Extracting Named Entity Translingual Equivalence with Limited Resources

FEI HUANG, STEPHAN VOGEL, AND ALEX WAIBEL
Carnegie Mellon University

In this article we present an automatic approach to extracting Hindi-English (H-E) Named Entity (NE) translingual equivalences from bilingual parallel corpora. In the absence of a Hindi NE tagger or H-E translation dictionary, this approach adapts a Chinese-English (C-E) surface string transliteration model for H-E NE extraction. The model is initially trained using automatically extracted C-E NE pairs, then iteratively updated based on newly extracted H-E NE pairs. For each English person and location NE in each sentence pair, this approach searches for its Hindi correspondence with minimum transliteration cost and constructs an H-E NE list from the bilingual corpus. Experiments show that this approach extracted 1000 H-E NE pairs with a precision of 91.8%.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Machine translation*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – Dictionaries; I.5.4 [**Pattern Recognition**]: Application – *Text processing*

General Terms: Algorithms, Language, Performance

Additional Key Words and Phrases: Named entity translation, transliteration, machine translation, information extraction

1. INTRODUCTION

Translingual equivalence refers to semantically equivalent expressions from different languages. Identifying translingual equivalence of named entities (NE), including named persons, locations, and organizations, is both semantically important and technically challenging. This is because NE translation involves both semantic translation and phonetic transliteration, and the frequent occurrence of Out-Of-Vocabulary words[1] in NEs further complicates the matter. Some approaches to named entity translation, such as bilingual dictionary lookup, word/character semantic translation or phonetic transliteration, have been explored in the past few years [Knight et al. 1997; Meng et al. 2001; Al-Onaizan et al. 2002; Huang et al. 2003]. However, challenges are also encountered in these approaches: for example, precompiled bilingual NE lists, although showing high translation precision, often suffer from low coverage for new documents, and word/character-based translation or transliteration sometimes fails to yield quality results due to lack of contextual information. For instance, "风 陵 渡/Fenglingdu", a Chinese location name, cannot be found in a dictionary with 50k entries provided by LDC, and it is also inappropriate to adopt the character-by-character semantic translation, which is "wind tomb cross."

Authors' address: Carnegie Mellon University, Pittsburgh, PA.

[1] Out-of-Vocabulary words refer to words not included in a precompiled translation dictionary.

One possible solution is to automatically extract and align named entity translingual equivalences from a parallel corpus, where named entities have been manually or automatically annotated [Huang et al. 2003]. This NE alignment strategy incorporates multiple features, such as transliteration, translation, and tagging features, and achieves an F score, the combined measure of precision and recall, of 81% on an automatically-tagged Chinese-English (C-E) corpus and 93% on a manually-annotated one. However, this approach cannot be applied directly to the language pair Hindi-English (H-E), as neither an H-E translation lexicon nor a Hindi NE tagger were available.

A second possibility is to automatically tag NEs on the English side and to use standard word alignment models [Brown et al.1993] to project NEs from English to Hindi. However, not only does this directly rely on the quality of the word alignment, but more importantly, in the context of machine translation, it gives no additional or more reliable information than using phrase-to-phrase translation pairs extracted from the bilingual corpus [Vogel et al. 2003].

Considering that person and location names are often phonetically translated and their written forms resemble their pronunciations, it is possible to discover NE translation pairs through their written forms by way of surface string transliteration. Compared with the traditional phoneme transliteration method, surface string transliteration does not require a pronunciation lexicon, which is an advantage, especially for rare names. For nonLatin languages like Chinese and Hindi, indirect surface string transliteration is feasible through a romanization process that maps each character into Latin letter(s) with similar pronunciation. For example, the Hindi word कलकत्ता is romanized as “kalakattaa,” which is the translation of “Calcutta.” In this article we propose an automatic approach to learn the transliteration model between romanized Hindi and English letters, and apply this model to extract H-E NE pairs from parallel corpora based on their similarities in written form, without the need for a Hindi NE tagger or H-E translation dictionary. The H-E transliteration model can either be learned directly from the parallel corpus, or adapted from an already learned C-E model. Due to the noise in the H-E parallel corpus and the high quality of the C-E alignment model baseline, the adapted model outperforms the directly learned model, as demonstrated by experiments in Section 4.

This article is structured as follows: in Section 2 we describe the C-E NE transliteration model; in Section 3 we demonstrate how to iteratively adapt the transliteration model and extract H-E NE translations; in Section 4 we present some experimental results; and we draw some conclusions in the last section.

2. NAMED ENTITY TRANSLITERATION MODEL

As mentioned above, romanization is required for both Hindi and Chinese. Pinyin, the romanized form for Chinese characters, provides a much smaller alphabet size, which alleviates data sparseness, and the similar alphabets shared by pinyin and English enable dynamic programming (DP)-based string matching.

A bilingual transliterated name list is usually required to train a model that maps pinyin syllables to English strings (e.g., “萨/sa 拉/la 热/re 窝/wo” to “Sarajevo”). To acquire such an NE list, we propose an unsupervised learning approach in which NE pairs are automatically extracted from a large bilingual dictionary. DP-based string matching is iteratively applied in order to estimate the transliteration probability from pinyin to English letter sequences.

To extract NE pairs from a given bilingual dictionary D, we want to find the NE pair $(f_{ne}*, e_{ne}*)$ with the highest joint probability,

$$(f_{ne}*, e_{ne}*) = \arg\max_{(f,e)\subset D} P_{ne}(f,e) = \arg\max_{(f,e)\subset D} P_{ne}(f) P_{ne}(e \mid f),$$

Here, $P_{ne}(f)$ is the probability of generating the character sequence of the Chinese NE, which can be computed directly from a character language model for Chinese NEs. The estimate for $P_{ne}(e \mid f)$, the probability of *transliterating* the Chinese NE f into an English NE e, is as follows:

Suppose f has m characters. For $i = 1,2,..m$, suppose character f_i is independently transliterated into an English letter string e_i through its pinyin syllable y_i. Given that mappings from Chinese characters to their pinyin syllables are mostly deterministic, i.e., $p(y_i \mid f_i) \approx 1$, we have

$$P_{ne}(e \mid f) = \prod_{i=1}^{m} p(e_i \mid f_i) = \prod_{i=1}^{m} p(e_i \mid y_i) p(y_i \mid f_i) \approx \prod_{i=1}^{m} p(e_i \mid y_i).$$

Suppose y_i is composed of m_i letters, and for $j = 1,2,...m_i$, the pinyin letter $y_{i,j}$ is aligned with the English letter $e_{i,k}$, where the alignment is represented as $k = a_j$. With the independence assumption about letter transliteration, we obtain

$$P_{ne}(e \mid f) \approx \prod_{i=1}^{m} p(e_i \mid y_i) = \prod_{i=1}^{m} \prod_{j=1}^{m_i} p(e_{i,k} \mid y_{i,j}).$$

Following the derivation of the transliteration model, the next steps are to identify letter-to-letter alignment and to train the transliteration model and language model.

Dynamic programming has been successfully applied in searching for the “optimal” alignment path between two strings, where “optimal” means the minimum accumulated editing cost between aligned word/letter pairs. Here the cost is usually defined as 0 if the aligned words/letters are the same, or 1 in case of an insertion, deletion, or substitution error. However, this binary cost function is not appropriate for pronunciation-based transliteration, since the phonetic similarity is more important than the orthographic one. Hence, the alignment cost between letters with similar pronunciations (e.g., “c” and “k” or “p” and “b”) should be smaller. We take the negative logarithm of the letter transliteration probability as the matching cost, where the transliteration probabilities are computed based on their alignment frequency. However, the alignment requires the alignment cost function. To resolve this model interdependence, the binary cost function is initially applied to the DP string alignment. Bilingual NE pairs are extracted from the dictionary according to their alignment cost. Based on this initial imperfect name list, the letter transliteration model and character language model are trained, and used for the NE joint probability estimate. In the following iterations, the alignment cost function as well as the transliteration probability are updated, NE pairs are reselected according to their joint probabilities, and transliteration and language models are retrained using the cleaner NE list.

3. ADAPTING A TRANSLITERATION MODEL FOR HINDI NE TRANSLATION

Considering the differences in language pairs and encoding schemes, the following problems must be tackled when applying the above C-E transliteration model to H-E:

- The Hindi sentences are encoded as Devanagari characters. A romanization tool based on code table lookup is applied to convert Devanagari characters into roman letters.
- The transliteration model was originally trained from C-E NE pairs. When applying it to H-E NE transliteration, model adaptation is required due to the different alignment patterns in H-E. In practice, the C-E transliteration model was first applied to compute the H-E transliteration cost, resulting in a list of NE pairs with minimum alignment cost. From these imperfect NE pairs, the H-E transliteration model was re-trained and applied in the next round of NE pair extractions. After each iteration, the transliteration model got updated according to the model described in Section 2.
- Given that no Hindi NE tagger is available, it is impossible to extract H-E NE pairs by "monolingual NE detection followed by bilingual NE alignment." On the other hand, Hindi NEs can be detected by projecting English NEs cross-lingually according to their phonetic similarity or transliteration cost, where the English NEs can be automatically detected using an HMM-based NE tagger [Bikel et. al.1997].

The following steps describe the procedure for H-E NE pair extraction:

1) Convert UTF-8 encoded Hindi Devanagari characters into roman letters;
2) Use the English NE tagger to detect NEs. For each detected NE, find the romanized Hindi word sequences in the Hindi counterpart, such that the transliteration cost between the English NE and the Hindi word sequence is minimal. The romanized words are then mapped back to their corresponding Devanagari Hindi words;
3) Sort the H-E NE pairs according to their transliteration cost weighted by alignment frequencies, and remove those with high transliteration cost;
4) Run the current string alignment model on the extracted H-E NE pairs, update the letter transliteration cost based on the new alignment frequency;
5) Repeat steps 2 to 4 until convergence is reached or over-fitting is noticed.

Figure 1 further illustrates how the transliteration model is initially trained for C-E and then adapted for H-E NE translation extraction.

Notice that, although this approach searches for an acoustically similar Hindi name for each detected English person and location name from the sentence-aligned bilingual

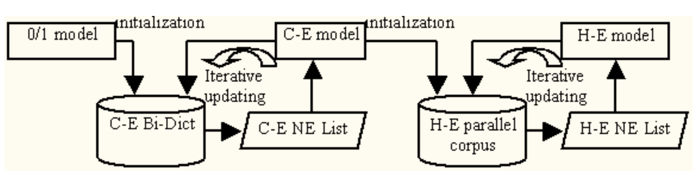

Fig. 1. Iterative training and adaptation of the transliteration model.

corpus, noisy parallel data such as comparable corpora can be exploited as well, as long as the Hindi monolingual data contains the corresponding Hindi names.

4. EXPERIMENTS

In our experiments, the parallel corpus is the *India Today* news corpus, with 10,096 sentence pairs, 223K Hindi words, and 215K English words. Automatic NE tagging resulted in 2,451 location NEs and 1,614 person NEs, giving a list of 1,172 unique names.

Several transliteration alignment models to extract bilingual NE pairs from the above corpus have been studied, including 0/1 binary cost, unadapted C-E alignment cost, and adapted H-E alignment cost (after the 1st and 2nd iterations). For each alignment model, 220 out of the extracted top 1000 NE pairs were randomly selected and evaluated by a native Hindi speaker. Table I shows the translation precision of the different NE lists. We can see that adapting the generic alignment model to H-E transliteration improves translation precision significantly from 79% to 91%. Further adaptation within the language pair still yields small but noticeable improvements. Another set of experiments were carried out to compare the adapted C-E alignment model with the directly learned H-E alignment model,. The H-E model was initialized with the binary cost, then re-trained iteratively, as described in Section 2. Table II gives the NE translation precisions after each iteration. A significant increase in the initial iterations was followed by a slight decrease in subsequent iterations. Still, the best result (88.2% in iteration 3) was not as good as the best result (91.8%) when using the C-E model as initialization. The reason is that the C-E model, to some extent, already captures letter pronunciation similarities, and so it will provide more reliable baseline NE pairs for further retraining. Some extracted H-E NE pairs are also shown in Figure 2, together with their transliteration cost (the lower the weighted cost, the more accurate the transliteration). We can find similar spelling patterns between aligned romanized Hindi NE and English NE, for both correct and incorrect (marked with “*”) NE translation pairs. Since for each detected English NE the proposed approach always searches for the best-matching Hindi NE, its recall rate depends mostly on that of the English NE detection. The bilingual NE list was shared within the TIDES Surprise Language Exercise community.

Table I.. H-E NE Pairs Translation Precision Under Different Alignment Models

Alignment models	0/1 binary	C-E	1st iter. H-E	2nd iter. H-E
Precision	79.1%	86.3%	90.9%	**91.8%**

Table II. Iterative Translation Precisions Starting With Binary Alignment Model

Iteration	0	1	2	3	4	5
Precision	79.1%	85.9%	86.8%	**88.2%**	87.2%	86.8%

Format: Weighted Cost # Devanagari Hindi NE # Romanized Hindi NE # English NE

-4.013 # पाकिस्तान # paakistaana # pakistan 0.619 # मार्गरेट अल्वा # maargareta alvaa # margaret alva
-0.125 # मुशर्रफ # musharrapha # musharraf 3.205 # गंजम जिले # ga~jama jile # ganjam district
-0.088 # कलकत्ता # kalakattaa # calcutta 3.253 # और मुंबई # aura mu~baii # sullied mumbai (*)

Fig. 2. Extracted Hindi-English NE pairs.

5. SUMMARY

We presented an automatic approach to extract Hindi-English NE pairs from a parallel corpus using limited resources. This approach adapts and iteratively updates a Chinese-English surface string transliteration model to Hindi-English NE extraction. For each English person and location NE in each sentence pair, this approach searches for its Hindi correspondence with minimum transliteration cost, and constructs a Hindi-English NE list from the bilingual corpus. Experiments show that this approach extracted 1000 Hindi-English NE pairs with a precision of 91.8%.

ACKNOWLEDGMENTS

Many thanks to Andrew Hardie of Lancaster University, who developed the Hindi romanization tool. We would also like to thank BBN for providing us with their English named entity tagging software, IdentiFinder™.

REFERENCES

AL-ONAIZAN, Y. AND KNIGHT, K. 2002. Translating named entity using bilingual and monolingual resources. In *The 40th Proceedings of the Association for Computational Linguistics Conference* (Philadelphia, PA, July 2002).

BIKEL, D., MILLER, S., SCHWARZ, R., AND WEISCHEDEL, R. 1997. Nymble: A high-performance learning name-finder. In *Proceedings of the Conference on Applied Natural Language Processing-97* (Washington DC, 1997), 194-201.

BROWN, P. F., DELLA PIETRA, S. A., DELLA PIETRA, V. J., AND MERCER, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguistics 19* (1993), 263-311.

HUANG, F., VOGEL, S., AND WAIBEL, A. 2003. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In *The 41st Proceedings of the Association for Computational Linguistics, Workshop on Multilingual and Mixed-Language Named Entity Recognition* (Sapporo, Japan, July 2003).

KNIGHT, K. AND GRAEHL, J. 1997. Machine transliteration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (Madrid, Spain, July 1997), 128-135.

MENG, H., LO, W. K., CHEN, B., AND TANG, K. 2001. Generating phonetic cognates to handle named entities in English-Chinese cross-language spoken document retrieval. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop* (Trento, Italy, Dec. 2001).

VOGEL, S., ZHANG, Y., HUANG, F., TRIBBLE, A., VENOGUPAL, A., ZHAO, B., AND WAIBEL, A. 2003. The CMU statistical machine translation system. In *Proceedings of the MT Summit IX Conference* (New Orlean, LA, Sept. 2003).

Received August 2003; revised September 2003; accepted October 2003