A3-108 Machine Translation System for Similar Language Translation Shared Task 2020

Saumitra Yadav, Manish Shrivastava
Machine Translation - Natural Lanugae Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
saumitra.yadav@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In this paper, we describe our submissions for Similar Language Translation Shared Task 2020. We built 12 systems in each direction for Hindi $\Longleftrightarrow$ Marathi language pair. This paper outlines initial baseline experiments with various tokenization schemes to train statistical models. Using optimal tokenization scheme among these we created synthetic source side text with back translation. And prune synthetic text with language model scores. This synthetic data was then used along with training data in various settings to build translation models. We also report configuration of the submitted systems and results produced by them.

1 Introduction

Machine Translation systems are models which aim to translate text from one language into another. There are multiple ways of building such a model (Rule Based, Data driven, Hybrid etc.). In this system description paper, we use data driven techniques to build MT systems. As the name suggests, data driven MT systems make use of parallel sentences (i.e. x^{th} sentence in two languages have same meaning). We make use of statistical (Koehn et al., 2003) and neural (Bahdanau et al., 2014) methods to build systems for Hindi Marathi pair.

Hindi Marathi language pair comes under purview of similar languages. Similar Languages are languages which exhibit lexical and structural similarities (Kunchukuttan et al., 2014a). This can be due to common ancestry or being in close proximity for long time. In current digital age communication, translation between similar language is a justifiable requirement. But there is a scarcity of good quality bitext for many language pairs, as is the case of Hindi Marathi. Hence, we used characteristics displayed by similar languages (in this case Hindi and Marathi) like similar form of spelling, pronunciation etc. Following Kunchukuttan and Bhattacharyya (2017) and Kunchukuttan et al. (2014b) we made use of byte pair encoding (Sennrich et al., 2016b) and morfessor toolkit (Virpioja et al., 2013) respectively as part of preprocessing step before training. Using cues from Koehn and Knowles (2017) and looking at the size of training data provided, we use statistical method to build initial models. To further salvage similarity between this language pair we made use of backtranslation (Sennrich et al., 2016a) to generate more synthetic data for further training using both neural and statistical methods.

For this shared task we developed 12 translation systems in each direction (Hindi $\Longleftrightarrow$ Marathi). To rank systems, we went through some test instances subjectively and also compared our BLEU scores with another Translation system. And chose top 2 systems in both direction using both subjective examination and detoknized BLEU scores. Subsequent sections give more detailed overview of systems developed.

2 Seed MT systems using different tokenization schemes

Experiments in Koehn and Knowles (2017) show that Statistical Machine Translation model fairs better when compared to Neural model in case of low resource setting. So, we make use of SMT model to make initial baseline systems using various tokenization schemes. We use these systems as seed system, used to create synthetic dataset for further training by back translation.

2.1 Data

For our initial experiments we just used parallel and monolingual corpora shared by the organizers. We include training data to monolingual corpus for each language (LM corpus) to make language model. Parallel text consisted of bitext from 3

Proceedings of the 5th Conference on Machine Translation (WMT), pages 451–455

Corpus/Language	Hindi				Marathi			
#of Tokens	basicTok	BPE	Morf	#of Sentences	basicTok	BPE	Morf	#of Sentences
Train	840863	977742	38246	38246	638467	867968	851394	38246
Dev	32106	36482	34600	1411	25552	33997	33828	1411
Monolingual	1455510657	1760885875	1629220967	77722389	4834280	6715047	6526439	369403

Table 1: Total number of Tokens in each file after various tokenization schemes, last sub-column in both languages column denotes total number of lines in respective corpus

sources namely *News*, *PM India*, *Indic WordNet*. *Indic Wordnet* is not used in training because we found multiple instances of sentence pairs in which one of the sentence was incomplete.

2.2 Preprocessing

We used the IndicNLP toolkit[1] to tokenize all corpora as first preprocessing step. Then we made use of a BPE (Sennrich et al., 2016b) model trained with 10000 merge operations on the LM corpus for both Hindi and Marathi. The resultant model was used to tokenize words to subwords in sentences for all texts. Morfessor (Virpioja et al., 2013) was also used as another alternative preprocessing step. We trained a morfessor model on the full LM corpus of Marathi and an equally sized Hindi Corpus. And taking cue from IndicNLP toolkit, we used '+' as delimiter when segmenting words into segments i.e. a word *xyz* which was to segment as *x yz* will segment as *x+ yz*. Table 1 shows statistics of preprocessed data. We used all possible combinations of tokenization schemes while training initial models, these tokenization schemes were,

- Basic tokenization denoted as BasicTok in Table 1 which make use of IndicNLP toolkit.
- BPE which tokenize words into subword and is denoted as BPE.
- Tokenization using Morfessor, which is denoted as Morfes.

2.3 Machine Translation Model

We made use of Moses toolkit (Koehn et al., 2007) to build statistical models trained with tokenized bitext. We also use GIZA++ (Och and Ney, 2003) to find alignments between parallel text and grow-diag-final-and method (Koehn et al., 2003) to extract aligned phrases. And utilize KenLM (Heafield, 2011) to train a trigram model with kneser ney smoothing on monolingual corpus of both languages. MERT (Och, 2003) is used for tuning the trained models. We evaluated these models on dev set. Results are given in Table 2.

2.4 Using back-translation to augment training data

Based on the results in Table 2 we make use of following tokenization schemes depending on direction of translation,

- BPE as tokenization preprocessing scheme on both languages when translation direction is from Hindi to Marathi.
- Morf as tokenization scheme for Marathi and Basic tokenization for Hindi when translating from Marathi to Hindi.

After translating monolingual corpus, we did the following post processing based on direction of translation,

- In case of Marathi to Hindi translation, in post processing we remove '+' delimiter. This is due to Marathi being morphological richer than Hindi.
- For Hindi to Marathi,we simply joined the subwords in text translated.

Due to time constraint we translated some part of Hindi monolingual corpus (Authentic$_{Hindi}$) to Marathi (Synthetic$_{Marathi}$). We used beam search with default setting in Moses for this translation. We used already trained LM from Section 2.3 to learn average LM score of BPE tokenized Marathi monolingual corpus. Synthetic$_{Marathi}$ is than pruned (SyntheticPruned$_{Marathi}$) by keeping back-translated sentences which have LM score higher then average LM score on aforementioned Corpus. Same process is followed while translating Authentic$_{Marathi}$ to Synthetic$_{Hindi}$ and further pruning to get SyntheticPruned$_{Hindi}$. Statistics related to back-translated data and resultant pruned corpus is given in Table 3.

[1] http://anoopkunchukuttan.github.io/indic_nlp_library/

Experiment.	Tokenization Based Exp	Hin To Mar	Mar To Hin
1	Hindi BasicTok – Mar BasicTok	19.937	24.542
2	Hindi BPE – Mar BasicTok	19.2251	23.13
3	Hindi Morfes. - Mar BasicTok	19.1327	23.44
4	Hindi BasicTok – Mar BPE	19.02	25.836
5	Hindi BPE – Mar BPE	**20.06**	26.07
6	Hindi Morfes. - Mar BPE	19.43	25.54
7	Hindi BasicTok – Mar Morfes.	19.37	**26.282**
8	Hindi BPE – Mar Morfes.	19.49	25.30
9	Hindi Morfes. - Mar Morfes.	19.33	26.03

Table 2: BLEU Scores on dev dataset when we use SMT models which are trained in all combinations of 3 tokenization schemes.

Back Translation direction L1 to L2	Hindi to Marathi	Marathi to Hindi
Sentences translated	456106	369403
Average KenLM Score of Monolingual data in L2	62.66	42.25
Sentences which are above this LM score	283043 (62.05%)	215417 (58.31%)
Average Sentence length of pruned corpus with standard deviation	10.25, 4.80	11.57, 4.52

Table 3: Statistics of back translated data

3 MT models using augmented bitext

For augmented data experiments we had following datasets available for training,

- Original training text
- Synthetic Marathi and Authentic Hindi
- Synthetic Pruned Marathi and Authentic Pruned Hindi
- Synthetic Hindi and Authentic Marathi
- Synthetic Pruned Hindi and Authentic Pruned Marathi

We ran experiments on following dataset combinations, for Hindi to Marathi Systems with BPE tokenization on both Hindi and Marathi,

1. Original training text + Synthetic Marathi and Authentic Hindi + Synthetic Hindi and Authentic Marathi
2. Original training text + Synthetic Pruned Marathi and Authentic Pruned Hindi + Synthetic Pruned Hindi and Authentic Pruned Marathi
3. Original training text + Synthetic Hindi and Authentic Marathi
4. Original training text + Synthetic Pruned Hindi and Authentic Pruned Marathi

And for Marathi to Hindi System, we ran following dataset combinations with morfessor model tokenization on Marathi and Basic Tokenization on Hindi,

1. Original training text + Synthetic Hindi and Authentic Marathi + Synthetic Marathi and Authentic Hindi
2. Original training text + Synthetic Pruned Hindi and Authentic Pruned Marathi + Synthetic Pruned Marathi and Authentic Pruned Hindi
3. Original training text + Synthetic Marathi and Authentic Hindi
4. Original training text + Synthetic Pruned Marathi and Authentic Pruned Hindi

All these dataset combinations were used to train following methods to build MT models with respective default configurations available in respective toolkits,

- SMT model using Moses toolkit (Koehn et al., 2007)
- NMT model with attention using Opennmt toolkit (Klein et al., 2017)
- NMT model with attention and copy attention (See et al., 2017) using Opennmt toolkit, to make use of similarity between Hindi Marathi language pair

4 Result

To submit two best systems out of 12 in each direction as directed by shared task, we did two evaluations. Firstly, we compared our system outputs to output of another publicly available translation model. Second, we went through some random outputs of all system outputs. We found that in most systems synthetic-authentic dataset which was not pruned with LM scores along with original training set performed better than pruned augmented bitext and original corpus. Following this, we selected following system outputs as our submission,

- Hindi to Marathi System:
  - Primary Submission: NMT with Attention + Original parallel text + SyntheticHindi_AuthenticMarathi
  - Contrastive Submission: NMT with Attention and CopyAttention + Original parallel text + SyntheticMarathi_AuthenticHindi + SyntheticHindi_AuthenticMarathi
- Marathi to Hindi System:
  - Primary Submission: NMT with Attention + Original parallel text + SyntheticMarathi_AuthenticHindi + SyntheticHindi_AuthenticMarathi
  - Contrastive Submission: SMT + Original parallel text + SyntheticMarathi_AuthenticHindi + SyntheticHindi_AuthenticMarathi

Table 4 gives the scores we received for these systems.

Language Direction	Submission Type	BLEU	RIBES	TER
Hindi to Marathi	Primary	11.41	57.2	79.96
Hindi to Marathi	Contrastive	10.21	55.17	82.01
Marathi to Hindi	Primary	18.32	59.31	77.35
Marathi to Hindi	Contrastive	21.11	60.76	77.28

Table 4: Scores for our systems

Both of our Hindi to Marathi Systems were somewhere in the middle compared to the other submissions. On the other hand Marathi to Hindi Contrastive submission (which was trained using SMT) was in top 5 standings.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Philipp Koehn, Franz J Och, and Daniel Marcu. 2003. Statistical phrase-based translation. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2017. Learning variable length units for SMT between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 14–24, Copenhagen, Denmark. Association for Computational Linguistics.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014a. Sata-anuvadak: Tackling multiway translation of indian languages. *pan*, 841(54,570):4–135.

Anoop Kunchukuttan, Ratish Pudupully, Rajen Chatterjee, Abhijit Mishra, and Pushpak Bhattacharyya. 2014b. The iit bombay smt system for icon 2014 tools contest. *NLP Tools Contest at ICON 2014*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, pages 160–167.

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

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

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

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.