Romanization – An Untapped Resource for Out-of-Vocabulary Machine Translation for CLIR

Fredric Gey
University of California, Berkeley
UC Data Archive & Technical Assistance
Berkeley, CA 94720-5100
510-643-1298

gey@berkeley.edu

ABSTRACT

In Cross-Language Information Retrieval (CLIR), the most continuing problem in query translation is the occurrence of out-of-vocabulary (OOV) terms which are not found in the resources available for machine translation (MT), e.g dictionaries, etc. This usually occurs when new named entities appear in news or other articles which have not been entered into the resource. Often these named entities have been phonetically rendered into the target language, usually from English. Phonetic back-transliteration can be achieved in a number of ways. One of these, which has been under-utilized for MT is Romanization, or rule-based transliteration of foreign typescript into the Latin alphabet. We argue that Romanization, coupled with approximate string matching, can become a new resource for approaching the OOV problem

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *abstracting methods, linguistic processing*

General Terms

Experimentation

Keywords

Machine Translation, Romanization Cross-Language Information Retrieval

1. INTRODUCTION

Successful cross-language information retrieval requires, at a minimum, the query (or document) in one language be translated correctly into the other language. This may be done using formal bilingual dictionaries or bilingual lexicons created statistically from aligned parallel corpora. But sometimes these resources have limited coverage with respect to current events, especially named entities such as new people or obscure places have appeared in news stories and their translation has yet to emerge within parallel corpora or enter into formal dictionaries. In addition, a plethora of name variants also confuse the issue of named entity recognition. Steinberger and Pouliquen (2007) discuss these issues in detail when dealing with multilingual news summarization. For non-Latin scripts, this becomes particularly problematic because the user of western scripted languages (such as in USA, England, and most of Europe) cannot guess phonetically what the name might be in his/her native language, even if the word or phrase was borrowed from English in the first place. In many cases, borrowed words enter the language as a phonetic rendering, or transliteration or the original language word. For example, the Japanese word **コンピュータ** (computer). Knight and Graehl (1997) jump-started transliteration research, particularly for Japanese-English by developing a finite state machine for phonetic recognition between the two languages. The phonetic transliteration of the above Japanese is 'konpyuutaa'.

There is, however, an alternative to phonetic transliteration, and that is Romanization, or rule-based rendering of a foreign script into the Latin alphabet. Romanization has been around for a long time. For Japanese, the Hepburn Romanization system was first presented in 1887. The Hepburn Romanization for the Japanese 'computer' above is 'kompyuta'. The Hepburn system is widely enough known that a PERL module for Hepburn is available from the CPAN archive.

In addition to Hepburn, there has been a long practice by the USA Library of Congress to Romanize foreign scripts when cataloging the titles of books written in foreign languages. Figure 1 presents a list of about 55 languages for which the Library of Congress has published Romanization tables. Note that major Indian subcontinent languages of Bengali, Gujarati, Hindi, Marathi, Punjabi, Tamil, Telugu and Urdu are included. For example, the Cyrillic Клинтон or the Greek Κλίντον can easily be Romanized to Klinton. For Russian and Greek, the transformation is usually reversible. For the major Indian language, Hindi, it is easily possible to find the translation for Clinton, but for the south Indian language of Tamil, translations are less easily found. Yet Tamil is a rather regular phonetic language and foreign names are often transliterated when news stories are written in Tamil (although one reviewer has remarked that Tamil has phonetic ambiguities not found in other Indian languages). Figure 2 is a translated news story in Tamil, when the main names (Presidents Clinton and Yeltsin) are Romanized.

SIGIR Workshop on Information Access in a Multilingual World July 23, 2009 Boston, Massachusetts USA.

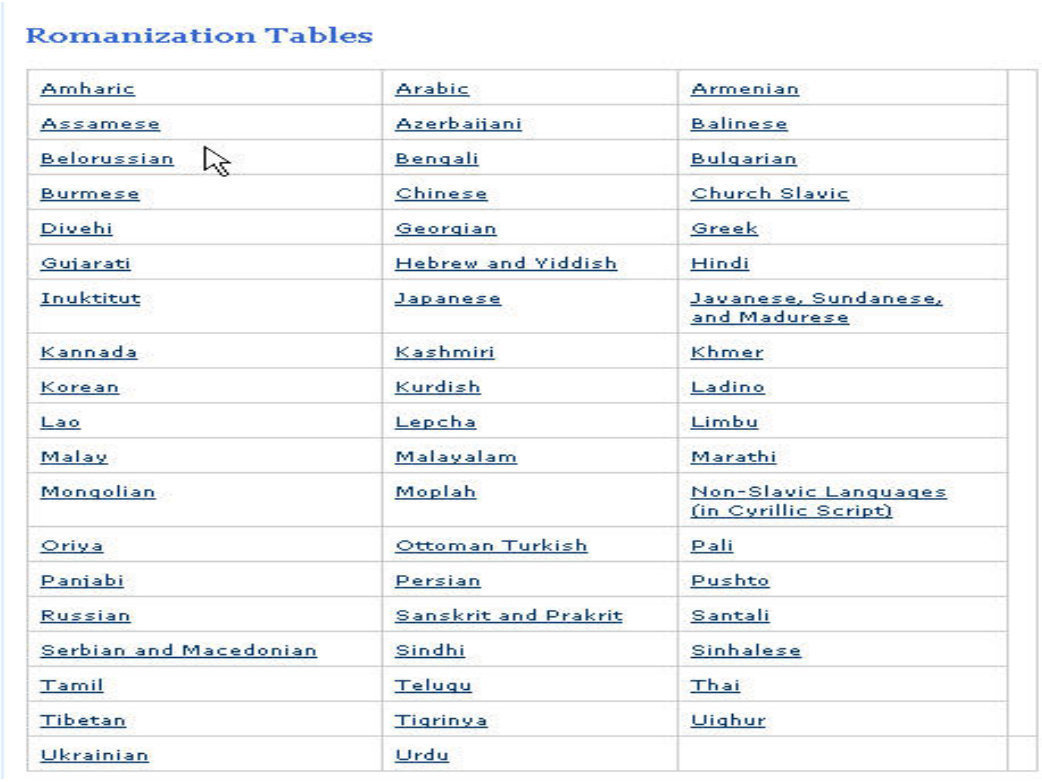

Figure 1: Library of Congress Romanization Language List

2. TRANSLITERATION/ROMANIZATION

In the sweep of methods for recognition of out-of-vocabulary terms between languages and for automatic phonetic recognition of borrowed terms, Romanization has become a much-neglected stepchild. However phonetic transliteration (and back-transliteration from the target language to the source language) requires large training sets for machine learning to take place. For less-commonly taught languages, such as, for example, some Indian subcontinent languages, such training sets may not be available. Romanization, on the other hand, requires that rules for alphabet mapping be already in place, developed by experts in both target and source languages. However, once the target language word has been rendered into its Latin alphabet equivalent, we still have the problem of matching it to its translation in the source language. So we ask: Is there a place for Romanization in CLIR? And how can it be exploited? The key is the examination of approximate string matching methods to find the correspondences between words of the target and source languages.

3. APPROXIMATE STRING MATCHING

Once one has Romanized a section of non-English text containing OOV, the task remains to find its English word equivalents. The natural way to do this is using approximate string matching techniques. The most well-known technique is edit distance, the number of insertions, deletions and interchanges necessary to transform one string to its matching string. For example, the edit distance between computer and kompyuta (コンピュータ) is 5. Easier to comprehend is between English and German, where the Edit distance between fish (E) and fisch (DE) is 1. However, the edit distance between fish(E) and frisch (DE) is 2, whereas between the correct translations fresh (E) and frisch (DE) is also 2. Thus Martin Braschler of the University of Zurich has remarked, "Edit distance is a terrible cross-lingual matching method." Approximate string matching has a lengthy history for both fast file search techniques as well as finding matches of minor word translation variants across languages. Q-grams, as proposed by Ukkonen (1992) counts the number of substrings of size 'q' in common between the strings being matched. A variant of q-grams are targeted s-grams where q is of size 2 and *skips* are allowed to omit letters from the match. Pirkkola and others (2003) used this technique for cross-language search between Finnish, Swedish and German. Using s-gram skips solves the fish – fisch differential above. An alternative approach, which has been around for some time, is the Phonix method of Gadd (1998) which applies a series of transformations to letters (for example, c → k, in many cases, e.g. Clinton → Klinton) and shrinks out the vowels, (Clinton → Klntn). If we apply this transformation to the English Japanese above, we have computer → kmptr and compyuta →kmpt. The original version of Phonix only kept the leading four resulting characters, and would result in an exact match. Zobel and Dart (1995) did an extensive examination of approximate matching methods for digital libraries and their second paper (1996) proposed an improved Phonix method they titled Phonix-plus which did not truncate to 4 characters, but instead rewarded matches at the beginning. They combined this with edit distance for the Zobel-Dart matching algorithm.

4. SUMMARY AND POSITION

The current fashion for utilizing statistical machine learning as the solution to all problems in machine translation has led to the neglect of rule-based methods which, this paper argues, are both well-developed and could complement statistical approaches. Romanization would work especially well for non-Latin scripted languages for which training corpora are limited. The approach has two steps: 1) Romanization of the script using well-documented methods, followed by 2) Approximate string matching between Romanized words in the target language and possible translation candidates in the source language.

5. ACKNOWLEDGMENTS

Much of this work was originally done while the author was a visiting researcher at the National Institute of Infomatics (NII) in Tokyo in the summer of 2007 supported by a grant from NII.

6. REFERENCES

[1] Knight, K and J Graehl (1997), Machine Transliteration, Association for Computational Linguistics (1997): http://www.ala.org/ala/acrl/acrlpubs/crljournal/collegeresearch.cfm

[2] T. Gadd (1988), Fisching fore Werds: Phonetic Retrieval of Written Text in Information Systems, ***Program, 22(3):222–237, 1988***

[3] R. Steinberger and B. Pouliquen (2007). Cross-lingual named entity recognition. Special issue of Lingvistic Investigationes, 30:135–162, 2007.

[4] J. Zobel and P. Dart (1995). Finding approximate matches in large lexicons. Softw. Pract. Exper., 25(3):331–345, 1995.

[5] J. Zobel and P. Dart (1996). Phonetic string matching: lessons from information retrieval. In SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, pages 166–172, New York, NY, USA, 1996. ACM Press.

[6] A. Pirkola, J. Toivonen, H. Keskustalo, K. Visala, and K. Jarvelin (2003). Fuzzy translation of cross-lingual spelling variants. In SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 345–352, New York, NY, USA, 2003. ACM Press.

[7] E. Ukkonen (1992). Approximate string-matching with q-grams and maximal matches. Theoretical Computer Science 92 (1992), 191-210

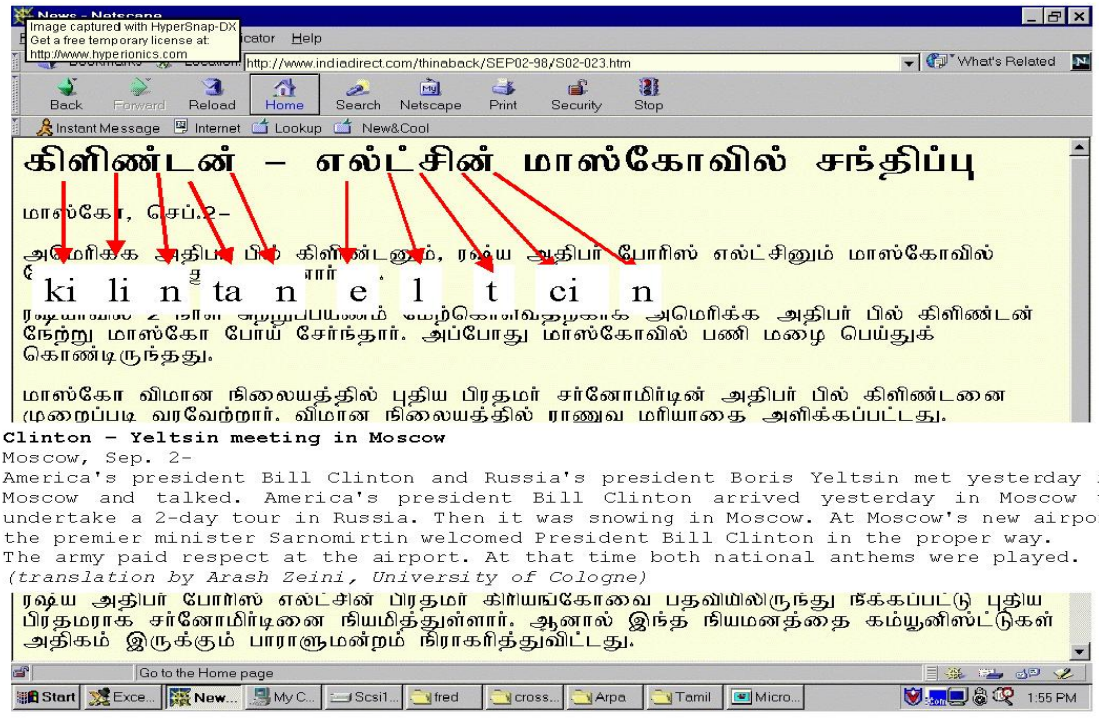

Figure 2: News story in the Tamil language of Clinton-Yeltsin Meeting, showing name Romanization (phonetic transliteration according to software from the University of Cologne, Germany)