Machine Translation
https://doi.org/10.1007/s10590-021-09291-z

Investigating the roles of sentiment in machine translation

Sainik Kumar Mahata[1] · **Dipankar Das**[1] · **Sivaji Bandyopadhyay**[1]

Received: 29 April 2021 / Accepted: 17 November 2021

Abstract
Parallel corpora are central to translation studies and contrastive linguistics. However, training machine translation (MT) systems by barely using the semantic aspects of a parallel corpus leads to unsatisfactory results, as then the trained MT systems are likely to generate target sentences that are semantically and pragmatically different from the source sentence. In the present work, we explore the improvement in the performance of an MT system when pragmatic features such as sentiment are introduced during its development. The language pair used for the experiments is English (source language) and Bengali (target language). The improvement in the MT output, before and after the introduction of sentiment features, is quantified by comparing various translation models, such as SMT, NMT and a newly developed translation model SeNA, with the help of automated (BLEU and TER) and manual evaluation metrics. In addition, the propagation of sentiment during the translation process is also studied extensively. We observe that the introduction of sentiment features during the system development process helps in elevating the translation quality.

Keywords Machine translation · Sentiment analysis · Parallel corpus · Neural networks

1 Introduction

Machine translation (MT)-related research has been carried out over several decades. MT has become capable of mimicking human translations for many language pairs. However, for many language pairs the quality is still poor as MT systems fail

✉ Sainik Kumar Mahata
skmahata.cse.rs@jadavpuruniversity.in

Dipankar Das
dipankar.dipnil2005@gmail.com

Sivaji Bandyopadhyay
sivaji_cse_ju@yahoo.com

[1] Jadavpur University, Kolkata, India

Published online: 09 December 2021

Table 1 Problems in translation quality when pragmatic-level information is not considered during training

Source (English)	Target (Bengali)	Change (Meaning/Sentiment)
Let's not go.	চল না (Let's Go)	Meaning
It's impossible not to like you	তোমাকে পছন্দ করা অসম্ভব (It's impossible to love you)	Meaning/Sentiment
Never believe everything you hear	আপনি যা শুনেছেন তা কখনও বিশ্বাস করবেন না (Whatever you hear, don't trust it)	Meaning

to capture the semantic and pragmatic issues involved during translation. This holds true for both resource-rich as well as resource-scarce languages. Training an MT system often requires a large, good quality parallel corpus (Resnik 1998, 1999). However, training an MT system using only a parallel corpus leads it to learn at best the syntactic nuances of the participating languages. Hence, translation quality takes a hit as such systems do not learn to avail of any semantic or pragmatic information during translation. This leads to poorer quality translations where the source and the translated sentence differ in meaning and sometimes in sentiment too (Lohar et al. 2017) (Table 1).

MT output is difficult to judge from pure linguistic perspectives only and should involve more cognitive, pragmatic and psycholinguistic aspects (Doherty et al. 2010; Joshi et al. 2010; Pal et al. 2014). It has often been observed in MT that sentiment preservation between the source and target sides of the language pair under consideration can replace the need for automatic post-editing (Pal et al. 2014). Given this, we were eager to investigate whether translation quality improves if pragmatic features such as sentiment are augmented during the MT system training phases. To answer this question, we followed two approaches.

The first approach deals with introduction of sentiment features in the raw parallel corpus itself and subsequently modifying the corpus, so that sentences become parallel to each other based on their meaning as well as sentiment. This is a supervised approach that uses Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) based on the Transformer architecture (Vaswani et al. 2017), trained using an English–Bengali parallel corpus tagged with sentiment features. In other words, as well as having the default semantic alignment, the training sentences of the parallel corpus are also aligned from a pragmatic perspective. This strategy is considered in the current paper, as parallel data has a major impact on the performance of an MT system and augmenting features directly into it can lead to gains in performance. We name the two systems $\mathbf{SMT_{Senti}}$ and $\mathbf{NMT_{Senti}}$, respectively.

The second approach does without this overhead, and learns to generate target sentences based on the sentiment of the source sentence. We build a semi-supervised character-based NMT system, that generates target sentences which are passed through a sentiment inference system, the output of which is then compared with the sentiment of the source sentence and the error calculated. Back-propagation uses this error to recalculate the weights of the intermediate nodes accordingly. We name this system Sentiment-assisted NMT Architecture (**SeNA**).

Furthermore, to test the effectiveness of the two approaches, we prepared two baseline systems ($\mathbf{SMT_{Base}}$ and $\mathbf{NMT_{Base}}$), which were trained using a general, non-sentiment-tagged English–Bengali parallel corpus. The results after comparison were quantified using the standard evaluation metrics BLEU (Papineni et al. 2002), Translation Edit Rate (TER) (Snover et al. 2006), fluency and adequacy [cf. Way (2018)]. Finally, we checked the amount of propagation of sentiment through the MT pipeline, i.e., we quantified the instances where the sentiment of the source and translated sentences matched/mismatched and the observations regarding the same.

In addition, we recognized that MT systems work better when trained using simple sentences only (Mahata et al. 2018). Given that observation, we were interested to check whether the statement holds true for sentiment-tagged simple sentences. In this regard, we prepared two additional sentiment-tagged parallel corpora: one comprising of only simple sentences, and another consisting of "Other" (complex and compound) sentences. The translation models were also trained using these corpora, and the results were documented.

When testing our models, we observed that MT systems trained using sentiment-tagged parallel data outperformed the baseline systems trained using data not tagged for sentiment. Moreover, our semi-supervised approach outperformed both our supervised and baseline systems and upon manual evaluation, showed greater preservation of sentiment between the source-translated sentence pairs.

The paper is organized as follows. Section 2 describes a brief survey of the work done in this area to date. Section 3 describes the methodology of data collection, segmenting the same according to complexities and preparation of the sentiment-augmented parallel data. Section 4 describes the various MT systems that were used in our experiments. Section 5 discusses the results and compares the sentiment-augmented models against the baseline systems. This is followed by our concluding remarks in Sect. 6.

2 Related work

To the best of our knowledge, relatively little work has been attempted so far that uses text simplification as a pre-processing tool for MT and quantifies its use in improving the quality of translation output. Tyagi et al. (2015) developed a classifier-based Text Simplification Model for English–Hindi MT using Support Vector Machines and Naïve Bayes classifiers. Similarly, Štajner and Popovic (2016) experimented with English-to-Serbian translation and showed that the use of more aggressive text simplification methods (which not only simplify the given sentence but also discard irrelevant information, thus producing syntactically simple sentences) also improves meaning preservation (adequacy) of the translation output. They used three state-of-the-art simplification approaches, namely lexical simplification, syntactic simplification, and content reduction. In contrast, Poornima et al. (2011) used rule-based techniques to simplify complex sentences based on connectives like relative pronouns, coordinating and subordinating conjunction. Punctuation marks were used as delimiters to split the sentences, and the simplification was done based on connectives. They claimed this method was useful as a pre-processing tool for MT.

However, all the previous work described here used simplification methods based either on machine learning methods or semi-syntactic rules. Our method differs in the fact that we have not used any text simplification algorithms to improve performance. Instead, we have incorporated sentiment as an additional attribute to improve the quality of MT. In addition, we used the full parse structures to classify the sentences, without having to simplify them sentences into simple, complex and compound categories using deep learning methods.

Similarly, the use of sentiment analysis to improve the quality of MT is a field that is not well explored and very little work has been done so far. Some research involved the development of sentiment lexicons and cross-lingual sentiment identification. Banea et al. (2008) generated resources for subjectivity augmentation in Spanish and Romanian using English corpora. Afli et al. (2017) built an Irish sentiment analysis system called *SentiWordTweet* for the analysis of Irish language tweets for the Irish General Election, using *Senti-Foclóir*, the first sentiment lexicon created for Irish. In the context of Indian languages, Das and Bandyopadhyay (2010) showed the development of a Bengali sentiment lexicon using an English-to-Bengali MT system. Similarly, Joshi et al. (2010) developed a Hindi sentiment lexicon using an English-to-Hindi MT system. Kanayama et al. (2004) developed a high-precision sentiment analysis system by making use of an existing transfer-based MT engine. Pal et al. (2014) showed how sentiment analysis can improve translation quality by incorporating the roles of sentiment holders, sentiment expressions and their corresponding objects and relations.

Work done by Lohar et al. (2017, 2018) focuses on developing multiple MT models according to different sentiment for translation of English tweets to German, which is adapted in Lohar et al. (2019) for the translation of English IMDb user movie reviews into Serbian. In contrast, our novel work concerns the development of a sentiment-augmented parallel corpus with sentences of various complexities and its deployment to improve translation quality by producing sentiment-preserved translations.

3 Data preparation

The present work has two main objectives. The first is to confirm whether simple sentences improve the quality of MT, and the second is to check whether the addition of sentiment enhances translation quality. In this regard, we obtained various versions of our parallel corpus in different stages.

The methodology of developing the sentiment-augmented parallel corpus is shown in Fig. 1. The first step consisted of the data collection process and the subsequent development of the English–Bengali parallel corpus. Thereafter, a stacked recurrent neural network (RNN)-based classifier was developed that groups the English sentences into simple, complex or compound sentences. Corresponding to the English sentences of various complexities, their Bengali counterparts were also organized to develop parallel corpora of different complexities. Later, sentiment analysis models for both the English and Bengali sentences

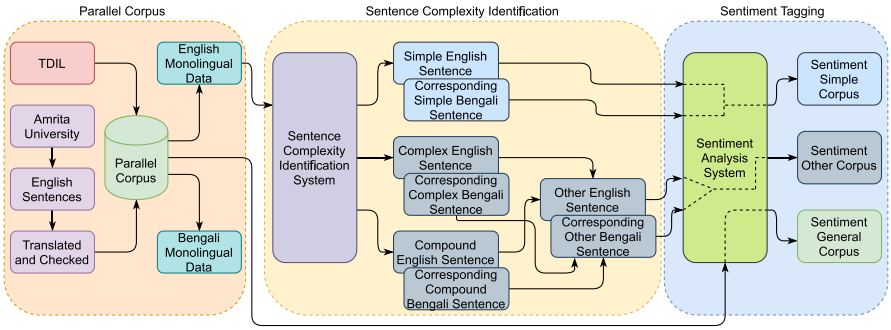

Fig. 1 Model for development of sentiment-tagged parallel corpus

were developed, which tagged the parallel sentences based on their sentiment and finally led to the development of the sentiment-augmented parallel corpus. All these steps are discussed in detail in the following sections.

3.1 Parallel corpus

For preparation of the initial English–Bengali parallel corpus, we collected 49,999 parallel sentences from the resource developed by Technology Development for Indian Languages Programme (TDIL).[1] Since having a large collection of parallel data is often crucial in improving MT performance, we collected an additional 57,985 English sentences from the resource of the Machine Translation in Indian Languages (MTIL) shared task,[2] organized by Amrita University. Thereafter the English sentences from MTIL were translated to Bengali, using the Google Translate API[3] for Python. The translated sentences were then checked manually and resulted in the formation of an English–Bengali parallel corpus consisting of 107,984 aligned sentences. For simplicity, we name this corpus $\mathbf{PC}_{\mathbf{Gen}}$, and the corresponding monolingual corpora $\mathbf{PC\_EN}_{\mathbf{Gen}}$ and $\mathbf{PC\_BN}_{\mathbf{Gen}}$ for English and Bengali, respectively.

3.2 Sentence complexity identification

To ratify our objective of classifying the raw parallel corpus into three subsets, based on sentence complexity (simple, complex and compound), a stacked RNN-based deep learning system was developed, as manual annotation identifying the complexity of sentences was not feasible.

[1] http://tdil.meity.gov.in/.
[2] http://nlp.amrita.edu/mtil_cen/.
[3] https://pypi.org/project/googletrans/.

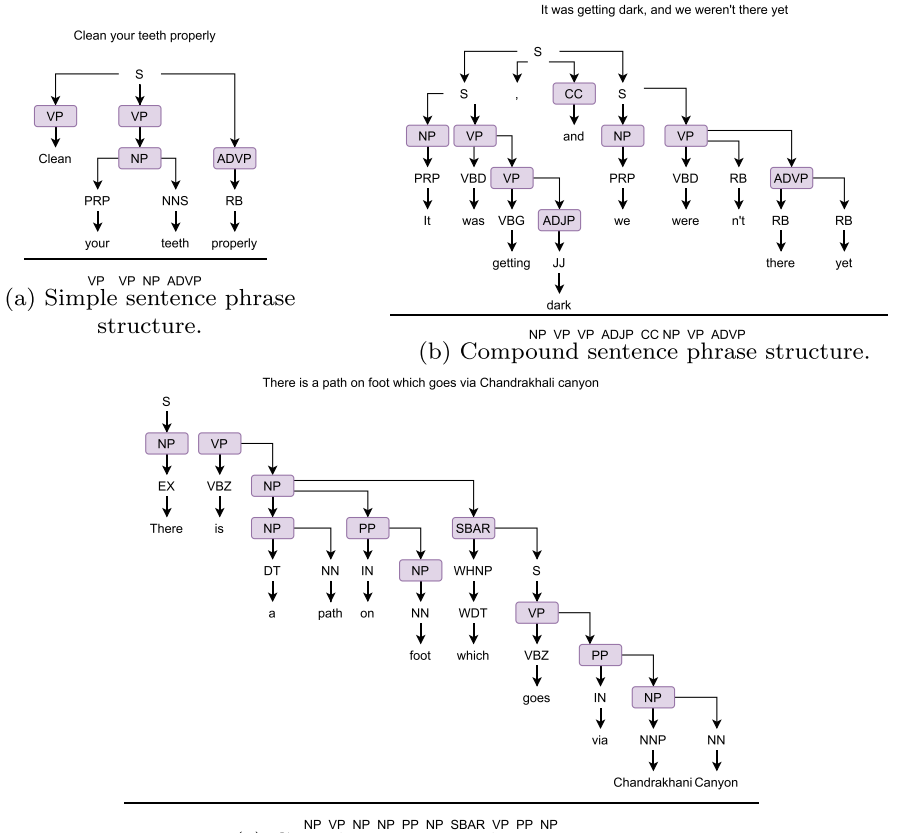

Fig. 2 Phrase structures of simple, complex and compound sentences

To train the classifier, a training dataset consisting of 3222 simple, 23,207 complex and 7548 compound sentences was developed by two linguists who were proficient in the English language. The Fleiss' Kappa (Fleiss and Cohen 1973) for the inter-annotator agreement, came to 0.932.

Subsequently, we POS-tagged the sentences using the Natural Language Tool Kit[4] (NLTK). A simple sentence is defined as a sentence that contains only one independent clause and has no dependent clauses. Generally, whenever two or more clauses are joined by conjunctions (coordinating and subordinating), it becomes a complex or a compound sentence. Accordingly, to handle cases involving conjunctions, we shallow-parsed the English sentences using the Stanford Parser[5] to extract phrase information: NP (Noun Phrase), VP (Verb Phrase), PP (Prepositional Phrase),

[4] http://www.nltk.org/.

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

Table 2 Example of POS tagging and shallow parsing

Sentence	POS tagging	Shallow parsing
The enemy soldiers surrendered to us	('The', 'DT'), ('enemy', 'NN'), ('soldiers', 'NNS'), ('surrendered', 'VBD'), ('to', 'TO'), ('us', 'PRP'), ('.', '.')	S (NP (DT The) (NN enemy) (NNS soldiers)) (VP (VBD surrendered) (PP (TO to) (NP (PRP us)))) (. .))

ADJP (Adjectival Phrase) and ADVP (Adverbial Phrase). An example of the phrase structures of sentences with different complexities is shown in Fig. 2.

We avoided POS tagging and shallow parsing the Bengali sentences $\mathbf{PC\_BN_{Gen}}$, as no standard library was available for the same. We hypothesized that parallel English–Bengali sentences should have the same complexity. An example of POS tagging and shallow parsing is shown in Table 2.

Since the training data had an unequal number of simple, complex and compound sentences, we assigned weights to the classes using the sklearn[6] package. Our model took as input the words of the sentences, the POS tags of the words and the phrase structure of the sentence. The respective embeddings were concatenated and the tensors passed to a stacked bidirectional Long Short Term Memory (LSTM) layer (Hochreiter and Schmidhuber 1997). The complexity labels were then mapped to the output of the LSTM layer through a Dense layer. A schematic diagram of the model is shown in Fig. 3. Other parameters of the model are as follows:

- Activation function: softmax
- Optimizer: adam
- Loss function: sparse categorical cross-entropy.
- Embedding dimension: 100

1000 sentences were used to test the model. An accuracy of 98.15% and F1 score of 0.97 were generated. The output was also tested by the same linguists and an agreement score using Fleiss' Kappa was calculated as 0.98.

We ran the source part $\mathbf{PC\_EN_{Gen}}$ of the general corpus through this set-up and extracted 16,654 simple, 45,091 complex and 46,239 compound sentences, respectively. The classified sentences were again ratified by the same linguists. and the subsets were named $\mathbf{PC_{Simple}}$, $\mathbf{PC_{Complex}}$ and $\mathbf{PC_{Compound}}$, respectively.

We assumed the corresponding Bengali sentences in $\mathbf{PC\_BN_{Gen}}$ to have the same complexity. We merged the complex and compound sentence corpus and named

[6] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html.

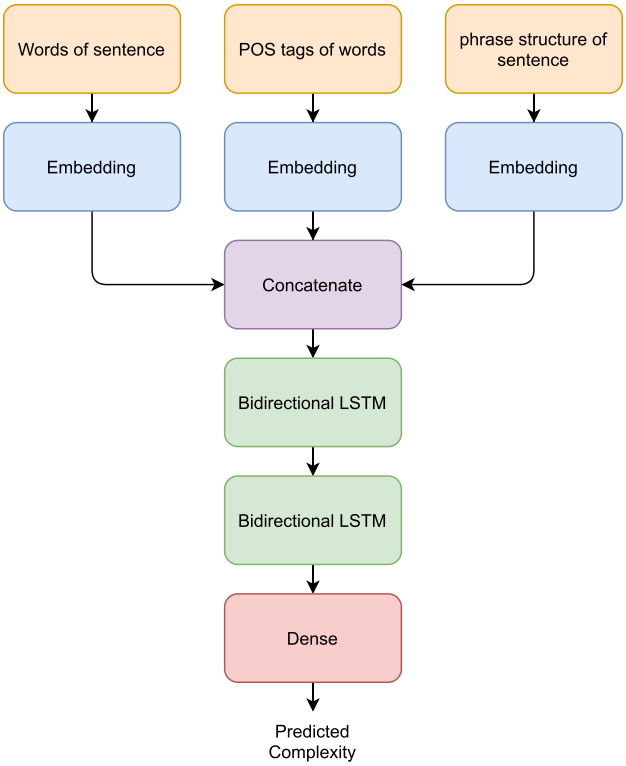

Fig. 3 Model for predicting complexity of a sentence

it $\mathbf{PC}_{\mathbf{Other}}$. Having done all this, we needed to annotate the data with sentiment as well, and we turn to this next.

3.3 Sentence sentiment identification

Sentiment analysis refers to the use of NLP to systematically identify, extract, quantify, and study affective states and subjective information. In case of sentiment, we planned to annotate both the English and Bengali corpus for the following reasons:

(i) To check whether sentiment is propagated 'as is' during translation, and
(ii) If so, whether any sentiment clue exists to help us link the source and target sentences.

Both our approaches (supervised and unsupervised) need the sentences in English and Bengali to be annotated with their sentiment. For this, we developed a sentiment analysis system that takes as input the words of the sentences, extracts the embedding using the embedding layer and then subjects the tensors to a bidirectional

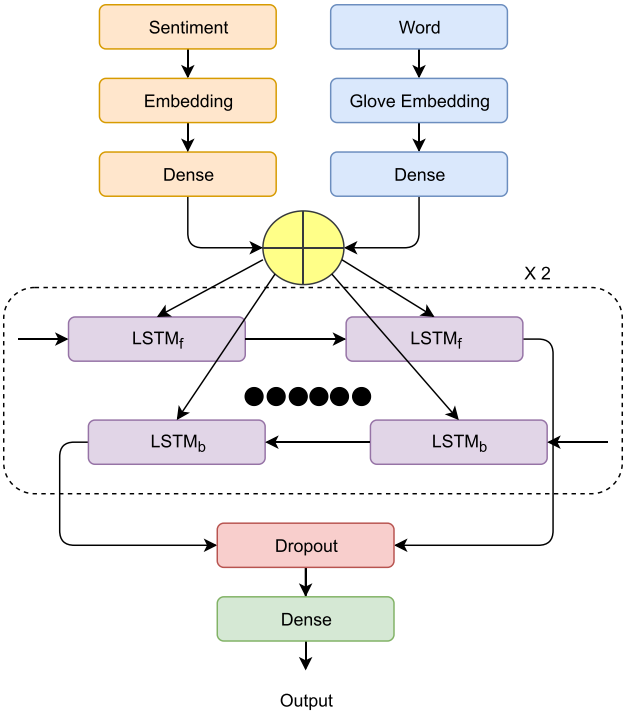

Fig. 4 Sentiment analysis system

LSTM layer. The output is mapped through a dense layer to its respective sentiment labels. Other parameters of the developed system are as follows:

- Activation: softmax
- Optimizer: adam
- Loss function: categorical cross-entropy
- Embedding dimension: 100

A schematic diagram of the developed sentiment analysis system is given in Fig. 4. For the sentiment analysis of the English language, the IMDB dataset[7] was used. The model was tested on 1000 sentences and it returned an accuracy of 86.184%. The tags were also checked by a linguist and an agreement score using Fleiss' Kappa was calculated as 0.926.

For the sentiment analysis of the Bengali language, we collected 24,802 polarity-tagged Bengali sentences from Kaggle,[8] Github[9] and Mendeley.[10] We collected

[7] https://ai.stanford.edu/~amaas/data/sentiment/.
[8] https://www.kaggle.com/tazimhoque/bengali-sentiment-text.
[9] https://github.com/socianltd/socian-bangla-sentiment-dataset-labeled.
[10] https://data.mendeley.com/datasets/n53xt69gnf/3.

Table 3 Statistics of general corpus data derived from the data preparation module

Corpus: $\mathbf{PC_{Gen}}$			
No. of sentences: 1,07,984			
Particulars	$\mathbf{PC^{Senti}_{Gen}}$	$\mathbf{PC\_BN^{Senti}_{Gen}}$	$\mathbf{PC\_EN^{Senti}_{Gen}}$
Positive	46,579	46,579	48,591
Negative	39,246	39,246	42,111
Neutral	17,282	22,159	17,282
Not parallel*	4877		
Total	107,984		

*'Not parallel' in this case means that the source–target sentence pairs do not have the same sentiment

Table 4 Statistics of simple corpus data derived from the data preparation module

Corpus: $\mathbf{PC_{Simple}}$			
No. of sentences: 16,654			
Particulars	$\mathbf{PC^{Senti}_{Simple}}$	$\mathbf{PC\_BN^{Senti}_{Simple}}$	$\mathbf{PC\_EN^{Senti}_{Simple}}$
Positive	7494	7521	7494
Negative	6398	6398	6494
Neutral	2666	2735	2666
Not parallel*	96		
Total	16,654		

*'Not parallel' in this case means that the source–target sentence pairs do not have the same sentiment

another 50,000 sentiment-tagged sentences from the work by Das and Bandyopadhyay (2013). After training, the model was tested using 1000 test sentences and an accuracy figure of 82.95% was returned. The output tags were also tested by a linguist and the agreement score using Fleiss' Kappa was calculated as 0.882.

Thereafter, both the $\mathbf{PC\_EN_{Gen}}$ and $\mathbf{PC\_BN_{Gen}}$ data were run through the respective sentiment analysis system and tagged with their respective sentiments. This step generated two separate corpora: $\mathbf{PC\_EN^{Senti}_{Gen}}$ and $\mathbf{PC\_BN^{Senti}_{Gen}}$. For building the sentiment-augmented parallel corpus $\mathbf{PC^{Senti}_{Gen}}$, we considered only those parallel English and Bengali sentences which had the same sentiment tags. Similarly, we ran the $\mathbf{PC_{Other}}$ and $\mathbf{PC_{Simple}}$ corpora through this system to generate four monolingual corpora: $\mathbf{PC\_EN^{Senti}_{Other}}$, $\mathbf{PC\_BN^{Senti}_{Other}}$, $\mathbf{PC\_EN^{Senti}_{Simple}}$ and $\mathbf{PC\_BN^{Senti}_{Simple}}$. Two parallel corpora were generated: $\mathbf{PC^{Senti}_{Other}}$ and $\mathbf{PC^{Senti}_{Simple}}$. A quantitative analysis of the data derived from the data preparation module is presented in Tables 3, 4 and 5.

Table 5 Statistics of other corpus data derived from the data preparation module

Corpus: $\mathbf{PC_{Other}}$			
No. of sentences: 91,330			
$\mathbf{PC_{Other}}$		$\mathbf{PC_{Complex}}$: 45,091 $\mathbf{PC_{Compound}}$: 46,239 Total: 91,330	
Particulars	$\mathbf{PC^{Senti}_{Other}}$	$\mathbf{PC\_EN^{Senti}_{Other}}$	$\mathbf{PC\_EN^{Senti}_{Other}}$
Positive	41,097	41,651	41,097
Negative	35,617	36,207	35,617
Neutral	13,472	13,472	14,616
Not parallel*	1144		
Total	91,330		

*'Not parallel' in this case means that the source–target sentence pairs do not have the same sentiment

4 Experiments

To reiterate, with our supervised approach, we wanted to test the effect of introducing sentiment features in the parallel corpus on the quality of the MT output. For this, we trained two sets of MT models:

(i) $\mathbf{SMT_{Base}}$ and character-level $\mathbf{NMT_{Base}}$, using the general, simple and other non-sentiment-augmented corpus and
(ii) $\mathbf{SMT_{Senti}}$ and character-level $\mathbf{NMT_{Senti}}$, using the general, simple and other sentiment-augmented corpus.

For the semi-supervised NMT model, we trained a character-level NMT system, that took as input the characters of the source and target sentences, and mapped them to two outputs: the characters of the target sentence, and the sentiment of the source sentence. All the models are discussed in detail in the following sections.

4.1 Supervised approach

4.1.1 Statistical machine translation

Moses (Koehn et al. 2007) is a statistical MT system that allows us to automatically train translation models for any language pair, making use of a parallel corpus of translated sentences. Once the model has been trained, an efficient beam search algorithm quickly finds the most probable translation. To train the SMT system, the corpus was tokenized, truecased and cleaned. Afterwards, a Language Model was built using the target side of the parallel data to ensure fluent output. KenLM (Heafield 2011), which comes bundled with the Moses toolkit, was used for building this model. Finally word alignment using GIZA++ (Och and Ney 2004) was performed

which created a phrase table and a translation table. Using both these tables, Moses scores the phrases for a given source sentence and produces the highest-scored phrases as output.

4.2 NMT based on transformer architecture

RNNs typically read one word at a time and perform multiple operations before generating output. However, Bahdanau et al. (2015) show that the bigger the number of steps, the harder it is for the network to learn how to make decisions. Note too that RNNs are sequential in nature, and hence taking advantage of parallel computing offered by state-of-the-art computing devices is a problem.

In contrast, the Transformer model (Vaswani et al. 2017) relies heavily on the instrument of self-attention, thus eliminating the concept of recurrence found in RNN-based architectures. In its absence, a positional encoding is added to the input and outputs to mimic the idea of time-steps in a recurrent network. This positional information thus provides the Transformer network with the order of input and output sequences.

NMT systems based on the Transformer model comprise of two parts, an encoder, and a decoder, where the encoder is composed of uniform layers, each built of two sub-layers, i.e., a multi-head self-attention layer and a position-wise feed-forward network layer. The multi-head attention sub-layer enables the use of multiple attention functions.

Instead of computing a single attention, this stage computes multiple attention blocks over the source, concatenates them, and projects them into space with the initial dimensionality. The feed-forward network sub-layer is a fully connected network used to process the attention sub-layers, by applying two linear transformations on each position and a ReLU activation. The decoder operates similarly, but generates one word at a time, from left to right. The first two steps are similar to the encoder where only the past words are attended to. The third stage is multi-head attention that attends to these past words, in addition to the final representations generated by the encoder. The fourth stage constitutes another position-wise feed-forward network. Finally, a softmax layer allows the mapping of target word scores into target words. The schematic diagram in Fig. 5 shows the architecture of the Transformer model.

Other parameters of the developed system for training are as follows:

- Batch size: 32
- No of encoder and decoder: 2
- Attention heads: 4
- Hidden dimension: 128
- Dropout: 0.05
- Optimizer: adam
- Loss function: sparse categorical cross-entropy
- Embedding dimension: 32

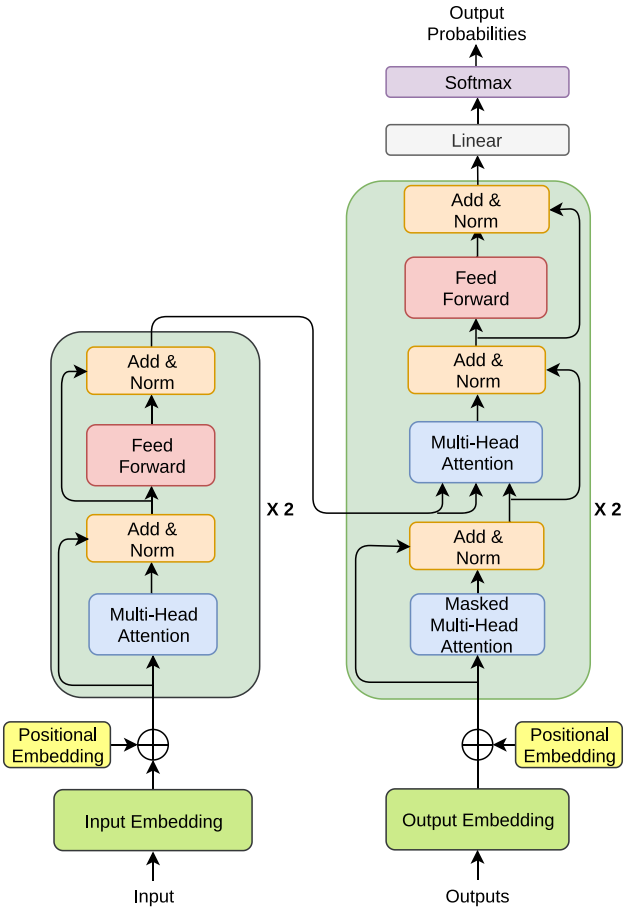

Fig. 5 NMT based on the transformer model

4.3 Semi-supervised approach

Apart from training SMT and NMT systems with parallel sentences obtained from our developed corpora, we wanted to use a semi-supervised approach in which the NMT system learns to mimic the sentiment of the source sentence when trying to generate the output sentence. We name this system Sentiment assisted NMT Architecture (**SeNA**).

The developed system is essentially a character-based NMT (CNMT) system, based on RNNs, where the encoder takes as input the characters of the English sentence. The decoder takes as input the characters of the Bengali language, and maps them to characters of the Bengali sentences, which are offset by one timestep. For the sentiment inference system, we mimic the sentiment analysis system discussed earlier. This system maps the generated sentence of the decoder to a sentiment vector, after passing through two bidirectional LSTMs and a Dense layer. The sentiment

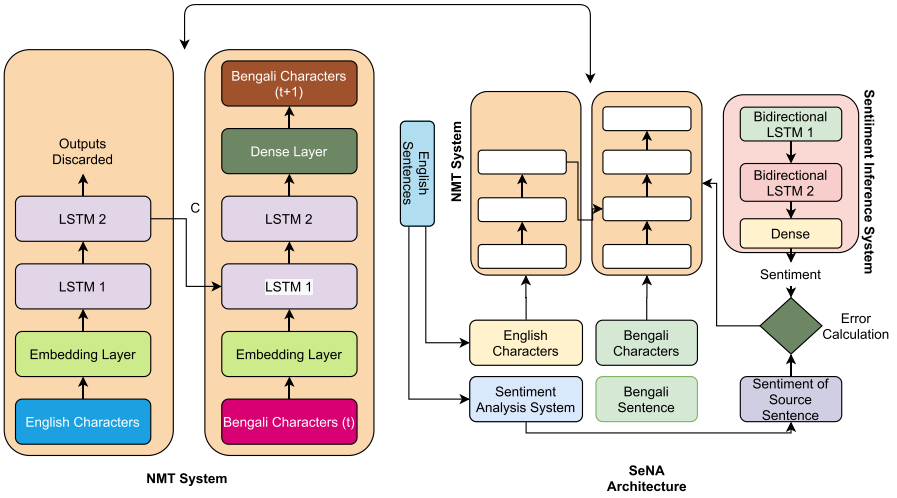

Fig. 6 Schematic diagram of the sentiment-aided NMT architecture (SeNA)

Table 6 Results of automated evaluation performed on the developed translation models

Corpus ⇒	$\mathbf{PC_{Gen}}$		$\mathbf{PC_{Simple}}$		$\mathbf{PC_{Other}}$	
Metric ⇒	BLEU	TER	BLEU	TER	BLEU	TER
$\mathbf{SMT_{Base}}$ ⇒	8.68	89.67	3.48	95.09	6.37	91.48
$\mathbf{NMT_{Base}}$ ⇒	13.22	74.13	6.96 ↑	89.56 ↓	10.11	85.89
$\mathbf{SMT_{Senti}}$ ⇒	8.96	89.55	2.94	98.23	8.06	89.09
$\mathbf{NMT_{Senti}}$ ⇒	13.77	71.61	5.26	91.98	11.01	85.12
SeNA ⇒	13.96 ↑	71.53 ↓	5.42	91.85	10.98 ↑	84.57 ↓

↑ and ↓ shows considerable improvement in BLEU and TER scores, respectively

vector produced is then compared with the sentiment vector of the source sentence, where the source sentence sentiment vector is calculated using the English sentiment analysis system discussed earlier. This system essentially forces the decoder to generate a target sentence that is similar to the source sentence with respect to meaning and sentiment. This system was developed to eliminate the overhead of developing a sentiment-tagged parallel corpus and then training MT systems using the same corpus.

For the SeNA model, the other parameters are as follows:

- Batch size: 64
- Epochs: 100
- Activation: softmax
- Optimizer: rmsprop

Table 7 Results of the manual evaluation performed on the developed translation models, where '1' is the lowest and '5' is the highest level of quality

Corpus ⇒	$\mathbf{PC_{Gen}}$		$\mathbf{PC_{Simple}}$		$\mathbf{PC_{Other}}$	
Metric ⇒	Adeq.	Flue.	Adeq.	Flue.	Adeq.	Flue.
$\mathbf{SMT_{Base}}$ ⇒	1.88	1.86	1.06	1.29	2.40	2.03
$\mathbf{NMT_{Base}}$ ⇒	3.54	3.42	2.28	1.81	3.19	3.14
$\mathbf{SMT_{Senti}}$ ⇒	3.06	2.16	0.63	1.07	2.95	2.72
$\mathbf{NMT_{Senti}}$ ⇒	4.09	3.73	2.02	1.96	3.22	3.25
SeNA ⇒	4.15 ↑	3.81 ↑	2.29 ↑	2.07 ↑	3.42 ↑	3.31 ↑

Scores shown are the average of the scores produced by two linguists

– Loss function: sparse categorical cross-entropy
– Learning rate: 0.001

The architecture of the SeNA model is shown in Fig. 6.

5 Evaluation

In this section, we evaluate the performance of our supervised and semi-supervised systems against two baseline models. We conduct our evaluation of translation quality using the well-established metrics BLEU, TER, Fluency and Adequacy. We subsequently examine the extent to which the sentiment is propagated through the MT pipeline. A test data of 5000 sentences was prepared to test the trained systems.

5.1 Translation quality at semantic level

The quality of the MT output as measured using the automated metrics is shown in Table 6. The translation quality was also judged manually, by two linguists who were native speakers of Bengali. The evaluation criteria were Adequacy and Fluency. Adequacy measures how much of the source meaning is expressed in the target translation. Fluency measures to what extent the translation is well-formed grammatically, is intuitively acceptable and can be sensibly interpreted by a native speaker. The linguists were asked to rate the translation in the range of 1–5, where '1' is the lowest and '5' is the highest level of quality. Table 7 shows the results of the manual evaluation, using the average of the scores from both linguists.

From the results in Tables 6 and 7, we can clearly see that for the corpus $\mathbf{PC^{Senti}_{Gen}}$ and $\mathbf{PC^{Senti}_{Other}}$, where there is a considerable amount of data, both the $\mathbf{SMT_{Senti}}$ and $\mathbf{NMT_{Senti}}$ systems are better compared to the baseline models. Moreover, when compared against each other, $\mathbf{NMT_{Senti}}$ produces better quality translations than $\mathbf{SMT_{Senti}}$.

Table 8 Agreement analysis of sentiment tags of source sentence and translated sentence trained using the $\mathbf{SMT_{Base}}$ system

System: $\mathbf{SMT_{Base}}$			
Corpus: $\mathbf{PC_{Gen}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	980	373	224
Neg.	373	1076	460
Neu.	368	426	720
F-score	0.5527		
System: $\mathbf{SMT_{Base}}$			
Corpus: $\mathbf{PC_{Simple}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	366	698	513
Neg.	699	378	832
Neu.	678	507	329
F-score	0.2154		
System: $\mathbf{SMT_{Base}}$			
Corpus: $\mathbf{PC_{Other}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	940	400	237
Neg.	471	1026	412
Neu.	400	442	672
F-score	0.5257		

For the corpus $\mathbf{PC^{Senti}_{Simple}}$, where the amount of data is less, we see that both SMT systems ($\mathbf{SMT_{Base}}$ or $\mathbf{SMT_{Senti}}$) are better than the NMT models $\mathbf{NMT_{Base}}$ and $\mathbf{NMT_{Senti}}$. NMT produces much more natural and fluent outputs when the amount of training data is high, which is not the case in our experiments.

Finally, we also observe that the **SeNA** system produces better outputs in general, as it takes into account the sentiment of the source sentence and tries to match the same sentiment when predicting the target sentences.

Table 9 Agreement analysis of sentiment tags of source sentence and translated sentence trained using the $\mathbf{NMT_{Base}}$ system

System: $\mathbf{NMT_{Base}}$			
Corpus: $\mathbf{PC_{Gen}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1067	345	165
Neg.	365	1106	438
Neu.	320	390	804
F-score	0.5933		
System: $\mathbf{NMT_{Base}}$			
Corpus: $\mathbf{PC_{Simple}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	348	729	502
Neg.	712	360	837
Neu.	691	511	312
F-score	0.2066		
System: $\mathbf{NMT_{Base}}$			
Corpus: $\mathbf{PC_{Other}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1011	367	199
Neg.	387	1065	427
Neu.	358	424	732
F-score	0.5633		

5.2 Translation quality at sentiment level

We wanted to check whether enriching the parallel corpus with sentiment features leads to the propagation of sentiment through the MT pipeline, e.g. if the polarity of the source sentence is positive, the polarity of the translated system should be positive too. If this statement holds, we can say that the translation is likely to be of good quality which, in turn, should greatly reduce post-editing effort.

For scoring the translations concerning their pragmatic quality, we used the English and Bengali sentiment analysis system, shown in Fig. 4. The source and the translated sentences from all the developed models were tagged with their sentiment, using the developed English and Bengali sentiment analysis system, and agreement analysis was examined. There were five MT models where

Table 10 Agreement analysis of sentiment tags of source sentence and translated sentence trained using the $\mathbf{SMT_{Senti}}$ system

System: $\mathbf{SMT_{Senti}}$			
Corpus: $\mathbf{PC^{Senti}_{Gen}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1078	327	172
Neg.	342	1136	431
Neu.	325	379	810
F-score	0.6035		
System: $\mathbf{SMT_{Senti}}$			
Corpus: $\mathbf{PC^{Senti}_{Simple}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	379	675	523
Neg.	682	391	836
Neu.	651	507	356
F-score	0.2263		
System: $\mathbf{SMT_{Senti}}$			
Corpus: $\mathbf{PC^{Senti}_{Other}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	953	396	228
Neg.	398	1050	447
Neu.	370	447	697
F-score	0.5376		

the baseline systems were trained using three parallel corpora: $\mathbf{PC_{Gen}}$, $\mathbf{PC_{Simple}}$ and $\mathbf{PC_{Other}}$. The agreement analysis of sentiment tags over source and target sentences using these systems are shown in Tables 8 and 9. The MT systems $\mathbf{SMT_{Senti}}$ and $\mathbf{NMT_{Senti}}$ were trained using $\mathbf{PC^{Senti}_{Gen}}$, $\mathbf{PC^{Senti}_{Simple}}$ and $\mathbf{PC^{Senti}_{Other}}$. The agreement analysis of sentiment tags over source and target sentences using these systems is shown in Tables 10 and 11. Finally, the **SeNA** system was trained using $\mathbf{PC_{Gen}}$, $\mathbf{PC_{Simple}}$ and $\mathbf{PC_{Other}}$ parallel data sets. The agreement analysis of sentiment tags over source and target sentences using this system are shown in Table 12. A graph comparing the F-measures of the agreement is shown in Fig. 7.

From Tables 8, 9, 10, 11, 12 and Fig. 7, we observe that, for $\mathbf{PC^{Senti}_{Gen}}$ and $\mathbf{PC^{Senti}_{Other}}$ corpora, the models $\mathbf{SMT_{Senti}}$ and $\mathbf{NMT_{Senti}}$ produce better results than the baseline models, when the sentiment matching of the source–target sentence

Table 11 Agreement analysis of sentiment tags of source sentence and translated sentence trained using the $\mathbf{NMT_{Senti}}$ system

System: $\mathbf{NMT_{Senti}}$			
Corpus: $\mathbf{PC^{Senti}_{Gen}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1132	285	160
Neg.	338	1172	399
Neu.	264	321	929
F-score	0.6466		
System: $\mathbf{NMT_{Senti}}$			
Corpus: $\mathbf{PC^{Senti}_{Simple}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	360	720	497
Neg.	701	376	832
Neu.	688	497	329
F-score	0.2133		
System: $\mathbf{NMT_{Senti}}$			
Corpus: $\mathbf{PC^{Senti}_{Other}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1035	352	190
Neg.	362	1095	452
Neu.	310	397	807
F-score	0.5833		

pair is taken into consideration. Note too that the NMT models outperform the SMT models in the larger training data set-up. As SMT is phrase-based in nature and generally phrases tend to overlap, this reduces the size of the phrase table so the decoder can produce relatively fewer candidate translations. NMT does not suffer from this restriction, so in general produces better output. Moreover, for the the corpus $\mathbf{PC^{Senti}_{Simple}}$, which has a smaller amount of training data, the SMT systems perform better than the NMT systems, when the pragmatics of the source–target sentence pair are considered.

Finally, we see that the **SeNA** architecture performs much better this time, as sentiment was considered as a deciding factor when training the system.

Table 12 Agreement analysis of sentiment tags of source sentence and translated sentence trained using the **SeNA** system

System: **SeNA**			
Corpus: $\mathbf{PC_{Gen}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1176	271	130
Neg.	297	1235	377
Neu.	182	245	1087
F-score	0.7013		
System: **SeNA**			
Corpus: $\mathbf{PC_{Simple}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	367	679	531
Neg.	692	381	836
Neu.	656	511	347
F-score	0.2200		
System: **SeNA**			
Corpus: $\mathbf{PC_{Other}}$			
Source	Target		
	Pos.	Neg.	Neu.
Pos.	1112	289	176
Neg.	339	1150	420
Neu.	255	323	936
F-score	0.6402		

6 Conclusions

In this paper, we discussed how we built a sentiment-augmented parallel corpus. In addition, we separately created two resources, namely a sentiment-augmented parallel corpus with only simple sentences, and another containing more complex sentences. We can see from the automated and manual evaluation that when trained using a sentiment-tagged parallel corpus, the $\mathbf{NMT_{Senti}}$ model performs significantly better in terms of both BLEU and TER when compared to the baseline systems. Moreover, we can see that our **SeNA** model also performs better than the other systems when the sentiment of the source sentence is considered during error calculation when training the NMT system. We also see that the baseline SMT system

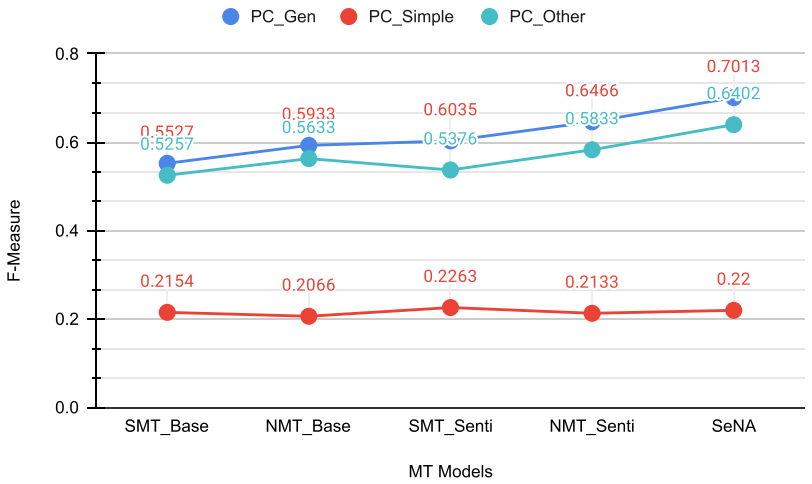

Fig. 7 Comparison of the F-measures when matching the sentiment of the source–target sentences produced by different MT models

performs well in the case of simple sentences, as NMT systems really only work well with huge amounts of data, which was not the case here (i.e., using only 16,654 parallel sentences). Furthermore, when we tested the sentiment matching score of the source–target pair, models using sentiment-tagged data or where sentiment was included performed much better, as seen in the F-Measure graph. Therefore, we can say with confidence that sentiment does play a role in improving MT output quality.

We also hypothesized that if the English sentences belonged to a certain complexity, the Bengali counterparts would automatically belong to the same complexity. While this was done for the English sentences, it was not done for Bengali, as there was no standard lexicon available for POS-tagging and shallow parsing the sentences in this language, so we leave this for future work.

We also saw that although the size of the simple sentence corpus was low, the automatic and manual evaluation results did not show a huge difference when compared to systems that were trained using a much higher number of training sentences. This leads us to believe that if all the sentences in the developed parallel corpus were simple sentences, the results would have been different and the overall quality of the translations would have been higher. As an avenue for future work, we would also like to investigate the fact that a complex compound sentence can be transformed into two or more simple sentences, and how converting the whole corpus to simple sentences only would affect the overall translation quality.

Acknowledgements This work is supported by Media Lab Asia, MeitY, Government of India, under the Visvesvaraya Ph.D. Scheme for Electronics & IT.

References

Afli H, McGuire S, Way A (2017) Sentiment translation for low resourced languages: experiments on irish general election tweets. In: Proceedings of 18th international conference on computational linguistics and intelligent text processing, Budapest, 10 pp

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd international conference on learning representations, ICLR 2015, conference track proceedings, San Diego, CA, 15pp

Banea C, Mihalcea R, Wiebe J, Hassan S (2008) Multilingual subjectivity analysis using machine translation. In: Proceedings of the conference on empirical methods in natural language processing, EMNLP '08. Association for Computational Linguistics, Honolulu, pp 127–135

Das D, Bandyopadhyay S (2010) Developing Bengali Wordnet affect for analyzing emotion. In: International conference on the computer processing of oriental languages, proceedings, Redwood City, CA, pp 35–40

Das D, Bandyopadhyay S (2013) Building language resources for emotion analysis in Bengali. In: Karim M, Kaykobad M, Murshed M (eds) Technical challenges and design issues in Bangla language processing. IGI Global, pp 346–368

Doherty S, O'Brien S, Carl M (2010) Eye tracking as an MT evaluation technique. Mach Transl 24(1):1–13

Fleiss JL, Cohen J (1973) The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educ Psychol Meas 33(3):613–619

Heafield K (2011). KenLM: faster and smaller language model queries. In: Proceedings of the sixth workshop on statistical machine translation. Association for Computational Linguistics, Edinburgh, pp 187–197

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Joshi A, Balamurali A, Bhattacharyya P (2010) A fall-back strategy for sentiment analysis in Hindi: a case study. In: Proceedings of the 8th international conference on natural language processing, Hyderabad, pp 124–130

Kanayama H, Nasukawa T, Watanabe H (2004) Deeper sentiment analysis using machine translation technology. In: COLING 2004: proceedings of the 20th international conference on computational linguistics, Geneva, pp 494–500

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the Association for Computational Linguistics Companion volume proceedings of the demo and Poster sessions, Prague, pp 177–180

Lohar P, Afli H, Way A (2017) Maintaining sentiment polarity in translation of user-generated content. Prague Bull Math Linguist 108:73–84

Lohar P, Afli H, Way A (2018) Balancing translation quality and sentiment preservation. In: Proceedings of the 13th conference of the Association for Machine Translation in the Americas (volume 1: research track). Association for Machine Translation in the Americas, Boston, MA, pp 81–88

Lohar P, Popović M, Way A (Aug. 2019) Building English-to-Serbian machine translation system for IMDb movie reviews. In: Proceedings of the 7th workshop on Balto-Slavic natural language processing. Association for Computational Linguistics, Florence, pp 105–113

Mahata SK, Mandal S, Das D, Bandyopadhyay S (2018) SMT vs NMT: a comparison over Hindi & Bengali simple sentences. In: 15th international conference on natural language processing, proceedings, Patiala, pp 139–147

Och FJ, Ney H (2004) The alignment template approach to statistical machine translation. Comput Linguis 30(4):417–449

Pal S, Patra BG, Das D, Naskar SK, Bandyopadhyay S, van Genabith J (2014) How sentiment analysis can help machine translation. In: Proceedings of the 11th international conference on natural language processing, Goa, pp 89–94

Papineni K, Roukos S, Ward T, Zhu W-J (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, Philadelphia, PA, pp 311–318

Poornima C, Dhanalakshmi V, Anand K, Soman K (2011) Rule based sentence simplification for English to Tamil machine translation system. Int J Comput Appl 25(8):38–42

Resnik P (1998) Parallel strands: a preliminary investigation into mining the web for bilingual text. In: Third conference of the Association for Machine Translation in the Americas. Springer, Langhorne, PA, pp 72–82

Resnik P (1999) Mining the web for bilingual text. In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics, College Park, MD, pp 527–534

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J (2006) A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th conference of the Association for Machine Translation in the Americas: technical papers. Association for Machine Translation in the Americas, Cambridge, MA, pp 223–231

Štajner S, Popovic M (2016) Can text simplification help machine translation? In: Proceedings of the 19th annual conference of the European Association for Machine Translation, Riga, pp 230–242

Tyagi S, Chopra D, Mathur I, Joshi N (2015) Classifier based text simplification for improved machine translation. In: 2015 international conference on advances in computer engineering and applications. IEEE, Ghaziabad, pp 46–50

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, Long Beach, CA, pp 5998–6008

Way A (2018) Quality expectations of machine translation. In: Moorkens J, Castilho S, Gaspari F, Doherty S (eds) Translation quality assessment: from principles to practice. Springer, Cham, pp 159–178