Phrase-based Transliteration System with Simple Heuristics

Avinesh PVS and **Ankur Parikh**
IIIT Hyderabad
Language Technologies Research Centre
{avinesh,shaileshkumar.parikh}@students.iiit.ac.in

Abstract

This paper presents modeling of transliteration as a phrase-based machine translation system. We used a popular phrase-based machine translation system for English-Hindi machine transliteration. We have achieved an accuracy of 38.1% on the test set. We used some basic rules to modulate the existing phrased-based transliteration system. Our experiments show that phrase-based machine translation systems can be adopted by modulating the system to fit the transliteration problem.

1 Introduction

Transliteration is the practice of converting a text from one writing system into another in a systematic way. Most significantly it is used in Machine Translation (MT) systems, Information Retrieval systems where a large portion of unknown words (out of vocabulary) are observed. Named entities (NE), technical words, borrowed words and loan words constitute the majority of the unknown words. So, transliteration can also be termed as the process of obtaining the phonetic translation of names across various languages (Shishtla et al., 2009). Transcribing the words from one language to another without the help of bilingual dictionary is a challenging task.

Previous work in transliteration include (Surana and Singh, 2009) who propose a transliteration system using two different approaches of transliterating the named entities based on their origin. (Sherif and Kondrak, 2007) use the Viterbi based monotone search algorithm for searching possible candidate sub-string transliterations. (Malik, 2006) solved some special cases of transliteration for Punjabi using a set of transliteration rules.

In the recent years Statistical Machine Translation (SMT) systems (Brown et al., 1990), (Yamada and Knight, 2001), (Chiang, 2005), (Charniak et al., 2003) have been in focus. It is easy to develop a MT system for a new pair of language using an existing SMT system and a parallel corpora. It isn't a surprise to see SMT being attractive in terms of less human labour as compared to other traditional systems. These SMT systems have also become popular in the transliteration field (Finch and Sumita, 2008), (Finch and Sumita, 2009), (Rama and Gali, 2009). (Finch and Sumita, 2008) use a bi-directional decoder whereas (Finch and Sumita, 2009) use a machine translation system comprising of two phrase-based decoders. The first decoder generated from first token of the target to the last. The second decoder generated the target from last to first. (Rama and Gali, 2009) modeled the phrase-based SMT system using minimum error rate training (MERT) for learning model weights.

In this paper we present a phrase-based machine transliteration technique with simple heuristics for transliterating named entities of English-Hindi pair using small amount of training and development data. The structure of our paper is as follows. Section 2 describes the modeling of translation problem to transliteration. Modeling of the parameters and the heuristics are presented in Section 3. Section 4 and 5 we give a brief description about the data-set and error-analysis. Finally we conclude in Section 6.

2 Modeling Approach

Transliteration can be viewed as a task of character-level machine translation process. Both the problems involve transformation of source tokens in one language to target tokens in another language.
Transliteration differs from machine translation in two ways (Finch and Sumita, 2009):

1. Reordering of the target tokens is generally

Proceedings of the 2010 Named Entities Workshop, ACL 2010, pages 81–84,
Uppsala, Sweden, 16 July 2010.

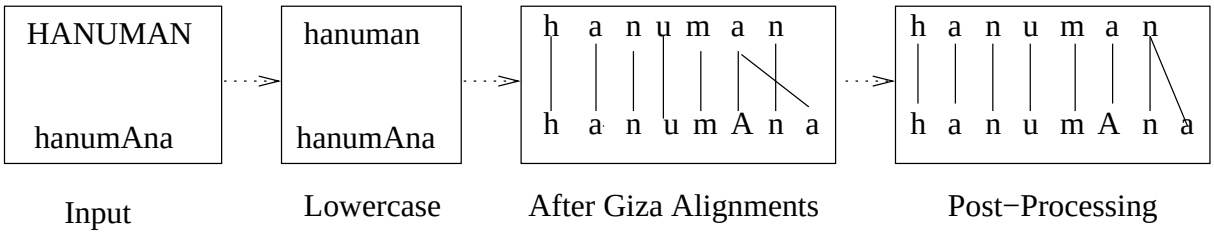

Figure 1: English-Hindi transliteration example through our system(To represent Hindi font roman script is used)

abscent in transliteration.

2. Number of token types (vocabulary) in the data is relatively very less and finite as compared to the translation data.

The work in this paper is related to the work of (Rama and Gali, 2009) who also use SMT directly to transliterate. We can model the translation problem to transliteration problem by replacing words with characters. So instead of sentences let us assume a given word is represented as a sequence of characters of the source language F=f_1,f_2,f_3,...f_n which needs to be transcribed as a sequence of characters in the target language E=e_1,e_2,e_3,...e_m. [1]

The best possible target language sequence of characters among the possible candidate characters can be represented as:

E_{best} = Argmax$_E$ P(E|F)

The above equation can be represented in terms of noisy channel model using Bayes Rule:

E_{best} = Argmax$_E$ P(F|E) $*$ P(E)

Here P(F|E) represents the transcription model where as P(E) represents the language model i.e the character n-gram of the target language. The above equation returns the best possible output sequence of characters for the given sequence of characters F.

We used some heuristics on top of Moses tool kit, which is a publicly available tool provided by (Hoang et al., 2007).

[1] F,E is used to name source and target language sequences as used in conventional machine translation notations

3 Method

3.1 Pre-processing

Firstly the data on the English side is converted to lowercase to reduce data sparsity. Each character of the words in the training and development data are separated with spaces. We also came across multi-word sequences which posed a challenge for our approach. We segmented the multi-words into separate words, such that they would be transliterated as different words.

3.2 Alignment and Post Processing

Parallel word lists are given to GIZA++ for character alignments. We observed *grow-diag-final-and* as the best alignment heuristic. From the differences mentioned above between transliteration and translation we came up with some simple heuristics to do post processing on the GIZA++ alignments.

1. As reordering of the target tokens is not allowed in transliteration. Crossing of the arcs during the alignments are removed.
   As shown in Fig 1. above.
   The second A → a is removed as it was crossing the arcs.

2. If the target character is aligned to *NULL* character on the source side then the *NULL* is removed, and the target language character is aligned to the source character aligned to previous target character.

   From Fig 1.

   n → n
   NULL → a

   to

n → na

3.3 Training and Parameter Tuning

The language models and translation models were built on the combined training and the development data. But the learning of log-linear weights during the MERT step is done using development data separately. It is obvious that the system would perform better if it was trained on the combined data. 8-gram language model and a maximum phrase length of 7 is used during training.

The transliteration systems were modeled using the minimum error rate training procedure introduced by (Och, 2003). We used BLUE score as a evaluation metric for our convenience during tuning. BLUE score is commonly used to evaluate machine translation systems and it is a function of geometric mean of n-gram precision. It was observed that improvement of the BLUE score also showed improvements in ACC.

4 Experiments and Results

Training data of 9975 words is used to build the system models, while the development data of 1974 words is used for tuning the log-linear weights for the translation engines. Our accuracies on test-data are reported in Table 1. Due to time constraints we couldn't focus on multiple correct answers in the training data, we picked just the first one for our training. Some of the translation features like word penalty, phrase penalty, reorder parameters don't play any role in transliteration process hence we didn't include them.

Before the release of the test-data we tested the system without tuning i.e. default weights were used on the development data. Later once the test-data was released the system was tuned on the development data to model the weights. We evaluated our system on ACC which accounts for Word Accuracy for top-1, Mean F-score, Mean Reciprocal Rank (MRR).

Table 1: Evaluation on Test Data

Measure	**Result**
ACC	0.381
Mean F-score	0.860
MRR	0.403
MAP_{ref}	0.381

5 Error Analysis

From the reference corpora we examined that majority of the errors were due to foreign origin words. As the phonetic transcription of these words is different from the other words. We also observed from error analysis that the correct target sequence of characters were occurring at lower rank in the 20-best list. We would like to see how different ranking mechanisms like SVM re-rank etc would help in boosting the correct accuracies of the system.

6 Conclusion

In this paper we show that the usage of some heuristics on top of popular phrase-based machine translation works well for the task of transliteration. First the source and target characters are aligned using GIZA++. Then some heuristics are used to modify the alignments. These modified alignments are used during estimation of the weights during minimum error rate training (MERT). Finally the Hindi characters are decoded using the beam-search based decoder. We also produced the 20-best outputs using the n-best list provided by moses toolkit. It is very interesting to see how simple heuristics helped in performing better than other systems.

References

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.

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *MT Summit IX. Intl. Assoc. for Machine Translation*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *In ACL*, pages 263–270.

Andrew Finch and Eiichiro Sumita. 2008. Phrase-based machine transliteration. In *In Proc. 3rd Int'l. Joint Conf NLP, volume 1*.

Andrew Finch and Eiichiro Sumita. 2009. Transliteration by bidirectional statistical machine translation. In *NEWS '09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 52–56, Morristown, NJ, USA. Association for Computational Linguistics.

Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, and Ondej Bojar. 2007. Moses: Open source toolkit for statistical machine translation. pages 177–180.

M. G. Abbas Malik. 2006. Punjabi machine transliteration. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1137–1144, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Taraka Rama and Karthik Gali. 2009. Modeling machine transliteration as a phrase based statistical machine translation problem. In *NEWS '09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 124–127, Morristown, NJ, USA. Association for Computational Linguistics.

Tarek Sherif and Grzegorz Kondrak. 2007. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 944–951, Prague, Czech Republic, June. Association for Computational Linguistics.

Praneeth Shishtla, V. Surya Ganesh, Sethuramalingam Subramaniam, and Vasudeva Varma. 2009. A language-independent transliteration schema using character aligned models at news 2009. In *NEWS '09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 40–43, Morristown, NJ, USA. Association for Computational Linguistics.

Harshit Surana and Anil Kumar Singh. 2009. *Digitizing The Legacy of Indian Languages*. ICFAI Books, Hyderabad.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. pages 523–530.