Babel: A Machine Transliteration Workbench

A. Kumaran Tobias Kellner
Multilingual Systems Research
Microsoft Research India
Bangalore, INDIA
a.kumaran@microsoft.com

Categories and Subject Descriptors: H.3.3 [Information Search & Retrieval]: Machine Transliteration, CLIR

General Terms: Algorithms

Keywords: Machine Transliteration, Cross-language Information Retrieval

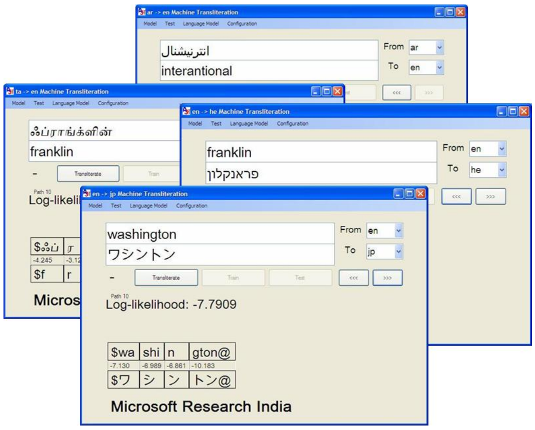
1. INTRODUCTION

Machine Transliteration deals with the conversion of text strings from one orthography to another, while preserving the phonetics of the strings in the two languages. Transliteration is an important problem in machine translation or cross-lingual information retrieval, as most proper names and generic iconic terms are out-of-vocabulary words, and therefore need to be transliterated. In this demo, we present BABEL, a transliteration workbench, with generic statistical learning algorithms and a scripting engine to model the transliteration process. We demonstrate quick assembly of necessary components – algorithmic modules and training scripts – for systematic experimentation of transliteration tasks in a given pair of languages.

2. TRANSLITERATION FRAMEWORK

The transliteration problem is modeled as a *noisy channel* in BABEL, as in the popular IBM Source-Channel models [1, 2, 3] for machine translation, with two specific changes: first, using graphemes instead of words, and, second, using monotonic alignment algorithms on training data. We bootstrap with an initial training using alignment probabilities estimated by the matching prefixes and suffixes of the paired strings in the training data, using an expectation maximization approach. Subsequently, we use viterbi algorithm to find the optimal alignments and iteratively refine the estimated model.

BABEL allows different algorithmic modules to be added, and currently includes a scripting engine to model the transliteration tasks. The transliteration quality measures are computed on top-n probable transliteration returned by BABEL, with respect to parameters, such as, the training data size, exact vs fuzzy matching, etc. In addition, user defined pre- or post-processing routines may be integrated easily, for any language-specific tasks. A simple front-end is provided for any online transliteration tasks. In the figure, we show a sample output of a transliteration task, from English to Arabic. The results highlight several trends, such as the quality of transliteration with respect to the training size, the effective size of potential transliteration for a resonable recall, the effect of allowing *fuzzy* matches to the transliteration tasks, etc.



	Exact matches			Fuzzy Matches		
Train Size	top 1	top 5	top 10	top 1	top 5	top 10
500	0.271	0.446	0.503	0.479	0.762	0.729
1,000	0.298	0.493	0.555	0.514	0.814	0.814
5,000	0.341	0.541	0.601	0.556	0.853	0.878
10,000	0.352	0.565	0.613	0.570	0.882	0.890
20,000	0.353	0.568	0.632	0.573	0.889	0.898

3. CONCLUSION

In this demo, we show BABEL, a generic workbench for experimenting with transliteration tasks, in a modular manner. In such a framework, an effective transliteration system between a given pair of languages may be put together quickly by existing reusable components.

References

[1] Brown, F. B. *et al.* The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 1993.
[2] Al-Onaizan, Y. and Knight, K. Machine transliteration of names in Arabic text. *Comp. Approaches for Semitic Languages*, 2002.
[3] Haizhou, L., Min, Z. and Jian, S. A joint source-channel model for machine transliteration. *42nd Meeting of ACL*, 2004.