70.35 F_1 score)) when tested on the translations by our best MT system (Base+BT+FT+DA). This classifier (LR) produces an F_1 score of 71.47 on the gold-standard test set (i.e. translations of the Hindi reviews by translators).

Table 6: Performance of the classifiers on the evaluation test set.

	NB	LR	DT	SVM	RF	LSTM
Base						
P	57.64	66.98	62.82	66.82	68.78	75.76
R	41.50	72.00	73.5	69.50	70.50	75.00
F_1	48.26	69.4	67.74	68.13	69.63	75.38
Base+BT						
P	58.78	71.28	61.50	70.00	69.85	77.44
R	43.50	69.50	69.50	63.00	69.50	75.50
F_1	50.00	70.38	65.26	66.32	69.67	76.46
Base+BT+FT						
P	59.49	70.71	64.38	68.98	70.62	76.12
R	47.00	70.00	70.5	64.50	68.5)	76.5
F_1	52.51	70.35	67.30	66.67	69.54	76.31
Base+BT+FT+DA						
P	58.17	70.71	65.33	68.56	70.47	77.16
R	44.50	70.00	73.50	66.50	68.00	76.00
F_1	50.43	**70.35**	69.18	67.51	69.21	**76.58**
Manual (Upper Bound)						
P	58.22	73.55	63.55	71.82	71.42	80.10
R	42.50	69.50	68.00	65.00	67.50	78.50
F_1	49.13	**71.47**	65.70	68.24	69.41	**79.29**
Hindi classifiers on Hindi reviews						
P	59.09	74.14	77.77	72.16	84.51	74.71
R	65.00	76.00	31.50	70.00	30.00	65.00
F_1	61.91	**75.06**	44.84	71.07	44.28	69.52

As for our NN-based classifier, the LSTM network trained on fastText embeddings performed excellently as we see from Table 6, where it obtains an excellent F_1 score (76.58 F_1) on the test set of translations of the Hindi reviews by Base+BT+FT+DA. It obtains an F_1 score of 79.29 on the test set of translations of the Hindi reviews by human translators.

For comparison, we also measured the performance of the Hindi classifiers that were built on the Hindi training data released by Singh et al. (2020) on the original reviews (i.e. the Hindi-side of the test set; cf. Table 2), and the results are shown in the last rows of Table 6. We see from Table 6 that in this case, the best-performing Hindi classifier is LR and it produces an F_1 of 75.06, which is 1.52 F_1 points lower than that produced by the best-performing English classifier on the translations by our best MT system.

We clearly see from the scores presented in Table 6 that the performance of the English classifiers on the translations produced by our customised MT system (Base+BT+FT+DA) is comparable to that of the Hindi classifiers on the original Hindi reviews. Thus, we can say that MT when customised or trained to translate texts of specific styles (e.g. reviews about a variety of products) can act as an alternative to the tools that rely on task-specific (in this work, complaint identification) training data which is expensive to prepare.

6 Conclusion

In this paper, we presented a strategy in which MT can be used to eliminate the requirement for expensive task-specific data creation for low-resource languages or domains. We investigated our strategy on complaint identification from reviews about products posted in Hindi. We used state-of-the-art NMT models for translating the Hindi reviews into English, and investigate the performance of the English complaint identifiers on the translations of the Hindi reviews by the Hindi-to-English MT systems. For comparison, we tested the performance of the English classifiers on two setups: (i) English translations of the Hindi reviews by the MT systems, and (ii) gold-standard English translations of the Hindi reviews. We also compared the performance of the Hindi classifiers built on a publicly available Hindi review training data set on the original Hindi reviews (i.e. the Hindi-side of the review test set).

In our experiments, we also aimed at preserving source-language stylistic properties and semantics in translation. For this, we applied standard and commonly-used data augmentation techniques and terminology-aware domain adaptation method (Jooste et al., 2020; Haque et al., 2020c; Nayak et al., 2020b; Parthasarathy et al., 2020) for building our Hindi-to-English NMT systems, and used task-specific target-language monolingual data. These strategies were found to be effective in this task. We demonstrated that the NMT systems when customised or trained to translate texts of specific styles (e.g. user-generated content or reviews) can act as an alternative to those tools that require task-specific (i.e. complaint identification) training data which are expensive to create.

We believe that this work would bring additional value to the social media analytics research and practice given the fact that many task-specific data are available in English only and does not exist in many low-resource and even some high-resource languages.

In future, we intend to test our method on different low-resource and high-resource non-English languages. We also plan to investigate this method on different NLP tasks.

Acknowledgments

This research has been supported by the ADAPT Centre for Digital Content Technology which is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–54, Online. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *CoRR*, abs/1512.05287.

Vineet Gupta, Devesh Varshney, Harsh Jhamtani, Deepam Kedia, and Shweta Karwa. 2014. Identifying purchase intent from social posts. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, pages 180–186, Ann Arbor, Michigan.

Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2019a. Investigating terminology translation in statistical and neural machine translation: A case study on English-to-Hindi and Hindi-to-English. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 437–446, Varna, Bulgaria.

Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2020a. Analysing terminology translation errors in statistical and neural machine translation. *Machine Translation (in press)*, 34(2):149–195.

Rejwanul Haque, Yasmin Moslem, and Andy Way. 2020b. The ADAPT system description for the STAPLE 2020 English-to-Portuguese translation task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 144–152, Online. Association for Computational Linguistics.

Rejwanul Haque, Yasmin Moslem, and Andy Way. 2020c. Terminology-aware sentence mining for nmt domain adaptation: Adapt's submission to the adap-mt 2020 english-to-hindi ai translation shared task. In *Proceedings of the Workshop on Low Resource Domain Adaptation for Indic Machine Translation (Adap-MT 2020)*, Patna, India (to appear).

Rejwanul Haque, Sergio Penkale, and Andy Way. 2014. Bilingual termbank creation via log-likelihood comparison and phrase-based statistical machine translation. In *Proceedings of the 4th International Workshop on Computational*

Terminology (Computerm), pages 42–51, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2018. TermFinder: log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Language Resources and Evaluation*, 52(2):365–400.

Rejwanul Haque, Arvind Ramadurai, Mohammed Hasanuzzaman, and Andy Way. 2019b. Mining purchase intent in twitter. *Computación y Sistemas*, 23(3).

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

Wandri Jooste, Rejwanul Haque, and Andy Way. 2020. The ADAPT Centre's neural MT systems for the WAT 2020 document-level translation task. In *Proceedings of the the 7th Workshop on Asian Translation (WAT 2020), AACL-IJCNLP 2020*, pages 142–146, Suzhou, China.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The IIT Bombay English–Hindi parallel corpus. *CoRR*, 1710.02855.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 232–243, Online. Association for Computational Linguistics.

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*, pages 3111–3119.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, Suzhou, China. Association for Computational Linguistics.

Prashanth Nayak, Rejwanul Haque, and Andy Way. 2020a. The ADAPT Centre's participation in WAT 2020 english-to-odia translation task. In *Proceedings of the the 7th Workshop on Asian Translation (WAT 2020), AACL-IJCNLP 2020*, pages 114–117, Suzhou, China.

Prashanth Nayak, Rejwanul Haque, and Andy Way. 2020b. The ADAPT's submissions to the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation (Shared Task Papers (Biomedical)*, pages 839–-846, Online Conference.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Venkatesh Balavadhani Parthasarathy, Akshai Ramesh, Rejwanul Haque, and Andy Way. 2020. The ADAPT system description for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 261-–267, Online Conference.

Daniel Preotiuc-Pietro, Mihaela Gaman, and Nikolaos Aletras. 2019. Automatically identifying complaints in social media. *arXiv preprint arXiv:1906.03890*.

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

Raghvendra Pratap Singh, Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2020. Identifying complaints from product reviews: A case study on hindi. In *Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2020)*, pages 217–228, Dublin, Ireland.

Amirhossein Tebbifakhr, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. Machine translation for machines: the sentiment classification use case. 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 1368–1374, Hong Kong, China.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 2214–2218, Istanbul, Turkey.

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 6000–6010.

Jinpeng Wang, Gao Cong, Wayne Xin Zhao, and Xiaoming Li. 2015. Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 318–324, Austin, TX.

Yequan Wang, Minlie Huang, Li Zhao, et al. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, Austin, TX.

Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*.