Statistical Machine Translation for Indian Languages: Mission Hindi

Raj Nath Patel
KBCS, CDAC Mumbai
rajnathp@cdac.in

Prakash B. Pimpale
KBCS, CDAC Mumbai
prakash@cdac.in

Sasikumar M.
KBCS, CDAC Mumbai
sasi@cdac.in

Abstract

This paper discusses Centre for Development of Advanced Computing Mumbai's (CDACM) submission to the NLP Tools Contest on Statistical Machine Translation in Indian Languages (ILSMT) 2014 (collocated with ICON 2014). The objective of the contest was to explore the effectiveness of Statistical Machine Translation (SMT) for Indian language to Indian language and English-Hindi machine translation. In this paper, we have proposed that suffix separation and word splitting for SMT from agglutinative languages to Hindi significantly improves over the baseline (BL). We have also shown that the factored model with reordering outperforms the phrase-based SMT for English-Hindi (*en-hi*). We report our work on all five pairs of languages, namely Bengali-Hindi (*be-hi*), Marathi-Hindi (*mr-hi*), Tamil-Hindi (*ta-hi*), Telugu-Hindi (*te-hi*), and *en-hi* for Health, Tourism, and General domains.

1 Introduction

In this paper, we present our experiments on SMT from Bengali, Marathi, Tamil, Telugu and English to Hindi. From the set of languages involved in the shared task, Bengali, Hindi and Marathi belong to IndoAryan family and Tamil and Telugu are from Dravidian language family. All languages except English, have the same flexibility towards word order, canonically following the SOV structure.

With reference to the morphology, Bengali, Marathi, Tamil, and Telugu are more agglutinative compared to Hindi. It is known that SMT produces more unknown words resulting in the bad translation quality if the morphological divergence between the source and target language is high. Koehn and Knight (2003), Popovic and Ney (2004) and Popović et al. (2006) have demonstrated ways to handle this issue with morphological segmentation of words before training the SMT system. To tackle the morphological divergence of Hindi with these languages we have explored Suffix Separation (SS) and Compound word Splitting (CS) as a pre-processing step.

For English to Hindi SMT, better alignment is achieved through the use of preordering developed by Patel et al. (2013) and stem as an alignment factor (Koehn and Hoang, 2007). The rest of the paper is organized as follows. In Section 2, we discuss our methodology, followed by dataset and experimental setup in section 3. Section 4 discusses experiments and results. Submitted systems to the shared task and error analysis are displayed in section 5 and 6 respectively, followed by conclusion and future work in section 7.

2 Methodology

Our methodology to tackle morphological and structural divergence involves the use of the suffix separation, compound splitting, and reordering for the language pairs under study. These methods are briefly described below. Pseudocode for the suffix separation and compound word splitting is detailed in Algorithms 1 and 2 respectively.

2.1 Suffix Separation (SS)

In this step, the source language words are preprocessed for suffix separation. We have considered only suffix from source language which corresponds to post-positions in Hindi. For example, in Marathi, [1] '***mahinyaaMnii***' is translated as '***mahiine meM***' in Hindi. In this case, we split

[1] All ***Non-English*** **(Marathi and Hindi)** words have been written in Itrans using http://sanskritlibrary.org/transcodeText.html

Algorithm 1 Suffix Separation

```
procedure SUFFIXSEP(word)
    suffixSet ← read file suffix list
    splits ← {word, "NULL"}
    for suffix ← suffixSet do
        if then word.ENDSWITH = suffix & word.LENGTH > suffix.LENGTH
            splits[0] ← word.SUBSTRING(0, word.LASTINDEXOF(suffix))
            splits[1] ← suffix return splits
        end if
    end for
end procedure
```

Algorithm 2 Compound Splitting

```
procedure COMPOUNDSPLIT
    vocab ← read monoligual corpus unique word list
    for word ← file do
        for vcb ← vocab do
            if then word.ENDSWITH = vcb & word.LENGTH > vcb.LENGTH+5
                compoundSuffix ← vcb
                Delete vcb for vocab return compoundSuffix
            end if
        end for
    end for
end procedure
```

the word into '***mahiny + aaMnii***'. Here, the suffix '***aaMnii***' corresponds to the word '***meM***' in Hindi. For this task, the list of suffixes is manually created with the linguistic expertise. When a word is subjected to SS, longest matching suffix from the list is considered for suffix separation. Suffix separation takes place only once for a word.

2.2 Compound Splitting (CS)

In this step source language compound words are split into constituents, recursively. For example, in Marathi, a compound word '***daMtatajGYaaMkaDuuna***' is translated as '***danta visheShaGYa se***' in Hindi. In this case we split the source word into constituents, '***daMta***' , '***tajGYaaM***' and '***kaDuuna***'. The list of the constituent suffixes for splitting is empirically prepared from a monolingual data. The compound suffix creation algorithm is very basic and simple, the pseudo code is detailed in Algoritm 2. A word is considered for compound suffix list, if it appears as a suffix in another word of the monolingual corpus.

2.3 Reordering (RO)

It is based on the syntactic transformation of the English sentence parse tree according to the target language (Hindi) structure. We have used source side reordering developed by Patel et al. (2013), and Ramanathan et al. (2008).

	health	tourism	general
training (TM)	24K	24K	48K
training (LM)	71K	71K	71K
test1	500	500	1000
test2	500	500	1000

Table 1: Sentence count for training, testing and development data; TM: Translation Model, LM: Language Model

3 Data-set and Experimental Setup

We now discuss training and testing corpus from health, tourism and general domains for *be-hi*, *mr-hi*, *ta-hi*, *te-hi*, and *en-hi* language pairs, followed by preprocessing, SMT system setup and evaluation metrics for experiments.

3.1 Corpus for SMT Training and Testing

For experiments, we have used corpus shared by ILSMT organizers, detailed in Table 1. Additional monolingual corpus of approx., 23K sentences (Khapra et al., 2010) is used to train the language model. The evaluation of the systems were done using Test1 data which was the development set. The quality of the submitted systems were estimated by the organizers against Test2 corpus.

3.2 Pre-Processing

To tackle the morphological divergence between the source and target languages (Bengali/Marathi/Tamil/Telugu to Hindi), we used suf-

fix separation and compound splitting, as explained in section 2. To handle the structural divergence for English-Hindi SMT, we exploited source side preordering (Patel et al., 2013; Ramanathan et al., 2008).

3.3 SMT System Set up

The baseline system was setup using the phrase-based model ((Brown et al., 1990; Marcu and Wong, 2002; Och and Ney, 2003; Koehn et al., 2003) and Koehn et al. (2007) was used for factored model. The language model was trained using KenLM (Heafield, 2011) toolkit with modified Kneser-Ney smoothing (Chen and Goodman, 1996). For factored SMT training, we used source and target side stem as an alignment factor. Stemming was done using lightweight stemmer (Ramanathan and Rao, 2003) for Hindi. For English, we used porter stemmer (Minnen et al., 2001).

3.4 Evaluation Metrics

We compared different experimental systems using BLEU (Papineni et al., 2002), NIST (Doddington, 2002) and translation edit rate (TER) (Snover et al., 2006). For a MT system to be better, higher BLEU and NIST scores with lower TER are desired.

4 Experiments and Results

In the following subsections, we present different experiments carried out for the shared task. We also study the impact of suffix separation, compound splitting and preordering on the SMT accuracy.

4.1 Impact of Suffix Separation and Compound Splitting

Pre-processing of source words for suffix separation and compound word splitting results in better alignment and hence the better translation. The alignment improvements can be seen in the Table 2. Improvement in the translation quality can be observed in the various evaluation scores detailed in Table 3. From the table, we can infer that for *be-hi* and *mr-hi*, suffix separation have shown significant improvements over the baseline, whereas, compound word splitting has caused slight improvement. However, compound splitting has found to be more effective than suffix separation for *ta-hi* and *te-hi*.

We have also tried a combination of compound word splitting and suffix separation, serially. We observed improvements over the BL+SS and BL+CS for *ta-hi* and *te-hi*, across the evaluation scores. BL+SS is better than all other systems for *be-hi*, across the evaluation metrics, whereas, for *mr-hi* BL+CS+SS has highest BLEU and BL+SS has the best NIST and TER scores. Compound splitting for Bengali and Marathi can be further investigated to improve the systems.

	models	BLEU	NIST	TER
be-hi	BL	30.16	6.906	48.26
	BL+SS	**31.73**	**6.993**	**46.92**
	BL+CS	30.33	6.792	49.08
	BL+CS+SS	30.34	6.804	48.58
mr-hi	BL	35.56	7.478	42.98
	BL+SS	39.84	**7.877**	**39.44**
	BL+CS	37.87	7.502	43.01
	BL+CS+SS	**39.88**	7.796	39.93
te-hi	BL	16.76	4.772	64.48
	BL+SS	17.49	4.927	63.76
	BL+CS	19.82	5.260	62.96
	BL+CS+SS	**20.16**	**5.300**	**62.69**
ta-hi	BL	24.89	6.205	52.31
	BL+SS	27.18	6.473	50.39
	BL+CS	27.47	6.448	51.01
	BL+CS+SS	**28.95**	**6.535**	**49.95**

Table 3: Effect of suffix separation and compound splitting

4.2 Impact of Reordering

Preordering of the source language sentence helps in the better alignment and decoding for English to Indian language (Ramanathan et al., 2008; Patel et al., 2013; Kunchukuttan et al., 2014) SMT. Table 4 details the results for the systems under study. We can see that BL+RO shows significant improvement over BL. Further, the factored SMT system with stem as alignment factor shows slight improvement in BLEU over the BL+RO, but other metrics show BL+RO is better compared to the factored system.

	models	BLEU	NIST	TER
en-hi	BL	18.52	5.813	64.33
	BL+RO	22.72	**6.035**	**59.85**
	BL+RO+FACT	**22.83**	5.994	60.17

Table 4: Effect of source reordering (RO) and factor (FACT)

5 Submission

In this section, we discuss evaluation scores of the submitted systems. We submitted BL+SS for *be-*

SRC sentence (mr)	*dara*	*sahaa*	*mahinyaaMnii*	*daMtatajGYaaMkaDuuna*	*tapaasuuna*	*ghyaa*			
BL aligned (hi)	*hara*	*Chaha*	*mahiine meM*	*danta visheShaGYa se chekaapa karaaeM*	-	-			
SRC' Sentence (mr)	*dara*	*sahaa*	*mahiny*	*aaMnii*	*daMta*	*tajGYaaM*	*kaDuuna*	*tapaasuuna*	*ghyaa*
BL+CS+SS aligned (hi)	*hara*	*Chaha*	*mahiine*	*meM*	*danta*	*visheShaGYa*	*se*	*chekaapa*	*karaaeM*

Table 2: Improvement in alignments; SRC: Source, SRC': Pre-processed Source

			health			tourism			general		
			BLEU	NIST	TER	BLEU	NIST	TER	BLEU	NIST	TER
be-hi	BL	T1	30.78	6.628	47.68	29.35	6.389	48.95	30.16	6.906	48.26
	FSS	T1	31.95	6.746	45.98	30.76	6.433	47.98	31.73	6.993	46.92
		T2	**31.40**	6.723	**45.55**	**31.17**	6.605	**46.25**	**31.34**	7.079	**45.98**
mr-hi	BL	T1	35.42	7.103	42.52	35.20	6.917	43.81	35.56	7.478	42.98
	FSS	T1	39.03	7.390	39.85	39.45	7.148	41.06	39.88	7.796	39.93
		T2	**39.31**	7.192	**41.85**	**37.73**	7.199	**41.25**	**39.10**	7.718	**41.02**
ta-hi	BL	T1	17.99	4.646	63.79	14.44	4.216	65.90	16.76	4.772	64.48
	FSS	T1	20.83	5.104	62.18	17.86	4.682	64.66	20.16	5.300	62.69
		T2	**20.69**	5.075	**62.59**	**17.07**	4.714	**63.78**	**19.81**	5.247	**62.33**
te-hi	BL	T1	24.83	5.856	52.25	23.85	5.679	53.57	24.89	6.205	52.31
	FSS	T1	29.90	6.322	49.41	27.45	5.934	51.69	28.95	6.535	49.95
		T2	**30.38**	6.373	**49.28**	**26.21**	5.862	**52.90**	**28.95**	6.569	**50.47**
en-hi	BL	T1	20.12	5.840	61.49	16.97	5.233	66.83	18.52	5.813	64.33
	FSS	T1	23.16	5.870	58.54	20.98	5.342	62.90	22.83	5.994	60.17
		T2	**21.84**	5.662	**60.60**	**19.45**	5.349	**62.95**	**20.97**	5.837	**61.56**

Table 5: Consolidated Results; FSS: Final Submitted System; T1: Test1; T2: Test2; **BOLD** scores were provided by the organizers

hi and BL+CS+SS for all other language pairs except *en-hi*. For *en-hi* BL+RO+FACT is submitted as the final system. Table 5 summarizes the evaluation of the submission against Test1 and Test2.

6 Error Analysis

A closer look at the performance of these systems to understand the utility of SS and CS has been done. We report a few early observations.

6.1 Superfluous Splitting

With the suffix separation, the Marathi word like '***dilaavara***' is getting split into '***dilaa***' + '***vara***' which is a wrong split. As, '***dilaavara***' is a proper noun and hence should not have been split. We tried to overcome this error by avoiding suffix separation and compound splitting of NNP POS tagged words, but that was stopping many other valid candidates from pre-processing.

6.2 Bad Split

The words like '***jarmaniitiila***' is getting split into '***jarmaniit*** + '***iila*** which actually should have been split into '***jarmanii*** +'***tiila***'. Similarly, many words on splitting have not given any valid Marathi word which causesed sparsity in training data up to some extent.

7 Conclusion and Future Work

In this paper, we presented various systems for translation from Bengali, English, Marathi, Tamil and Telugu to Hindi. These SMT systems with the use of source side suffix separation, compound splitting and preordering showed significantly higher accuracy over the baseline. In future, we could investigate the formulation of more effective solutions for SS, CS and RO. Reasons for lower BLEU due to the combination of suffix separation and compound word splitting for Bengali and Marathi is another interesting case to study further.

References

[Brown et al.1990] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A Statistical Approach to Machine Translation. *Computational linguistics*, 16(2):79–85.

[Chen and Goodman1996] Stanley F Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing

Techniques for Language Modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics.

[Doddington2002] George Doddington. 2002. Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

[Heafield2011] Kenneth Heafield. 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

[Khapra et al.2010] Mitesh M Khapra, Anup Kulkarni, Saurabh Sohoney, and Pushpak Bhattacharyya. 2010. All Words Domain Adapted WSD: Finding a Middle Ground between Supervision and Unsupervision. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1532–1541. Association for Computational Linguistics.

[Koehn and Hoang2007] Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *EMNLP-CoNLL*, pages 868–876.

[Koehn and Knight2003] Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 187–193. Association for Computational Linguistics.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

[Koehn et al.2007] 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.

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

[Marcu and Wong2002] Daniel Marcu and William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 133–139. Association for Computational Linguistics.

[Minnen et al.2001] Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering*, 7(03):207–223.

[Och and Ney2003] Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational linguistics*, 29(1):19–51.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

[Patel et al.2013] Raj Nath Patel, Rohit Gupta, Prakash B Pimpale, and Sasikumar M. 2013. Reordering rules for english-hindi smt. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 34–41. Association for Computational Linguistics.

[Popovic and Ney2004] Maja Popovic and Hermann Ney. 2004. Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. In *LREC*.

[Popović et al.2006] Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical Machine Translation of German Compound Words. In *Advances in Natural Language Processing*, pages 616–624. Springer.

[Ramanathan and Rao2003] Ananthakrishnan Ramanathan and Durgesh D Rao. 2003. A Lightweight Stemmer for Hindi. In *The Proceedings of EACL*.

[Ramanathan et al.2008] Ananthakrishnan Ramanathan, Jayprasad Hegde, Ritesh M Shah, Pushpak Bhattacharyya, and M Sasikumar. 2008. Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation. In *Proceedings of International Joint Conference on NLP (IJCNLP08)*, pages 513–520.

[Snover et al.2006] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.