A Language-Independent Transliteration Schema Using Character Aligned Models At NEWS 2009

Praneeth Shishtla, Surya Ganesh V, Sethuramalingam Subramaniam, Vasudeva Varma
Language Technologies Research Centre,
IIIT-Hyderabad, India
praneethms@students.iiit.ac.in
{suryag,sethu}@research.iiit.ac.in, vv@iiit.ac.in

Abstract

In this paper we present a statistical transliteration technique that is language independent. This technique uses statistical alignment models and Conditional Random Fields (CRF). Statistical alignment models maximizes the probability of the observed (source, target) word pairs using the expectation maximization algorithm and then the character level alignments are set to maximum posterior predictions of the model. CRF has efficient training and decoding processes which is conditioned on both source and target languages and produces globally optimal solution.

1 Introduction

A significant portion of out-of-vocabulary (OOV) words in machine translation systems, information extraction and cross language retrieval models are named entities (NEs). If the languages are written in different scripts, these named entities must be transliterated. Transliteration is defined as the process of obtaining the phonetic translation of names across languages. A source language word can have more than one valid transliteration in the target language. In areas like Cross Language Information Retrieval (CLIR), it is important to generate all possible transliterations of a Named Entity.

Most current transliteration systems use a generative model for transliteration such as freely available GIZA++[1] (Och and Ney , 2000), an implementation of the IBM alignment models (Brown et al., 1993) and HMM alignment model. These systems use GIZA++ to get character level alignments from word aligned data. The transliteration system (Nasreen and Larkey , 2003) is built by counting up the alignments and converting the counts to conditional probabilities.

In this paper, we describe our participation in *NEWS 2009 Machine Transliteration Shared Task* (Li et al., 2009). We present a simple statistical, language independent technique which uses statistical alignment models and Conditional Random Fields (CRFs) (Hanna , 2004). Using this technique a desired number of transliterations are generated for a given word.

2 Previous work

One of the works on Transliteration is done by Arababi et al. (Arababi et. al., 1994). They model forward transliteration through a combination of neural net and expert systems. Work in the field of Indian Language CLIR was done by Jaleel and Larkey (Larkey et al., 2003). They did this based on their work in English-Arabic transliteration for CLIR (Nasreen and Larkey , 2003). Their approach was based on HMM using GIZA++ (Och and Ney , 2000). Prior work in Arabic-English transliteration for machine translation purpose was done by Arababi (Arbabi et al., 1994). They developed a hybrid neural network and knowledge-based system to generate multiple English spellings for Arabic person names. Knight and Graehl (Knight and Graehl , 1997) developed a five stage statistical model to do back transliteration, that is, recover the original English name from its transliteration into Japanese Katakana. Stalls and Knight (Stalls and Knight , 1998) adapted this approach for back transliteration from Arabic to English of English names. Al-Onaizan and Knight (Onaizan and Knight , 2002) have produced a simpler Arabic/English transliterator and evaluates how well their system can match a source spelling. Their work includes an

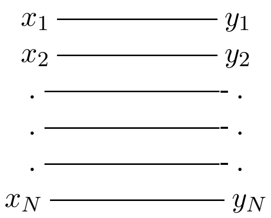
[1] http://www.fjoch.com/GIZA++.html

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 40–43,
Suntec, Singapore, 7 August 2009. ©2009 ACL and AFNLP

evaluation of the transliterations in terms of their reasonableness according to human judges. None of these studies measures their performance on a retrieval task or on other NLP tasks. Fujii and Ishikawa (Fujii and Ishikawa , 2001) describe a transliteration system for English-Japanese CLIR that requires some linguistic knowledge. They evaluate the effectiveness of their system on an English-Japanese CLIR task.

3 Problem Description

The problem can be stated formally as a sequence labeling problem from one language alphabet to other. Consider a source language word $x_1x_2..x_i..x_N$ where each x_i is treated as a word in the observation sequence. Let the equivalent target language orthography of the same word be $y_1y_2..y_i..y_N$ where each y_i is treated as a label in the label sequence. The task here is to generate a valid target language word (label sequence) for the source language word (observation sequence).



Here the valid target language alphabet (y_i) for a source language alphabet (x_i) in the input source language word may depend on various factors like

1. The source language alphabet in the input word.
2. The context (alphabets) surrounding source language alphabet (x_i) in the input word.
3. The context (alphabets) surrounding target language alphabet (y_i) in the desired output word.

4 Transliteration using alignment models and CRF

Our approach for transliteration is divided into two phases. The first phase induces character alignments over a word-aligned bilingual corpus, and the second phase uses some statistics over the alignments to transliterate the source language word and generate the desired number of target language words. The selected statistical model for transliteration is based on a combination of statistical alignment models and CRF. The alignment models maximize the probability of the observed (source, target) word pairs using the expectation maximization algorithm. After the maximization process is complete, the character level alignments are set to maximum posterior predictions of the model. This alignment is used to get character level alignment of source and target language words. From the character level alignment obtained we compare each source language character to a word and its corresponding target language character to a label. Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting sequential data. We use CRF to generate target language word (similar to label sequence) from source language word (similar to observation sequence). CRFs are undirected graphical models which define a conditional distribution over a label sequence given an observation sequence. We define CRFs as conditional probability distributions $P(Y|X)$ of target language words given source language words. The probability of a particular target language word Y given source language word X is the normalized product of potential functions each of the form

$$e^{(\sum_j \lambda_j t_j(Y_{i-1},Y_i,X,i))+(\sum_k \mu_k s_k(Y_i,X,i))}$$

where $t_j(Y_{i-1},Y_i,X,i)$ is a transition feature function of the entire source language word and the target language characters at positions i and $i-1$ in the target language word; $s_k(Y_i,X,i)$ is a state feature function of the target language word at position i and the source language word; and λ_j and μ_k are parameters to be estimated from training data.

$$F_j(Y,X) = \sum_{i=1}^{n} f_j(Y_{i-1},Y_i,X,i)$$

where each $f_j(Y_{i-1},Y_i,X,i)$ is either a state function $s(Y_{i-1},Y_i,X,i)$ or a transition function $t(Y_{i-1},Y_i,X,i)$. This allows the probability of a target language word Y given a source language word X to be written as

$$P(Y|X,\lambda) = (\frac{1}{Z(X)})e^{(\sum \lambda_j F_j(Y,X))}$$

$Z(X)$ is a normalization factor.

5 Our Transliteration system

The whole model has three important phases. Two of them are off-line processes and the other is a online process. The two off-line phases are preprocessing the parallel corpora and training the model using CRF++[2] (Lafferty et al., 2001). CRF++ is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs) for segmenting/labeling sequential data. The on-line phase involves generating desired number of target language transliterations (UTF-8 encoded) for the given English input word. In our case, the source is always an English word. The same system is used for every language pair which makes it a language independent. The target languages consist of Chinese, Hindi, Kannada Tamil and Russian words.

5.1 Preprocessing

The training file is converted into a format required by CRF++. The sequence of steps in preprocessing are

1. Both source and target language words were prefixed with a begin symbol B and suffixed with an end symbol E which correspond to start and end states. English words were converted to lower case.

2. The training words were segmented in to unigrams and the source-target word pairs were aligned using GIZA++ (IBM model1, HMM alignment model, IBM model3 and IBM model4).

3. The alignment consist of $NULLs$ on source language i.e., a target language unigram is aligned to $NULL$ on the source language. These $NULLs$ are problematic during online phase (as positions of $NULLs$ are unknown). So, these $NULLs$ are removed by appending the target language unigram to the unigram of its previous alignment. For example, the following alignment,

   $$k - K$$

   $$NULL - A$$

   transforms to -

   $$k - KA$$

   So, in the final alignment, the source side always contains unigrams and the target side might contain ngrams which depends on alphabet size of the languages. These three steps are performed to get the character level alignment for each source and target language training words.

4. This final alignment is transformed to training format as required by CRF++ to work. In the training format, a source language unigram aligned to a target language ngram is called a token. Each token must be represented in one line, with the columns separated by white space (spaces or tabular characters). Each token should have equal number of columns.

5.2 Training Phase

The preprocessing phase converts the corpus into CRF++ input file format. This file is used to train the CRF model. The training requires a template file which specifies the features to be selected by the model. The training is done using Limited memory Broyden-Fletcher-Goldfarb-Shannon method (L-BFGS) (Liu and Nocedal, 1989) which uses quasi-newton algorithm for large scale numerical optimization problem. We used English characters as features for our model and a window size of 5.

5.3 Transliteration

For a language pair, the list of English words that need to be transliterated is taken. These words are converted into CRF++ test file format and transliterated using the trained model which gives the top n probable English words. CRF++ uses forward Viterbi and backward A* search whose combination produces the exact n-best results. This process is repeated for all the five language pairs.

6 Results

In this section, we present the results of our participation in the NEWS-2009 shared task. We conducted our experiments on five language pairs namely English-Chinese (Li et al., 2004), English-{Hindi, Kannada, Tamil, Russian} (Kumaran and Kellner , 2007). As specified in *NEWS 2009 Machine Transliteration Shared Task* (Li et al., 2009), we submitted our standard runs on all the five language pairs. Table 1 shows the results of our system.

[2] http://crfpp.sourceforge.net/

Language Pair	Accuracy in top-1	Mean F-score	MRR	MAP_{ref}	MAP_{10}	MAP_{sys}
English-Tamil	0.406	0.894	0.542	0.399	0.193	0.193
English-Hindi	0.407	0.877	0.544	0.402	0.195	0.195
English-Russian	0.548	0.916	0.640	0.548	0.210	0.210
English-Chinese	0.493	0.804	0.600	0.493	0.192	0.192
English-Kannada	0.350	0.864	0.482	0.344	0.175	0.175

Table 1: Transliteration results for the language pairs

7 Conclusion

In this paper, we have described our transliteration system build on a discriminative model using CRF and statistical alignment models. As mentioned earlier, our system is language independent and works on any language pair provided parallel word lists are available for training in the particular language pair. The main advantage of our system is that we use no language-specific heuristics in any of our modules and hence it is extensible to any language-pair with least effort.

References

A. Kumaran, Tobias Kellner. 2007. *A generic framework for machine transliteration, Proc. of the 30th SIGIR.*

A. L. Berger. 1997. *The improved iterative scaling algorithm: A gentle introduction.*

Arbabi, M. and Fischthal, S. M. and Cheng, V. C. and Bart, E. 1994. *Algorithms for Arabic name transliteration, IBM Journal of Research And Development.*

Al-Onaizan Y, Knight K. 2002. *Machine translation of names in Arabic text. Proceedings of the ACL conference workshop on computational approaches to Semitic languages.*

Arababi Mansur, Scott M. Fischthal, Vincent C. Cheng, and Elizabeth Bar. 1994. *Algorithms for Arabic name transliteration. IBM Journal of research and Development.*

D. C. Liu and J. Nocedal. 1989. *On the limited memory BFGS method for large-scale optimization, Math. Programming 45 (1989), pp. 503–528.*

Fujii Atsushi and Tetsuya Ishikawa. 2001. *Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration. Computers and the Humanities, Vol.35, No.4, pp.389-420.*

H. M. Wallach. 2002. *Efficient training of conditional random fields. Masters thesis, University of Edinburgh.*

Hanna M. Wallach. 2004. *Conditional Random Fields: An Introduction.*

Haizhou Li, A Kumaran, Min Zhang, Vladimir Pervouchine. 2009. *Whitepaper of NEWS 2009 Machine Transliteration Shared Task. Proceedings of ACL-IJCNLP 2009 Named Entities Workshop (NEWS 2009), Singapore.*

Haizhou Li, A Kumaran, Vladimir Pervouchine, Min Zhang. 2009. *Report on NEWS 2009 Machine Transliteration Shared Task. Proceedings of ACL-IJCNLP 2009 Named Entities Workshop (NEWS 2009), Singapore.*

Haizhou Li, Min Zhang, Jian Su. 2004. *A joint source channel model for machine transliteration. Proc. of the 42nd ACL.*

J. Darroch and D. Ratcliff. 1972. *Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics, 43:14701480.*

J. Lafferty, A. McCallum, and F. Pereira. 2001. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. of ICML, pp.282-289.*

Knight Kevin and Graehl Jonathan. 1997. *Machine transliteration. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 128-135. Morgan Kaufmann.*

Larkey, Connell,AbdulJaleel. 2003. *Hindi CLIR in Thirty Days.*

Nasreen Abdul Jaleel and Leah S. Larkey. 2003. *Statistical Transliteration for English-Arabic Cross Language Information Retrieval.*

Och Franz Josef and Hermann Ney. 2000. *Improved Statistical Alignment Models. Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hong Kong, China.*

P. F. Brown, S. A. Della Pietra, and R. L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263-311.*

Phil Blunsom and Trevor Cohn. 2006. *Discriminative Word Alignment with Conditional Random Fields.*

Stalls Bonnie Glover and Kevin Knight. 1998. *Translating names and technical terms in Arabic text.*