Faster decoding for subword level Phrase-based SMT between related languages

Anoop Kunchukuttan, Pushpak Bhattacharyya
Center For Indian Language Technology,
Department of Computer Science & Engineering
Indian Institute of Technology Bombay
{anoopk,pb}@cse.iitb.ac.in

Abstract

A common and effective way to train translation systems between related languages is to consider sub-word level basic units. However, this increases the length of the sentences resulting in increased decoding time. The increase in length is also impacted by the specific choice of data format for representing the sentences as subwords. In a phrase-based SMT framework, we investigate different choices of *decoder parameters* as well as *data format* and their impact on decoding time and translation accuracy. We suggest best options for these settings that significantly improve decoding time with little impact on the translation accuracy.

1 Introduction

Related languages are those that exhibit lexical and structural similarities on account of sharing a **common ancestry** or being in **contact for a long period of time** (Bhattacharyya et al., 2016). Examples of languages related by common ancestry are Slavic and Indo-Aryan languages. Prolonged contact leads to convergence of linguistic properties even if the languages are not related by ancestry and could lead to the formation of *linguistic areas* (Thomason, 2000). Examples of such linguistic areas are the Indian subcontinent (Emeneau, 1956), Balkan (Trubetzkoy, 1928) and Standard Average European (Haspelmath, 2001) linguistic areas. Both forms of language relatedness lead to related languages sharing vocabulary and structural features.

There is substantial government, commercial and cultural communication among people speaking related languages (Europe, India and South-East Asia being prominent examples and linguistic regions in Africa possibly in the future). As these regions integrate more closely and move to a digital society, translation between *related* languages is becoming an important requirement. In addition, translation to/from related languages to a *lingua franca* like English is also very important. However, in spite of significant communication between people speaking related languages, most of these languages have few parallel corpora resources. It is therefore important to leverage the relatedness of these languages to build good-quality statistical machine translation (SMT) systems given the lack of parallel corpora.

Modelling the lexical similarity among related languages is the key to building good-quality SMT systems with limited parallel corpora. **Lexical similarity** implies that the languages share many words with the similar form (spelling/pronunciation) and meaning *e.g.* `blindness` is `andhapana` in Hindi, `aandhaLepaNaa` in Marathi. These words could be cognates, lateral borrowings or loan words from other languages.

Sub-word level transformations are an effective way for translation of such shared words. Using subwords as basic units of translation has been shown to be effective in improving translation quality with limited parallel corpora. Subword units like character (Vilar et al., 2007; Tiedemann, 2009a), character n-gram (Tiedemann and Nakov, 2013) and orthographic syllables (Kunchukuttan and Bhattacharyya, 2016) have been explored and have been shown to improve translation quality to varying degrees.

Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects,
pages 82–88, Osaka, Japan, December 12 2016.

Original	this is an example of data formats for segmentation
Subword units	thi s i s a n e xa m p le o f da ta fo rma t s fo r se gme n ta tio n
Internal Marker	thi_ s i_ s a_ n e_ xa_ m_ p_ le o_ f da_ ta fo_ rma_ t_ s fo_ r se_ gme_ n_ ta_ tio_ n
Boundary Marker	thi s_ i s_ a n_ e xa m p le_ o f_ da ta_ fo rma t s_ fo r_ se gme n ta tio n_
Space Marker	thi s _ i s _ a n _ e xa m p le _ o f _ da ta _ fo rma t s _ fo r _ se gme n ta tio n

Table 1: Formats for sentence representation with subword units (example of orthographic syllables)

However, the use of subword units increases the sentence length. This increases the training, tuning and decoding time for phrase-based SMT systems by an order of magnitude. This makes experimentation costly and time-consuming and impedes faster feedback which is important for machine translation research. Higher decoding time also makes deployment of MT systems based on subword units impractical.

In this work, we systematically study the choice of data format for representing sentences and various decoder parameters which affect decoding time. Our studies show that the use of cube-pruning during tuning as well as testing with a lower value of the stack pop limit parameter improves decoding time substantially with minimal change in translation quality.

The rest of the paper is organized as follows. Section 2 discusses the factors that affect decoding time which have been studied in this paper. Section 3 discusses our experimental setup. Section 4 discusses the results of our experiments with decoder parameters. Section 5 discusses the results of our experiments with corpus formats. Section 6 discusses prior work related to optimizing decoders for phrase-based SMT. Section 7 concludes the paper.

2 Factors affecting decoding time

This section describes the factors affecting the decoding time that have been studied in this paper.

2.1 Unit of translation

The decoding time for a sentence is proportional to length of the sentence (in terms of the basic units). Use of subword units will obviously result in increased sentence length. Various units have been proposed for translation (character, character n-gram, orthographic syllable, morpheme, etc.). We analysed the average length of the input sentence on four language pairs (Hindi-Malayalam, Malayalam-Hindi, Bengali-Hindi, Telugu-Malayalam) on the ILCI corpus (Jha, 2012). The average length of an input sentence for character-level representation is 7 times of the word-level input, while it is 4 times the word-level input for orthographic syllable level representation. So, the decoding time will increase substantially.

2.2 Format for sentence representation

The length of the sentence to be decoded also depends on how the subword units are represented. We compare three popular formats for representation, which are illustrated in Table 1:

- **Boundary Marker**: The subword at the boundary of a word is augmented with a marker character. There is one boundary subword, either the first or the last chosen as per convention. Such a representation has been used in previous work, mostly related to morpheme level representation.

- **Internal Marker**: Every subword internal to the word is augmented with a marker character. This representation has been used rarely, one example being the Byte Code Encoding representation used by University of Edinburgh's Neural Machine Translation system (Williams et al., 2016; Sennrich et al., 2016).

- **Space Marker**: The subword units are not altered, but inter-word boundary is represented by a space marker. Most work on translation between related languages has used this format.

For boundary and internal markers, the addition of the marker character does not change the sentence length, but can create two representations for some subwords (corresponding to internal and boundary positions), thus introducing some data sparsity. On the other hand, space marker doubles the sentence length (in terms in words), but each subword has a unique representation.

2.3 Decoder Parameters

Given the basic unit and the data format, some important decoder parameters used to control the search space can affect decoding time. The decoder is essentially a search algorithm, and we investigated important settings related to two search algorithms used in the *Moses* SMT system: (i) stack decoding, (ii) cube-pruning (Chiang, 2007). We investigated the following parameters:

- **Beam Size**: This parameter controls the size of beam which maintains the best partial translation hypotheses generated at any point during stack decoding.
- **Table Limit**: Every source phrase in the phrase table can have multiple translation options. This parameter controls how many of these options are considered during stack decoding.
- **Cube Pruning Pop Limit**: In the case of cube pruning, the parameter limits the number of hypotheses created for each stack .

Having a lower value for each of these parameters reduces the search space, thus reducing the decoding time. However, reducing the search space may increase search errors and decrease translation quality. Our work studies this time-quality trade-off.

3 Experimental Setup

In this section, we describe the language pairs and datasets used, the details of our experiments and evaluation methodology.

3.1 Languages and Dataset

We experimented with four language pairs (Bengali-Hindi, Malayalam-Hindi, Hindi-Malayalam and Telugu-Malayalam). Telugu and Malayalam belong to the Dravidian language family which are agglutinative. Bengali and Hindi are Indo-Aryan languages with a relatively poor morphology. The language pairs chosen cover different combinations of morphological complexity between source and target languages.

We used the multilingual ILCI corpus for our experiments (Jha, 2012), consisting of sentences from tourism and health domains. The data split is as follows – *training: 44,777, tuning: 1000, test: 500* sentences.

3.2 System details

As an example of subword level representation unit, we have studied the orthographic syllable (OS) (Kunchukuttan and Bhattacharyya, 2016) in our experiments. The OS is a linguistically motivated, variable length unit of translation, which consists of one or more consonants followed by a vowel (a C^+V unit). But our methodology is not specific to any subword unit. Hence, the results and observations should hold for other subword units also. We used the *Indic NLP Library*[1] for orthographic syllabification.

Phrase-based SMT systems were trained with OS as the basic unit. We used the *Moses* system (Koehn et al., 2007), with *mgiza*[2] for alignment, the *grow-diag-final-and* heuristic for symmetrization of word alignments, and Batch MIRA (Cherry and Foster, 2012) for tuning. Since data sparsity is a lesser concern due to small vocabulary size and higher order n-grams are generally trained for translation using subword units (Vilar et al., 2007), we trained 10-gram language models. The language model was trained on the training split of the target language corpus.

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

[2] https://github.com/moses-smt/mgiza

	Translation Accuracy				Relative Decoding Time			
	ben-hin	**hin-mal**	**mal-hin**	**tel-mal**	**ben-hin**	**hin-mal**	**mal-hin**	**tel-mal**
default (stack, tl=20,ss=100)	33.10	11.68	19.86	9.39	46.44	65.98	87.98	76.68
Stack								
tl=10	32.84	11.24	19.21	9.47	35.49	48.37	67.32	80.51
tl=5	32.54	11.01	18.39	9.29	15.05	21.46	30.60	41.52
ss=50	33.10	11.69	19.89	9.36	17.33	25.81	35.76	43.45
ss=10	33.04	11.52	19.51	9.38	4.49	7.32	10.18	11.75
+tuning	32.83	11.01	19.57	9.23	5.24	8.85	11.60	9.31
Cube Pruning								
pl=1000	33.05	11.47	19.66	9.42	5.67	9.29	12.38	17.85
+tuning	33.12	11.3	19.77	9.35	7.68	13.06	15.18	14.56
pl=100	32.86	10.97	18.74	9.15	2.00	4.22	5.41	5.29
pl=10	31.93	9.42	15.26	8.5	1.51	3.64	4.57	3.84
Word-level	31.62	9.67	15.69	7.54	100.56 ms	65.12 ms	50.72 ms	42.4 ms

Table 2: Translation accuracy and Relative decoding time for orthographic syllable level translation using different decoding methods and parameters. Relative decoding time is indicated as a multiple of word-level decoding time. The following methods & parameters in *Moses* have been experimented with: (i) *normal stack decoding* - vary `ss`: stack-size, `tt`: table-limit; (ii) *cube pruning*: vary `pl`:cube-pruning-pop-limit. `+tuning` indicates that the decoder settings mentioned on previous row were used for tuning too. Translation accuracy and decode time per sentence for word-level decoding (in milliseconds) is shown on the last line for comparison.

The PBSMT systems were trained and decoded on a server with Intel Xeon processors (2.5 GHz) and 256 GB RAM.

3.3 Evaluation

We use BLEU (Papineni et al., 2002) for evaluating translation accuracy. We use the sum of user and system time minus the time for loading the phrase table (all reported by Moses) to determine the time taken for decoding the test set.

4 Effect of decoder parameters

We observed that the decoding time for OS-level models is approximately 70 times of the word-level model. This explosion in the decoding time makes translation highly compute intensive and difficult to perform in real-time. It also makes tuning MT systems very slow since tuning typically requires multiple decoding runs over the tuning set. Hence, we experimented with some heuristics to speed up decoding.

For normal stack decoding, two decoder parameters which impact the decode time are: (1) *beam size* of the hypothesis stack, and (2) *table-limit*: the number of translation options for each source phrase considered by the decoder. Since the vocabulary of the OS-level model is far less than that of the word-level model, we hypothesize that lower values for these parameters can reduce the decoding time without significantly affecting the accuracy. Table 2 shows the results of varying these parameters. We can see that with a beam size of 10, the decoding time is now about 9 times that of word-level decoding. This is a 7x improvement in decoding time over the default parameters, while the translation accuracy drops by less than 1%. If a beam size of 10 is used while decoding too, the drop in translation accuracy is larger (2.5%). Using this beam size during decoding also slightly reduces the translation accuracy. On the other hand, reducing the table-limit significantly reduces the translation accuracy, while resulting in lesser gains in decoding time.

We also experimented with *cube-pruning* (Chiang, 2007), a faster decoding method first proposed for use with hierarchical PBSMT. The decoding time is controlled by the `pop-limit` parameter in the Moses implementation of cube-pruning. With a pop-limit of 1000, the decoding time is about 12 times

	Translation Accuracy				Relative Decoding Time			
	ben-hin	**hin-mal**	**mal-hin**	**tel-mal**	**ben-hin**	**hin-mal**	**mal-hin**	**tel-mal**
default (stack, tl=20,ss=100)	27.29	6.72	12.69	6.06	206.98	391.00	471.96	561.00
Cube Pruning pl=1000	26.98	6.57	11.94	5.99	10.23	19.57	24.59	26.20
Word-level	31.62	9.67	15.69	7.54	100.56 ms	65.12 ms	50.72 ms	42.4 ms

Table 3: Translation accuracy and Relative decoding time for character level translation using different decoding methods and parameters. Relative decoding time is indicated as a multiple of word-level decoding time. Translation accuracy and decode time per sentence for word-level decoding (in milliseconds) is shown on the last line for comparison.

	Translation Accuracy				Relative Decoding Time			
	ben-hin	**hin-mal**	**mal-hin**	**tel-mal**	**ben-hin**	**hin-mal**	**mal-hin**	**tel-mal**
Boundary Marker	32.83	**12.00**	**20.88**	9.02	**7.44**	11.80	17.08	18.98
Internal Marker	30.10	10.53	19.08	7.53	7.82	**10.81**	**14.43**	17.06
Space Marker	**33.12**	11.30	19.77	**9.35**	7.68	13.06	15.18	**14.56**
Word-level	31.62	9.67	15.69	7.54	100.56 ms	65.12 ms	50.72 ms	42.4 ms

Table 4: Translation accuracy and Relative decoding time for orthographic syllable level translation using different data formats. Relative decoding time is indicated as a multiple of word-level decoding time. Translation accuracy and decode time per sentence (in milliseconds) for word-level decoding is shown on the last line for comparison.

that of word-level decoding. The drop in translation accuracy is about 1% with a 6x improvement over default stack decoding, even when the model is tuned with a pop-limit of 1000. Using this pop-limit during tuning also hardly impacts the translation accuracy. However, lower values of pop-limit reduce the translation accuracy.

While our experiments primarily concentrated on OS as the unit of translation, we also compared the performance of stack decoding and cube pruning for character lavel models. The results are shown in Table 3. We see that character level models are 4-5 times slower than OS level models and hundreds of time slower than word level models with the default stack decoding. In the case of character based models also, the use of cube pruning (with `pop-limit`=1000) substantially speeds up decoding (20x speedup) with only a small drop in BLEU score.

To summarize, we show that reducing the beam size for stack decoding as well as using cube pruning help to improve decoding speed significantly, with only a marginal drop in translation accuracy. Using cube-pruning while tuning only marginally impacts translation accuracy.

5 Effect of corpus format

For these experiments, we used the following decoder parameters: cube-pruning with cube-pruning-pop-limit=1000 for tuning as well as testing. Table 4 shows the results of our experiments with different corpus formats.

The internal boundary marker format has a lower translation accuracy compared to the other two formats whose translation accuracies are comparable. In terms of decoding time, no single format is better than the others across all languages. Hence, it is recommended to use the space or boundary marker format for phrase-based SMT systems. Neural MT systems based on encoder decoder architectures, particularly without attention mechanism, are more sensitive to sentence length, so we presume that the boundary marker format may be more appropriate.

6 Related Work

It has been recognized in the past literature on translation between related languages that the increased length of subword level translation is challenge for training as well as decoding (Vilar et al., 2007). Aligning long sentences is computationally expensive, hence most work has concentrated on corpora with short sentences (*e.g.* OPUS (Tiedemann, 2009b)) (Tiedemann, 2009a; Nakov and Tiedemann, 2012; Tiedemann, 2012). To make alignment feasible, Vilar et al. (2007) used the phrase table learnt from word-level alignment, which will have shorter parallel segments, as parallel corpus for training subword-level models. Tiedemann and Nakov (2013) also investigated reducing the size of the phrase table by pruning, which actually improved translation quality for character level models. The authors have not reported the decoding speed, but it is possible that pruning may also improve decoding speed since fewer hypothesis may have to be looked up in the phrase table, and smaller phrase tables can be loaded into memory.

There has been a lot of work looking at optimizing specific components of SMT decoders in a general setting. Hoang et al. (2016) provide a good overview of various approaches to optimizing decoders. Some of the prominent efforts include efficient language models (Heafield, 2011), lazy loading (Zens and Ney, 2007), phrase-table design (Junczys-Dowmunt, 2012), multi-core environment issues (Fernández et al., 2016), efficient memory allocation (Hoang et al., 2016), alternative stack configurations (Hoang et al., 2016) and alternative decoding algorithms like cube pruning (Chiang, 2007).

In this work, we have investigated stack decoding configurations and cube pruning as a way of optimizing decoder performance for the translation between related languages (with subword units and monotone decoding). Prior work on comparing stack decoding and cube-pruning has been limited to word-level models (Huang and Chiang, 2007; Heafield et al., 2014).

7 Conclusion and Future Work

We systematically study the choice of data format for representing subword units in sentences and various decoder parameters which affect decoding time in a phrase-based SMT setting. Our studies (using OS and character as basic units) show that the **use of cube-pruning during tuning as well as testing with a lower value of the stack pop limit parameter** improves decoding time substantially with minimal change in translation quality. Two data formats, the space marker and the boundary marker, perform roughly equivalently in terms of translation accuracy as well as decoding time. Since the tuning step contains a decoder in the loop, these settings also reduce the tuning time. We plan to investigate reduction of the time required for alignment.

Acknowledgments

We thank the Technology Development for Indian Languages (TDIL) Programme and the Department of Electronics & Information Technology, Govt. of India for their support. We also thank the anonymous reviewers for their feedback.

References

Pushpak Bhattacharyya, Mitesh Khapra, and Anoop Kunchukuttan. 2016. Statistical machine translation between related languages. In *NAACL Tutorials*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, June.

Murray B Emeneau. 1956. India as a lingustic area. *Language*.

M Fernández, Juan C Pichel, José C Cabaleiro, and Tomás F Pena. 2016. Boosting performance of a statistical machine translation system using dynamic parallelism. *Journal of Computational Science*.

Martin Haspelmath. 2001. The european linguistic area: Standard average european. In *Language Typology and Language Universals*.

Kenneth Heafield, Michael Kayser, and Christopher D Manning. 2014. Faster phrase-based decoding by refining feature state. In *Annual Meeting-Association For Computational Linguistics*, pages 130–135.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*.

Hieu Hoang, Nikolay Bogoychev, Lane Schwartz, and Marcin Junczys-Dowmunt. 2016. Fast, scalable phrase-based smt decoding. In *arXiv Pre-print arXiv:1610.04265*.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Annual Meeting-Association For Computational Linguistics*.

Girish Nath Jha. 2012. The TDIL program and the Indian Language Corpora Initiative. In *Language Resources and Evaluation Conference*.

Marcin Junczys-Dowmunt. 2012. A space-efficient phrase table implementation using minimal perfect hash functions. In *International Conference on Text, Speech and Dialogue*.

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*.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2016. Orthographic syllable as basic unit for smt between related languages. In *Empirical Methods in Natural Language Processing*.

Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Sarah Thomason. 2000. Linguistic areas and language history. In *lLanguages in Contact*.

Jörg Tiedemann and Preslav Nakov. 2013. Analyzing the use of character-level translation with sparse and noisy datasets. In *RANLP*.

Jörg Tiedemann. 2009a. Character-based psmt for closely related languages. In *Proceedings of the 13th Conference of the European Association for Machine Translation (EAMT 2009)*.

Jörg Tiedemann. 2009b. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*.

Jörg Tiedemann. 2012. Character-based pivot translation for under-resourced languages and domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

Nikolai Trubetzkoy. 1928. Proposition 16. In *Actes du premier congres international des linguistes La Haye*.

David Vilar, Jan-T Peter, and Hermann Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Barry Haddow, and Ondrej Bojar. 2016. Edinburghs statistical machine translation systems for wmt16. In *Proceedings of the First Conference on Machine Translation*.

Richard Zens and Hermann Ney. 2007. Efficient phrase-table representation for machine translation with applications to online mt and speech translation. In *HLT-NAACL*.