Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

Project report
on
"Nepali Translator"
(For the partial fulfillment of 6th Semester in Computer Science)

Submitted By:
Sharad Duwal (12)
Amir Manandhar (21)
Saurav Maskey (22)
Subash Hada (56)

Submitted To:
Dr Gajendra Sharma
Department of Computer Science and Engineering

Submission Date: 23 July 2019

Certification of Originality

This project is a bona fide work of Sharad Duwal, Amir Manandhar, Saurav Maskey and Subash Hada, who carried it out under my supervision.

Dr Bal Krishna Bal
Project Supervisor

Dr Gajendra Sharma
Project Coordinator

Acknowledgements

We would like to thank Department of Computer Science and Engineering at Kathmandu University for offering COMP 313 under which this work was done. We would like to thank our project supervisor Assoc. Prof. Bal Krishna Bal, whose careful supervision and insight immensely helped us in giving this project the shape it currently has.

Thanks are also due to Language Technology Kendra, by whose courtesy we received a large part of our training data. We would also like to thank Mr. Praveen Acharya for providing us the data that he has collected.

We also thank all the researchers whose work this project stands on. Without their work, this work wouldn't have been possible.

Abstract

Machine Translation (MT) has been a major field of AI almost ever since the first computers were developed. Research on it is still very active because of the wide range of applications it has. While work on MT has never stopped, the introduction of deep learning techniques provided it a much-needed push. Ever since interest in deep learning grew exponentially in the early 2010s, MT has seen a lot of development. Newer and more accurate methods are developed and refined, and more under-resourced language pairs are added to the research roster every year.

While deep learning techniques have been showing a lot of promise, they have also been proven to require huge amounts of data. The Nepali–English language pair—which this project work is on—is under-resourced when it comes to parallel data. Under this project, we try tackling this very problem. We employ different techniques on various noisy comparable corpora in order to obtain a small collection of clean parallel corpus. We then train a neural model on this data, observe the results, and develop comparisons with different benchmarks and discuss them.

Keywords: machine translation, AI, deep learning, data

Contents

List of Figures

List of Tables

Abbreviations

MT	Machine Translation
NLP	Natural Language Processing
NMT	Neural Machine Translation
SMT	Statistical Machine Translation
NNC	Nepali National Corpus
BLEU	Bilingual Evaluation Understudy
SOTA	State-of-the-art
RNN	Recurrent Neural Network
LSTM	Long Short-term Memory
CNN	Convolutional Neural Network
BPE	Byte-Pair Encoding
NE	Nepali
EN	English

Chapter 1

Introduction

Translation of text or speech from one natural language to another with the help of a computer is called machine translation. It is currently one of the major problems in the NLP subfield of Artificial Intelligence. One need not think very hard to see why that is the case. A proper automated translation system (or even semi-automated, like Computer Aided Translation systems) can be very influential in the way human beings communicate. We can already see popular translation services introducing all kinds of features to make language barriers feel as nonexistent as possible. As of 2019, Google Translate provides image and handwriting translation, and Facebook provides automatic translation of all posts and comments on it—services intended to make the act of translation as seamless as possible. If advancements towards this end continue, we cannot rule out the possibilty of a language barrier–less future.

Ever since its earliest mention (Warren Weaver, late-1940s), MT has come a long way. From the simple grammar-based word-substitution models of the 50s and 60s to the statistical methods of the 80s to the current deep-learning approaches, MT has seen a lot of innovation through the years. Just from the amount of research that MT has attracted (and keeps attracting) from all over the world, we can see MT is the future of translation.

Nepali is the official language of Nepal. According to the 2001 census of Nepal, it is spoken by approximately 45 million people in the country. Outside the country, it is spoken in neighboring countries like India, Bhutan and Myanmar (Yadava et al., 2008). Nepali is written in the Devanagari script. Likewise, English is a language spoken by around 2 billion people all around the world (Crystal, 2008). The English language is written in the Latin script.

There hasn't been a great amount of work done towards collection and curation of side-by-side data for the Nepali–English language pair. Until recently that was the case for most language pairs, but now with computational methods and linguistics making new headways, many language communities have begun to catch up. It has, therefore, become important for both languages in this pair (more so for the Nepali language than English) to overcome this lack of corpus collection and analysis. This project began with this premise in mind.

At the time of writing this, Google Translate and Facebook Translation provide Nepali translation services. The translations are not very qualitative, but they are passable. Translate started out as a phrase-based translation system, but after interest in NMT exponentially grew in the early 2010s, it started implementing neural techniques bit by bit until finally it completely replaced the SMT system with an NMT system called Google Neural Machine Translation (GNMT) (Wu et al., 2016). After this overhaul, translations for high-resource languages (especially the Germanic languages) have seemed to increase in quality, but for a low-resource language like Nepali, there is still a lot of room for improvement. A big reason for this disparity is the unavailability of qualitative corpus for the low-resource languages.

For a low-resource language pair, MT systems should make use of any kind of resource available, namely noisy comparable corpora and monolingual data (Guzmán et al., 2019). Even with the kind of internet penetration that Nepal has been seeing of late, there hasn't been any large-scale aggregation of parallel data[1] as a direct consequence of it. The only significant parallel corpora available online for the Nepali–English pair are the translations of different Linux distributions and translations of Bible excerpts.

[1] In natural language translation problems, parallel data (or corpus) for a pair of languages is a collection of two sets of data (one for each language in the pair) in which text from the source language and the respective translation in the target language are aligned in sentence-level or paragraph-level or document-level.

The collection of Nepali monolingual data, though, is considerable. However, since unsupervised learning techniques, which make use of monolingual data, have been shown to not work well with this pair (Guzmán et al., 2019), parallel data is the only hope until research toward unsupervised learning in low-resources languages advances.

A major work in the direction of corpus collection was done by Yadava et al. (2008) that culminated in the Nepali National Corpus (NNC). NNC comprises of monolingual as well as parallel corpus. This parallel corpus, however, is aligned in the document level. This is not favorable for several MT architectures currently used (Vaswani et al., 2017). A broader discussion on suitability of corpus will come later in this report.

MT has had about 70 years of history. The rule-based grammar systems for MT were mainstream translation services back when computational cost was still a bottleneck. With Moore's law coming into action and computational power increasing in leaps, statistical methods became popular. SMT methods were better than rule-based systems because they could make rules less strict. Language translation is hardly a rule-based problem and SMT showed that it was possible to use ideas of probability to ease up on the rules. With computational power increasing even more in the 2010s, all of ML took a turn. In MT, Kalchbrenner and Blunsom (2013), Sutskever et al. (2014) and Cho et al. (2014) proposed the idea of using neural networks for machine translation. Unlike the phrase-based system that was popular before that, neural machine translation involved training a single large neural network (as opposed to subsystems used in phrase-based systems) (Bahdanau et al., 2015).

For this project we collect parallel data from various sources and use an NMT architecture known as Transformer (Vaswani et al., 2017) to train on the data. Transformer is a self-attention NMT model that Vaswani et al. (2017) have shown to be better than recurrent methods that are popular for sequence-to-sequence learning. We use the popular BLEU metric for evaluation purposes. The algorithm checks the correspondence between the correct translations and the model's translations. There is another metric called METEOR Indic, more suited for Nepali, but we haven't used it here since much of the benchmarks we evaluate against are in BLEU.

In this report, we show how we collect a small parallel corpus from several noisy comparative corpora available to us and what results we obtain after training and testing. In Chapter 2, we discuss some related works, their results and how we set benchmarks to evaluate our work. In Chapter 3, we describe the sources of the data we collect, the cleaning methodologies we use, and the training settings. Additionally, we describe how we scale the existing data-cleaning techniques and use it on the Nepali–English pair. We also discuss the various tools we ourselves come up with to facilitate the cleaning. Chapter 4 is a discussion on the results we obtain, which we compare and contrast with our baselines in Chapter 5.

Objectives and Significance. Most researches in low-resource language pairs are simulated by using smaller sets of parallel data from high-resource language pairs. For example, a common way is to sample small sizes of parallel data from language pairs like German–English and French–English and use such samples to simulate a low-resource environment. While such methods probably have their own merits, working in an actual low-resource environment is more practical. Simulations are not always able to capture the reality of low-resource language pairs (Neubig and Hu, 2018; Guzmán et al., 2019). The Nepali–English pair, as we have discussed, is a real low-resource environment. We, therefore, hope our results will further research on the pair and on low-resource MT.

Chapter 2

Literature Review

2.1 Neural Machine Translation

NMT systems generally have an encoder-decoder architecture. They are different from SMT systems in their methods and their build. While SMT systems are made up of various smaller components that are each trained separately, an NMT system is trained one-piece. In NMT, a single large neural network reads an input sequence and outputs its translation (Bahdanau et al., 2015). Sutskever et al. (2014) used an NMT implementation based on RNN with LSTM units and achieved the SOTA SMT performance in 2014. Similar results were reported by Cho et al. (2014). Additionally, incorporation of NMT techniques into conventional phrase-based systems allowed the new systems to surpass the SOTA performance. Such results expedited work on NMT.

Similarly, Google Translate implemented a full NMT model in 2016. The details of this method are in (Wu et al., 2016). In the paper, Wu et al. (2016) conclude that a CNN-based NMT technique reduced translation errors by 60% as compared to the phrase-based prediction system that Google had been using for a long time. This clearly shows the advantage of a neural method when compared to a statistical method.

Almost all current SOTA techniques and scores in MT are from neural methods.

Recurrent Neural Networks (RNNs) were the default choice for NMT because they are ideal for sequence-to-sequence learning. In the RNN NMT implementation, a bidirectional RNN called an encoder is used to encode the input sentence and then a decoder RNN is used to translate the sentence into the target language. RNNs implement hidden layers that retain the meaning of an input vector as it encodes it token by token. Then the final hidden layer is used in the decoder side as well to output the target vector, which gives the translation. The other popular neural network, CNN, is actually better for long continuous sequences, but due to architectural weaknesses they weren't used in MT for some time (Bahdanau et al., 2015). (Unlike RNNs, CNNs do not depend on computations of previous steps.) Recent work on attention-based CNNs have shown better results, however.

Bahdanau et al. (2015) could very well be the first to use attention techniques. Instead of retaining meaning of a whole sentence (as was done in RNNs with LSTM units) in a single hidden state and then deriving each output word using the meaning (of the *whole* sentence) so retained, attention mechanisms only use the most relevant input information to output a word. This relieves the encoder from having to encode all the information from an input sequence and also makes the decoder more accurate since only relevant information is used while decoding.

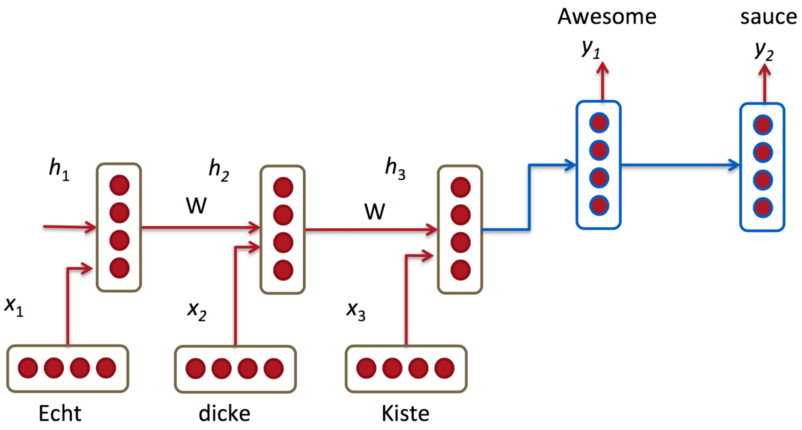


Figure 2.1: Recurrent Neural Network in MT. Encoder (Red) and Decoder (Blue) with input vectors x_i and output vectors y_i, weight W and hidden layers h_i. [Image source: http://cs224d.stanford.edu/lectures/CS224d-Lecture8.pdf]

2.2 Transformer

Building on top of attention mechanisms, a new NMT architecture called Transformer was recently introduced by Vaswani et al. (2017). While it still maintains the encoder-decoder architecture, everything else about it is different from an RNN. A major difference is the input-output pair for these models. While RNN is a recurrent mechanism that takes a whole sequence, develops hidden layers and uses a decoder in tandem with the hidden layers to generate a sentence, all done in a single iteration, the Transformer is not a recurrent technique. It generates translations one word per iteration.

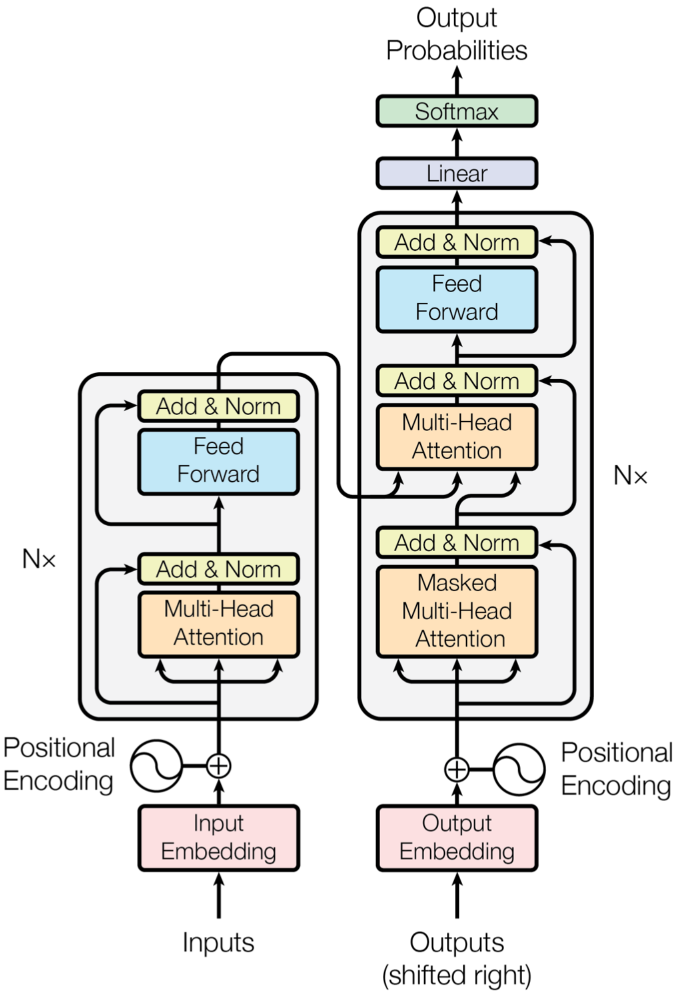


Figure 2.2: The Transformer-model architecture [Source: (Vaswani et al., 2017)]

In Figure 2.2, the left half is the encoder and the right half is the decoder. After inputs and outputs are fed into the model, embeddings are generated. Since Transformer is not a recurrent architecture and translations are produced one word at a time it can be difficult to know the right order of the words. Positional encoding uses trigonometric functions (discussed in §3.5 in (Vaswani et al., 2017)) to generate positions of each token.

Next are the encoder and decoder blocks. These blocks perform multihead attention on their respective inputs. Three important things come out of them: values and keys from the encoder, and queries from the masked multihead attention on the decoder. The multihead attention on the encoder side returns a tensor called values (**V**) of the most interesting features

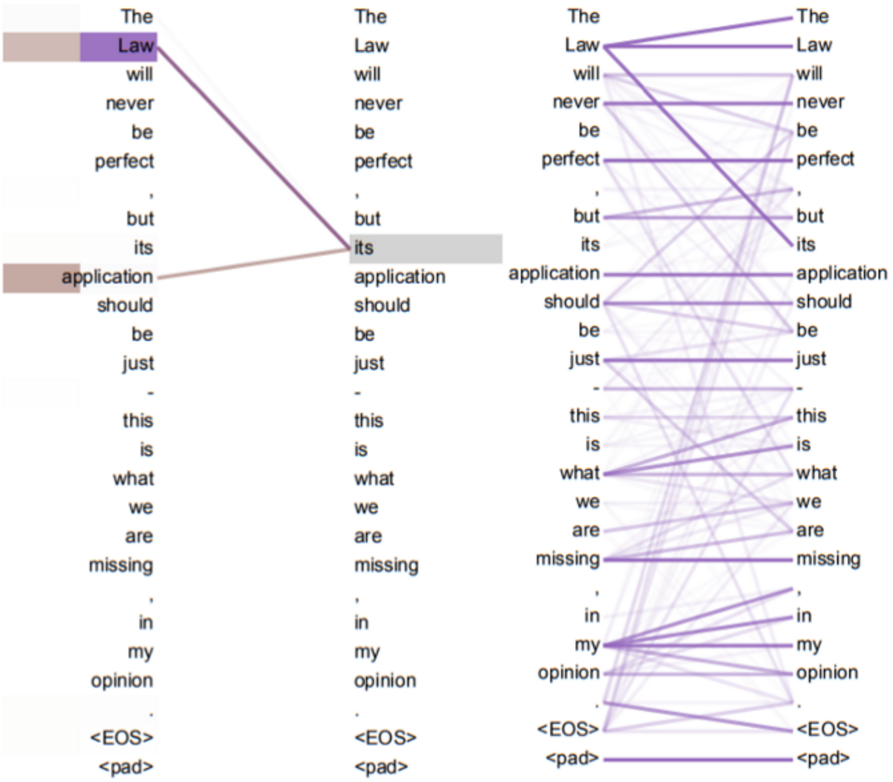


Figure 2.3: Anaphora resolution by Transformer. The Transformer model relating different words to other words in the same sentence, trying to extract context of each word. Darker the color stronger the correspondence. [Source: (Vaswani et al., 2017)]

from the source sentence. Keys ($\mathbf{K}$) is also a tensor and it contains information regarding the positions of the entries in $\mathbf{V}$. (Again, position of tokens is important in Transformer because it is not recurrent.) Queries ($\mathbf{Q}$) is the output of the masked multihead attention.

The dot-product of $\mathbf{K}$ and $\mathbf{Q}$ returns positional information for the target. After normalization and a softmax on the dot-product, it is multiplied with $\mathbf{V}$. These operations happen concurrently in many layers and finally the word probabilities are output. The matrix of outputs is given by:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

The complexity per layer of the Transformer model is $O(n^2 \cdot d)$ where n is the length of the sequence and d is the representation dimension. For RNNs, the complexity is $O(n \cdot d^2)$. A *restricted* Transformer model has complexity of $O(r \cdot n \cdot d)$ where r is the size of neighborhood (Vaswani et al., 2017).

Transformer models are computationally cheaper and quicker to train. They can handle longer-range dependencies than most other translation models. Upon release, it had surpassed BLEU benchmarks of SOTA methods in English–German and English–French pairs by 1-2 points. For these reasons we thought it a good fit for this project.

2.3 Nepali–English

The first machine translation project for the Nepali-English pair was called Dobhase. Dobhase was a rule-based system that accepted an input string, parsed it, generated the syntax for the target language and output the translation. It was unable to handle sentences of complex structures (multiple conjunctions, ambiguous words, etc) and has thus been discontinued (Acharya and Bal, 2018).

NMT techniques require large amounts of parallel data (Tiedemann, 2012). This is problematic for many low-resource pairs, but once enough data has been collected, it works very well. It can be concluded from several recent works on NMT that with sufficient parallel training resources a neural MT can achieve near human-level translation performance (eg., Chinese-English)

(Guzmán et al., 2019). The Nepali-English language pair, however, doesn't have a large amount of parallel data.

NMT is relatively new for the Nepali–English pair. In 2018, Acharya and Bal (2018) used a small portion of the parallel corpus from Nepali National Corpus (NNC) collected by Yadava et al. (2008). They collected a total of 6535 sentence pairs and divided them into a train set, a development set and a testing set. They applied SMT and NMT techniques. On their test sets, the highest BLEU scores they obtained were 5.27 and 3.28 in SMT and NMT respectively. They do not explicitly state the translation direction, but from the sample translations they provide, we assume it is the English to Nepali direction.

More recently, Guzmán et al. (2019) consolidated parallel data from the translations of different Linux distributions, from Bible translations and from other sources. They collected a total of about 600k noisy[2] parallel sentence pairs. They used fully supervised, semi-supervised and fully unsupervised methods on this parallel corpus. In their paper, they point out that apart from obtaining a qualitative parallel corpus, another big problem in low-resource MT is that of evaluation. Without proper benchmarks, new work will be hard to evaluate. They offer a solution for this in their work.

They provide various evaluation sets for the Nepali–English and the Sinhala–English pairs. A comprehensive discussion on the process of collecting the evaluation sets can be found in their paper. For the Nepali–English pair, they end results are three evaluation sets: dev, devtest and test with 2559, 2835 and 2924 sentences pairs respectively.

On their *devtest* set, in the English to Nepali direction, the supervised method they employed yielded a BLEU score of 4.3, semi-supervised method yielded a BLEU score of 6.8 and unsupervised method yielded 0.0. In the Nepali to English direction, the supervised, semi-supervised and unsupervised methods gave BLEU scores of 7.6, 15.1 and 0.1 respectively.

We set the scores and methods used by Guzmán et al. (2019) and Acharya and Bal (2018) as our baselines.

[2] We consider repeated sentences, missing sentences and misaligned sentences to be noise.

2.4 Miscellaneous

2.4.1 English–Hindi

Hindi is very similar to Nepali. The Nepali language does not only share the script with Hindi – they are very similar spoken as well, with many words inflecting into one another. Both languages are morphologically rich, which makes translation into and from other languages relatively harder than for languages of moderate morphological complexity (like English, German, etc) (Ataman and Federico, 2018; Kunchukuttan et al., 2018; Guzmán et al., 2019).

Owing to the similarity of Nepali and Hindi, research and results in one language can be related with slight reservations to the other language. We therefore also looked at related works and baselines for the Hindi–English pair.

In their 2018 work Kunchukuttan et al. (2018) collected 1.49 million parallel segments for the English–Hindi pair, making it the largest publicly available parallel corpus for the pair. In addition to parallel data available at OPUS (Tiedemann, 2012), they used judicial domain corpus and Indian Goverment corpora. For a noisy corpus called Gyaan-Nidhi which was originally in HTML and was not sentence-aligned, they used the sentence alignment technique proposed by Moore (2002) to align the noisy sentence pairs. They divided their final corpus into sets train, test and dev with 1492827, 2507 and 520 sentence pairs respectively.

They trained an SMT and an NMT system on the data. The highest BLEU scores for NMT were 12.23 and 12.83 for English to Hindi and Hindi to English directions respectively. The same for SMT in the same order were 11.75 and 14.49.

2.4.2 Sentence-level alignment

Parallel corpora have shown promise in MT ever since the SMT days. It can probably be considered as the most valuable training resource in natural language translation purposes.

Parallel data that is aligned in the word token–level is a set of dictionaries. Other levels of alignment are sentence-level, paragraph-level and document-level. In the word token–level alignment, there is no context. The sentence-level alignment has minimal context. Paragraph-level alignment has some context and a document-level alignment has a lot of context. Which alignment level is the most preferable and what amount of context is needed in a translation task?

MT techniques still haven't been able to make great use of textual context. The SOTA NMT methods can parse the context in simple straightforward sentences, but as the sentences get longer the parsing task gets more complex—in terms of computation as well as semantics. The algorithmic complexity per layer of the Transformer model, as we have discussed in §2.2, increases quadratically with the length of the sequence. Even if the complexity increased linearly, associating words and getting context will still be easier for shorter sentences. Longer texts that are paragraph-level or document-level aligned will be difficult to process. Also, since translation is not a metric precision–based task, there can be multiple ways of correctly translating a single sentence. A five-sentence paragraph can be translated into a ten-sentence paragraph in another language. The meanings would be the same, but a machine will have lesser accuracy extracting word-level associations in this scenario. This gets worse in the document-level. The preferable alignment, therefore, is sentence-level.

Chapter 3

Methods

3.1 Corpus

Discussions on the various sources that contributed to our parallel corpus are as follows:

Bible

A Bible translation[3] contributes about 31000 sentence pairs to our corpus. There are actually two different English versions and a single Nepali translation, so we could use a total of 62k sentence pairs. One English version was in archaic writing and the other was in more or less modern English. Training the model with and without the archaic version didn't affect our results very much, so we chose not to use it.

Linux distribution translations (LDT)

The Linux operating system has various distributions like KDE, Arch, Gnome, etc. Users are allowed to offer to translate the OS into other languages. There are Nepali translations as well for KDE, Gnome and Ubuntu distributions. This data was collected as a parallel corpus by Tiedemann (2012) under project OPUS. The translations to the three distributions amount to about 495k sentence pairs, but this is very noisy. While the sentences are mostly

[3] `https://github.com/christos-c/bible-corpus`

properly aligned, repetitions are rife. There are as many as 20 repetitions for a single sentence. Thus, the number of parallel sentences post cleaning went down to about 59k pairs.

Penn Treebank Corpus (PTC)

Nepali corpora parallel to 88k words of common English source from PENN Treebank corpus. There are a total of 4k sentences. Some sentences are mis-aligned by a sentence or two.

Global Voices 2018 (GV18)

Contributed more than 3000 sentence pairs. The sentences are extracted from an XML file. Lines run long, with multiple sentences in the same line with only ocassional alignment. Most sentences are mis-aligned by a few lines.

Corpus collected by Acharya and Bal (2018) (PB18)

This was about 6500 sentences. Upon close examination, parts of Penn Treebank Corpus was found as well, so we removed repeated sentence pairs.

NNC Parallel Corpus (NPC)

By courtesy of Language Technology Kendra, we obtained National Development Plan texts which were a part of the NNC Parallel Corpus collected by Yadava et al. (2008). The corpus was aligned in the document level, so it was necessary to first bring it down to sentence-level alignment. We obtained around 13k sentence pairs. The alignment isn't perfect. A manual examination of 50 sentence pairs showed 75% accuracy in alignment.

Automatic Translation of Monolingual Data (AMono)

Due to the dearth of parallel data and as an experiment on its own, we also extracted some sections of the NNC monolingual corpus and obtained automatic translations using Google Translate. We used the source sentences and the translations so generated as parallel corpus. This contributes to around 37k sentences.

3.2 Cleaning

Different methodologies that were put to use in cleaning the corpus are discussed below:

3.2.1 Manual cleaning

Manual cleaning is the most time-consuming method of cleaning noisy parallel corpus, but it guarantees quality. About 4500 sentence pairs were manually cleaned—this involves reading lines and making sure they align on the sentence level and if they don't, making changes so that they do. "Changes" involve correction and deletion. *Applied to:* GV18, parts of NPC, parts of AMono.

3.2.2 Automated cleaning

Automated cleaning would involve computational methods used to clean the data. Following is a list of several types of noise encountered in the collected corpus and brief discussions on the methods used to attempt to eliminate them[4].

3.2.2.1 Repeated sentence pairs

Many instances of repeated sentence pairs were found. The LDT corpora had the most repetitions. Similarly, the PTC was repeated in PB18. Repetition of sentences is not a big problem if dealing with a single file, but since we are dealing with sentence pairs we have to remove repetitions from both the files in a way that the rest of the file is still properly aligned.

A feasible solution is using the `set` type in Python over the `zip()` of `lists` of sentences. *Applied to:* LDT (brought down 600k pairs to less than

[4] All the functions discussed below have been implemented in `functions.py` file.

120k pairs), also applied to the rest of the corpus to make sure there are no repeated pairs.

3.2.2.2 Wrong translations

This describes the situation when a line doesn't at all match with its "translation" in the parallel file even when the parallel pairs above and below that line are properly aligned. This was another major problem in the LDT corpus. A low-cost check based on the sentence-length was devised in order to eliminate this problem.

Algorithm 1 Length Similarity

```
procedure LengthSimilarity(sent1, sent2)
    $a \leftarrow$ number of words in sent1
    $b \leftarrow$ number of words in sent2
    if $a > b$ then
        $s \leftarrow (b/a)$
    else
        $s \leftarrow (a/b)$
    return $s + ((1 - s)/(1 + abs(a - b)))$
```

The function LengthSimilarity takes two sentences and always[5] returns a value between 0 and 1. Higher the value, greater the length similarity. We set a cut-off of 0.53 for a sentence pair. If they score lower, they are discarded.

Additionally, surface observation of the LDT corpus showed that shorter phrases were often mistranslated, so we set a cut-off here as well: a sentence must be 4 words or longer. *Applied to:* LDT (brought down 120k pairs to less than 60k pairs), not applied to other files because they did not have this problem[6].

[5] Based on several tests. Mathematical verifications hasn't been done.

[6] Most conclusions like this, which scale throughout a large corpus, are backed by surface observations only.

3.2.2.3 Document-level parallel corpus

The parallel corpus within NNC is in document-level and we have discussed in §2.4.2 why sentence-level alignment is more preferable for machine translation with the SOTA methods. We follow the algorithm below to bring the document-level aligned corpus to sentence-level:

1. Use delimiters (fullstops [.] for English and *purnabiraam*s [।] for Nepali) to convert every line—not sentence; every *line* in a file—into sentences or, in one word, *sentensify.*

2. The previous step will *sentensify* Nepali well since in Nepali *purnabiraam*s are only used to end a sentence, but the period symbol has several uses in English, for example, in salutations, acronyms, Latin borrowings, etc.: *Mr., Mrs., Dr., Rs., 1.1.2, a., etc.* and so on. We use regular expressions to fix as much of it as possible[7].

   (a) **Joining an incomplete line to the line above:** We look for lines that start with a small letter. This accounts for every split that would have been caused due to salutations or acronyms or any word that ends in a period.

   (b) **Joining a numeral line with the next line:** We look for lines that start with a numeral and are followed by only whitespace. We join such lines with the next line, which, hopefully, was cut off at a period that followed the numeral. For example: a sentence `1. The Sixth Plan ...` broken into two sentences: `1` and `The Sixth Plan ...` because of the period after the numeral.

3. Manual checking of random sentence pairs can be done to see if anything else is correctable.

4. Use Bleualign described by Sennrich and Volk (2011) to align the sentences. Bleualign is a MT-based sentence-alignment algorithm. It requires a source file, the misaligned target file and an automatic translation of the source file. The main disadvantage of an algorithm like

[7] This has been implemented in `bad_parallel_fixes.py`

Bleualign is that it requires a language model already, which means it is not attractive for low-resource language pairs (Sennrich and Volk, 2011). Trying to align sentences to collect good data using a bad language model doesn't generate good results. A brief overview of how Bleualign works (Sennrich and Volk, 2011):

(a) Sentence-align the parallel training corpus:
   i. In its first iteration it matches the lengths of the different lines using Gale & Church algorithm or Moore (2002) algorithm.
   ii. In all other iterations:
      A. Automatically translate the corpus using an SMT system trained in the last iteration.
      B. Align the text using Bleualign and this translation

(b) Train an SMT system on the sentence-aligned corpus.

A major problem with sentence-alignment algorithms is related to what's called beads. A 1-to-1 bead is a collection of one sentence in the source file mapping to one sentence in the target file. An n-to-m bead would be the collection of n sentences in the source file that translate into m sentences in the target file. As we can guess, 1-to-1 beads are easy to align. Bigger beads are harder to align. Sennrich and Volk (2011) report a 69.5% alignment quality in lax settings and 94.4% alignment quality in strict settings with Bleualign.

Applied to: NPC (extracted about 13k sentence pairs from four National Development Plan documents).

3.2.2.4 Extracting Monolingual data

The NNC monolingual corpus has been tagged by Hardie et al. (2005) using the Nelralec tagset. Every monolingual document in the NNC corpus is an XML file and every token word in these XML files is wrapped by tags describing the type of that word (NN for common noun, PMX for first person pronoun, etc.). We devised a method for extracting only the text from these files and we assume it will be helpful for anyone working with the NNC monolingual corpus, so we brief about the method[8]:

1. Read the XML file, parse it, and find all `<s>`[9] tags in it.

2. Just removing the tags to obtain the words within them and then joining the words to get the text works fine. But there are connectives like postpositions and numeral classifiers (all listed and described by Hardie et al. (2005)) that are, in the Nepali language, connected with origin nouns without spaces. A noun connected with a postposition and one without give different meanings and, since such subtleties are important in translation, they need to be written as the same word. For example, क्षेत्रबाट is formed using individual words क्षेत्र and बाट, which give the collective meaning only when they are joined. We achieve this by adding some conditions.

3. Go through the `<w>` tags one by one and use regular expressions to check whether the value of the attribute `ctag` starts with `I` or `M` (accounts for postpositions, number and numeral classifiers). Also check that it doesn't have `M` as the second character (this removes tags that mistakenly get caught). Implement a stack for these word types and then go through the lines to extract the words in proper form and order.

This provides a good enough extraction of monolingual data from the NNC corpus. Interested readers can look up (Hardie et al., 2005) and read about the different tags.

3.3 Training

We use the Transformer model described in §2.2 to create a fully supervised NMT system.

We combine the various cleaned corpus files into two files, one for Nepali sentences and the other for English sentences. From the corpora we have

[8] Implemented as function `xml_to_text()` in `functions.py` file.

[9] This is the sentence tag. The word types (NN, PMX, etc.) are the values of an attribute `ctag` in the `<w>` tags.

collected and cleaned, we have about 150k sentence pairs (including 37k sentence pairs from AMono data). A step-by-step discussion on the training procedure is as follows:

1. **Text Normalization:** We use IndicNLP[10] library to normalize and tokenize the Nepali sentences. It is a text-processing library for the Indic languages like Hindi, Nepali, etc. There are two ways to represent a Nepali character with *nukta*: we either consider the character and its nukta to be seperate unicode characters or we consider the composite character. We use the former representation since often the *nukta* has its own meaning and gives context to the character. We do not normalize the English sentences in any way.

2. **Tokenization:** For Nepali, we use the Nepali tokenizer in the IndicNLP library. For English, we use Sacremoses.

3. **NMT setup:** We use the Transformer model in the Fairseq toolkit[11] for training.

4. **Vocabulary:** We use BPE[12] to learn the vocabulary of the source and target files. Translation is not a fixed-vocabulary problem and we cannot feed all possible words in a language into a model. However exhaustively we feed dictionaries into a model, there will still be out-of-vocabulary words. Sennrich et al. (2016b) describe how every word is actually a collection of "subwords"—characters that combine to form a word. Instead of feeding actual words into a model, we can therefore feed these subwords using the BPE algorithm, which breaks data into constituent parts. We use a vocabulary of 20000 subwords for Nepali to English translation and that of 5000 subwords for English to Nepali direction[13]. Also, we train separate BPE models for English and Nepali since the do not share alphabet and writing system. We use the sentencepiece[14] library to learn the BPE over the languages.

[10] `https://anoopkunchukuttan.github.io/indic_nlp_library`

[11] `https://github.com/pytorch/fairseq`

[12] Byte-Pair Encoding. Originally a data compression technique.

[13] We arrive at these numbers after running some experiments.

[14] `https://github.com/google/sentencepiece`

5. **Model architecture:** We use a Transformer model with 5 encoder and 5 decoder layers. We use 2 attention heads. The number of embedding dimension and inner-layer dimension are 512 and 2048 respectively.

6. **Training details:** We use a batch size of 128 sentences and use Adam optimizer (Kingma and Ba, 2014). We remove sentences with number of BPE tokens less than 1 and those with more than 250. We train every model to a maximum of 100 epochs.

Chapter 4

Results

4.1 Final scores

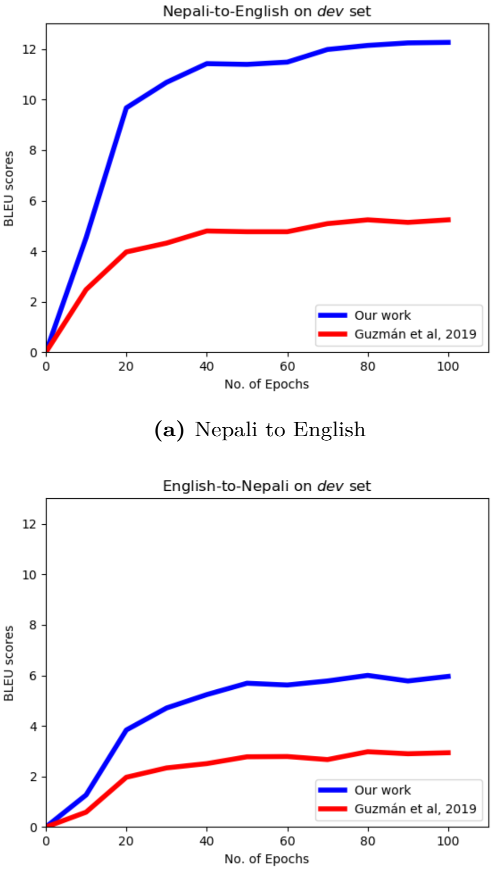
We used the evaluation sets made available by Guzmán et al. (2019) to evaluate our system. Since their *test* set is not available until August 2019, we evaluate BLEU scores on their *dev* set, which has 2559 sentence pairs. We reproduced their supervised system and evaluated it on the *dev* set as well.

On the *dev* set, the BLEU score we obtained from our system in the Nepali to English direction is 12.26 while in the English to Nepali direction, it is 6.0. The supervised system of Guzmán et al. (2019) scored 5.24 BLEU (−2.36 from the score on the *devtest* set) on the *dev* set in the Nepali to English direction. In the English to Nepali direction, it scored 2.98 (−1.32 from the score on the *devtest* set). We trained both these systems for 100 epochs.

	Corpus size (in sentence pairs)	**NE-EN**	**EN-NE**
Guzmán et al. (2019)	564k	5.24	2.98
This work	150k	12.26	6.0

Table 4.1: BLEU scores on *dev* set.

These values need to be taken in with some considerations. The major concentration of Guzmán et al. (2019) was *not* to obtain good BLEU scores. They did not even clean the corpus they collected. Their main aim was to develop evaluation sets for two low-resource language pairs, namely Nepali-English and Sinhala-English, which they did. In addition to building these sets, they only provided benchmark scores on the low-resource pairs using the noisy corpora and the evaluation sets they came up with. So our BLEU scores—the main concentration of our work—and their scores are better compared with some awareness of our varying motivations.



(a) Nepali to English

(b) English to Nepali

Figure 4.1: BLEU scores

However, it can be concluded from these figures that just having a larger corpora doesn't guarantee better results. Much of the parallel corpus we use is similar to what Guzmán et al. (2019) used. Cleaning much of it and adding some more sentences reduced the corpus size and also reduced training time and the model size. It still yielded better results (almost double on both translation directions). A more comprehensive comparison is in the next chapter.

We do not include the scores of Acharya and Bal (2018) in Table 4.1 because they evaluated their scores on completely different sets. With their NMT system, the highest BLEU score they had obtained was 3.28. This was on their TestSet1. On their TestSet2, their NMT had obtained a BLEU score of 1.73. They had used an NMT system comprising of an RNN with LSTM units and had used a much smaller parallel corpus.

From Figures 4.1 we can see that BLEU scores improve steeply within the first 20 or 30 epochs and then plateau.

A selection of translations output by the system is in the Appendix A.

Chapter 5

Discussion

In this chapter, we will discuss the results and the factors that influenced the decisions we made pertaining to the corpus and the training procedure.

5.1 Comparisons with (Guzmán et al., 2019)

The work carried out by Guzmán et al. (2019) was very important and timely for our own work. Also, there hasn't really been a lot of work in the Nepali language in the NMT department, so it is only obvious for us to compare and contrast our work with theirs again and again. There are a few things we did differently from them and we will talk about them here.

5.1.1 Repetitions

A big difference in their corpus and ours is the amount of repetition. Much of LDT is very repetitive, as we have mentioned, and LDT is a large part of their corpus. We eliminated all repeated sentence pairs.

At one point, because the corpus was too small, we tried training the model by repeating data that we felt was qualitative assuming it would only improve the model, if anything. We assumed that the model wouldn't know that it has already seen a sentence pair and if anything it will just absorb newer information than the last time it parsed the sentence pair.

This didn't improve the model so much, however, and we removed the repetitions.

Another important issue is the alternative version of the Bible. There are two version of the English Bible: one archaic and one modern. It was possible to use both the versions and have 62k sentence pairs from just the Bible translations, which Guzmán et al. (2019) did. We trained our system with and without the archaic version and came to conclude it didn't increase the scores, so we removed the archaic version. We removed it on the assumption that differing translations of the same sentence appends conflicting information into the model.

5.1.2 Cleaning

The cleaning we did on the corpus definitely had a large part in getting the increase of over 7 BLEU points. The corpus we use and the one Guzmán et al. (2019) used have an intersection of over 100k sentence pairs, which is about two-thirds of our corpus. Just the new sentence pairs in our corpus (much of which is not even perfectly clean) could not have contributed to such a significant increase.

We can observe from this that noisy translations affect the model significantly.

5.2 German–English

We used a Transformer model not very different from the one used here and trained it on 120k parallel sentences from the English-German language pair provided under IWSLT 2016 (Cettolo et al., 2012). By the 20th epoch, the BLEU score on *tst2014* test set was already 22+. We can observe from only this result that morphologically rich languages like Nepali are hard to translate to and from.

5.3 Ne-En *v* En-Ne

The BLEU score for translation in the Nepali to English direction is higher than that in the reverse direction. Guzmán et al. (2019) had similar results and they claimed this is “not surprising” since translating into morphologically rich languages is inherently more difficult. We are still running tests to exactly determine the reason behind this, but morphology is without doubt a big reason.

Chapter 6

Conclusion and Future Work

In this report, we discussed one way of collecting a small corpus from several noisy comparable corpora, of using it to build an NMT system, and finally analysing the results for a low-resource language pair. Chapter 1 introduced the topic of machine translation, the emergence of deep learning techniques and obtaining good results using it, Chapter 2 discussed related works, a brief history of the various techniques that have been used in NMT, Chapter 3 briefed about the methods employed, and Chapter 4 and 5 presented and discussed the results.

Even if we obtain better results than the benchmarks, our experiments show that current SOTA approaches in NMT are rather poor in handling this language pair, especially in lack of parallel data (comparing to the result for the German–English pair). We hope that this research work will help the Nepali–English MT community make faster developments. Nepali and English are languages with very different grammar rules and morphology and this makes it a perfect pair for research on low resource MT.

With internet penetration increasing in Nepal and all levels of Nepali users contributing to the ever-growing datastore that is the Internet, we can rest assured that Nepali language will make its presence felt even more in due time. A more significant presence would mean more data which can then be used in progressing research on translation to and from the Nepali language. In this work, we release the data we have collected and cleaned and we will also release the different scripts/techniques we have written to clean the data.

We believe a bigger parallel corpus for the Nepali-English language pair can benefit the pair very much. Work on unsupervised and semi-supervised NMT methods should also benefit it. Guzmán et al. (2019) reported a BLEU score of 12.7 for the Nepali to English direction using a semi-supervised technique in which monolingual data was also used in addition to the parallel data. They used back-translation techniques described by Sennrich et al. (2016a) and used a much larger Transformer model. Given the relative availability of monolingual data for the Nepali language, a semisupervised method using back-translation on this work should also improve the BLEU scores in both the translation directions.

Bibliography

Acharya, P. and Bal, B. K. (2018). A comparative study of SMT and NMT: Case study of English-Nepali language pair. *The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages*, pages 90–93.

Ataman, D. and Federico, M. (2018). An evaluation of two vocabulary reduction methods for neural machine translation. *Proceedings of AMTA 2018, vol. 1: MT Research Track.*

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *ICLR 2015.*

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. *Proc. of EAMT*, pages 261–268.

Cho, K., Merrienboer, B. v., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259v2.*

Crystal, D. (2008). Two thousand million? *English Today.*

Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., and Ranzato, M. (2019). Two new evaluation datasets for low-resource machine translation: Nepali-English and Sinhala-English. *arXiv preprint arXiv:1902.01382.*

Hardie, A., Lohani, R., Regmi, B., and Yadava, Y. (2005). Categorisation for automated morphosyntactic analysis of Nepali: introducing the Nelralec tagset (NT-01).

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference of Learning Representations.*

Kunchukuttan, A., Mehta, P., and Bhattacharyya, P. (2018). The IIT Bombay English-Hindi parallel corpus. *arXiv 1710.02855v2.*

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora.

Neubig, G. and Hu, J. (2018). Rapid adaptation of neural machine translation to new languages. *Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. *arXiv 1511.06709v4.*

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. *arXiv 1508.07909v5.*

Sennrich, R. and Volk, M. (2011). Iterative, MT-based sentence alignment of parallel texts. *NODALIDA 2011 Conference Proceedings*, pages 175–182.

Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems.*

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762.*

Wu, Y., Schuster, M., Chen, Z., V. Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Macduff, H., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv 1609.08144*.

Yadava, Y. P., Hardie, A., Lohani, R. R., Regmi, B. N., Gurung, S., Gurung, A., McEnery, T., Allwood, J., and Hall, P. (2008). Construction and annotation of a corpus of contemporary Nepali. *Corpora Vol. 3*, pages 213–225.

Appendix A

Example Translations

NE→EN

Source	ठूला गोदामहरुले , यस क्षेत्रका साना साना धेरै निर्माता हरु द्वारा बनाईएका जुत्ताहरु भण्डार गर्न थाले ।
Reference	Large warehouses began to stock footwear in warehouses , made by many small manufacturers from the area .
System	Large warehouses began to store shoe made by small producers of this area .

Source	प्राविधिक लेखकहरूले पनि व्यापारिक , पेशागत वा घरेलु प्रयोगका लागि विभिन्न कार्यविधिहरूका बारे लेख्दछन् ।
Reference	Technical writers also write various procedures for business , professional or domestic use .
System	Technical authors also write about various procedures for commercial , professional or domestic use .

Source	यी लक्षणहरू लामो अवधिसम्म देखा परिरहन्छन् र विशेषतः समयसँगै अझ बिग्रँदै जान्छ ।
Reference	This symptons appear for a long time and specially get worse with time .
System	These symptoms appear to be a long period of time and especially the time goes worse .

Source	सवारी साधन र टायर निर्माताहरूले कसरी टायरहरू कसरी जाँच गर्ने र मर्मतसम्भार गर्ने भन्ने विषयमा निर्देशनहरू सहितको निर्देशिका प्रदान गर्दछन् ।
Reference	Vehicle and tire manufacturers provide owner s &apos; manuals with instructions on how to check and maintain tires .
System	Ride means and Tyre manufacturers provide instructions with instructions about how to check Tyre and repair .

EN→NE

Source	On the occasion of Mangsir Shukla , people go to Bhatekhola for its value and on the same day Deusadi takes place in Dhami village .
Reference	मंसिरशुक्ल द्वितीयाकादिन विहान भातेखोलामा निमाउने पूजिने र त्यसको वँयाल गएर त्यसै दिन धामी गाउँमा दिउसडी हुने गर्दछ ।
System	मंगसिर शुक्लाका अवसरमा जनता आफ्नो मूल्य र त्यसै दिन देसादी गाउँमा गएर भोटेखोला जान्छन् ।

Source	People who have recently been depressed , a pulmonary rearrangement improves the quality of life , the capacity to exercise , and reduces death .
Reference	मानिसहरू जसलाई हालैमा प्रकोपन भएको छ , पल्मोनरी पुनर्सुधारले सम्पूर्ण जीवनको गुणस्तर , व्यायाम गर्ने क्षमतालाई सुधार गर्दछ , र मृत्युलाई कम गर्दछ ।
System	भर्खरै दबाब भएका मानिसहरुले जीवनको गुणस्तर सुधार गर्दछ , व्यायाम गर्ने क्षमता र मृत्यु घटाउंछ ।

Source	The body is Keshav &apos;s incantation and the scripture is the holy land .
Reference	शरिर केशवको मन्त्र र शास्त्र हिमद्यु उहि पाद तिर्थ ।
System	शरीर केशवको अर्थ र धर्मशास्त्र पवित्र भूमि हो ।

Source	So he established a circle of friends in Kathmandu with his uncle Naradev Pandey together with Laxmidutta Panta , Gopinath Lohani , Bhojraj Pandey , etc .
Reference	त्यसैले उनले काठमाडौमा आफ्ना मामा नरदेव पाण्डे तथा लक्ष्मीदत्त पन्त , गोपीनाथ लोहनी , भोजराज पाण्डे आदिसँग मिलेर ' मित्रमण्डली ' गठन गरे ।
System	त्यसैले उनले लक्समिदत्त पन्त , गोपिनाथ लोहनी , भोजराज पाण्डे आदिसँगै काठमाडौंमा साथीहरूको वृत्त स्थापित गरे ।