Statistical Transliteration for Cross Langauge Information Retrieval using HMM alignment and CRF

Surya Ganesh, Sree Harsha
LTRC, IIIT
Hyderabad, India
suryag,sreeharshay@students.iiit.net

Prasad Pingali, Vasudeva Varma
LTRC, IIIT
Hyderabad, India
pvvpr,vv@iiit.net

Abstract

In this paper we present a statistical transliteration technique that is language independent. This technique uses Hidden Markov Model (HMM) alignment and Conditional Random Fields (CRF), a discriminative model. HMM alignment maximizes the probability of the observed (source, target) word pairs using the expectation maximization algorithm and then the character level alignments (n-gram) 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 solutions. We apply this technique for Hindi-English transliteration task. The results show that our technique perfoms better than the existing transliteration system which uses HMM alignment and conditional probabilities derived from counting the alignments.

1 Introduction

In cross language information retrieval (CLIR) a user issues a query in one language to search a document collection in a different language. Out of Vocabulary (OOV) words are problematic in CLIR. These words are a common source of errors in CLIR. Most of the query terms are OOV words like named entities, numbers, acronyms and technical terms. These words are seldom found in Bilingual dictionaries used for translation. These words can be the most important words in the query. These words need to be transcribed into document language when query and document languages do not share common alphabet. The practice of transcribing a word or text written in one language into another language is called transliteration.

A source language word can have more than one valid transliteration in target language. For example for the Hindi word below four different transliterations are possible .

गौतम् - gautam, gautham, gowtam, gowtham

Therefore, in a CLIR context, it becomes important to generate all possible transliterations to retrieve documents containing any of the given forms.

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). These systems use GIZA++ (which uses HMM alignment) to get character level alignments (n-gram) from word aligned data. The transliteration system was built by counting up the alignments and converting the counts to conditional probabilities. The readers are strongly encouraged to refer to (Nasreen and Larkey , 2003) to have a detailed understanding of this technique.

In this paper, we present a simple statistical technique for transliteration. This technique uses HMM alignment and Conditional Random Fields (Hanna , 2004) a discriminative model. Based on this technique desired number of transliterations are generated for a given source language word. We also describe the Hindi-English transliteration system built by us. However there is nothing particular to both these languages in the system. We evaluate the transliteration system on a test set of proper names from Hindi-English parallel transliterated word lists. We compare the efficiency of this system with the system that was developed using HMMs (Hidden Markov Models) only.

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

2 Previous work

Earlier work in the field of Hindi CLIR was done by Jaleel and Larkey (Larkey et al., 2003). They did this based on their work in English-Arabic transliteration for cross language Information retrieval (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 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 cross language IR that requires some linguistic knowledge. They evaluate the effectiveness of their system on an English-Japanese cross language IR task.

3 Problem Description

The problem can be stated formally as a sequence labelling 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 suquence) for the source language word (observation sequence).

$$
\begin{array}{c}
x_1 \text{ ———————— } y_1 \\
x_2 \text{ ———————— } y_2 \\
\text{. ———————— .} \\
\text{. ———————— .} \\
x_N \text{ ———————— } y_N
\end{array}
$$

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 HMM alignment 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 HMM alignment and CRF. HMM alignment maximizes the probability of the observed (source, target) word pairs using the expectation maximization algorithm. After the maximization process is complete, the character level alignments (n-gram) are set to maximum posterior predictions of the model. This alignment is used to get character level alignment (n-gram) of source and target language words. From the character level alignment obtained we compare each source language character (n-gram) to a word and its corresponding target language character (n-gram) to a label. Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting sequential data. We use CRFs 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 (n-gram) 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.

The parameters of the CRF are usually estimated from a fully observed training data $\{(x^{(k)}, y^{(k)})\}$. The product of the above equation over all training words, as a function of the parameters λ, is known as the likelihood, denoted by $p(\{y^{(k)}\}|\{x^{(k)}\}, \lambda)$. Maximum likelihood training chooses parameter values such that the logarithm of the likelihood, known as the log-likelihood, is maximized. For a CRF, the log-likelihood is given by

$$L(\lambda) = \sum_k [log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)})]$$

This function is concave, guaranteeing convergence to the global maximum. Maximum likelihood parameters must be identified using an iterative technique such as iterative scaling (Berger , 1997) (Darroch and Ratcliff, 1972) or gradient-based methods (Wallach , 2002). Finally after training the model using CRF we generate desired number of transliterations for a given source language word.

5 Hindi - English Transliteration system

The whole model has three important phases. Two of them are off-line processes and the other is a run time process. The two off-line phases are preprocessing the parallel corpora and training the model using CRF++[2]. 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 transliterations for the given Hindi word (UTF-8 encoded).

5.1 Preprocessing

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

1. Both Hindi and English 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 English-Hindi word pairs were aligned using GIZA++, with English as the source language and Hindi as target language.

3. The instances in which GIZA++ aligned a sequence of English characters to a single Hindi unicode character were counted. The 50 most frequent of these character sequences were added to English symbol inventory. There were hardly any instances in which a sequence of Hindi unicode characters were aligned to a single English character. So, in our model we consider Hindi unicode characters, $NULL$, English unigrams and English n-grams.

4. The English training words were re segmented based on the new symbol inventory, i.e., if

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

a character was a part of an n-gram, it was grouped with the other characters in the n-gram. If not, it was rendered separately. GIZA++ was used to align the above Hindi and English training word pairs, with Hindi as source language and English as target language.

These four steps are performed to get the character level alignment (n-grams) for each source and target language training words.

5. The alignment file from the GIZA++ output is used to generate training file as required by CRF++ to work. In the training file a Hindi unicode character aligned to a English uni-gram or n-gram 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(LBFGS) (Liu and Nocedal, 1989) which uses quasi-newton algorithm for large scale numerical optimization problem. We used Hindi unicode characters as features for our model and a window size of 5.

5.3 Transliteration

The list of Hindi 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 produce the exact n-best results.

6 Evaluation

We evaluate the two transliteration systems for Hindi - English that use HMM alignment and CRF with the system that uses HMM only in two ways. In first evaluation method we compare transliteration accuracies of the two systems using in-corpus (training data) and out of corpus words. In second method we compare CLIR performance of the two systems using Cross Language Evaluation Forum (CLEF) 2007 ad-hoc bilingual track (Hindi-English) documents in English language and 50 topics in Hindi Language. The evaluation document set consists of news articles and reports from Los Angeles Times of 2002. A set of 50 topics representing the information need were given in Hindi. A set of human relevance judgements for these topics were generated by assessors at CLEF. These relevance judgements are binary relevance judgements and are decided by a human assessor after reviewing a set of pooled documents using the relevant document pooling technique. The system evaluation framework is similar to the Craneld style system evaluations and the measures are similar to those used in TREC[3].

6.1 Transliteration accuracy

We trained the model on 30,000 words containing Indian city names, Indian family names, Male first names and last names, Female first names and last names. We compare this model with the HMM model trained on same training data. We tested both the models using in-corpus (training data) and out of corpus words. The out of corpus words consist of both Indian and foreign place names, person names. We evaluate both the models by considering top 5, 10, 15 and 20 transliterations. Accuracy was calculated using the following equation below

$$Accuracy = \frac{C}{N} * 100$$

C - Number of test words with the correct transliteration appeared in the desired number (5, 10, 15, 20, 25) of transliterations.
N - Total number of test words.

The results for 30,000 in-corpus words and 1,000 out of corpus words are shown in the table 1 and table 2 respectively. In below tables 1 & 2 HMM model refers to the system developed using HMM alignment and conditional probabilities derived from counting the alignments, HMM & CRF model refers to the system developed using HMM

[3]Text Retrieval Conferences, http://trec.nist.gov

Model	Top 5	Top 10	Top 15	Top 20	Top 25
HMM	74.2	78.7	81.1	82.1	83.0
HMM & CRF	76.5	83.6	86.5	88.9	89.7

Table 1: Transliteration accuracy of the two systems for in-corpus words.

Model	Top 5	Top 10	Top 15	Top 20	Top 25
HMM	69.3	74.3	77.8	80.5	81.3
HMM & CRF	72.1	79.9	83.5	85.6	86.5

Table 2: Transliteration accuracy of the two systems for out of corpus words.

alignment and CRF for generating top n transliterations.

CRF models for Named entity recognition, POS tagging etc. have efficiency in high nineties when tested on training data. Here the efficiency (Table 1) is low due to the use of HMM alignment in GIZA++.

We observe that there is a good improvement in the efficiency of the system with the increase in the number of transliterations up to some extent(20) and after that there is no significant improvement in the efficiency with the increase in the number of transliterations.

During testing, the efficiency was calculated by considering only one of the correct transliterations possible for a given Hindi word. If we consider all the correct transliterations the efficiency will be much more.

The results clearly show that CRF model performs better than HMM model for Hindi to English transliteration.

6.2 CLIR Evaluation

In this section we evaluate the transliterations produced by the two systems in CLIR task, the task for which these transliteration systems were developed. We tested the systems on the CLEF 2007 documents and 50 topics. The topics which contain named entities are few in number; there were around 15 topics with them. These topics were used for evaluation of both the systems.

We developed a basic CLIR system which performs the following steps

1. Tokenizes the Hindi query and removes stop words.
2. Performs query translation; each Hindi word is looked up in a Hindi - English dictionary and all the English meanings for the Hindi word were added to the translated query and for the words which were not found in the dictionary, top 20 transliterations generated by one of the systems are added to the query.
3. Retrieves relevant documents by giving translated query to CLEF documents.

We present standard IR evaluation metrics such as precision, mean average precision(MAP) etc.. in the table 3 below for the two systems.

The above results show a small improvement in different IR metrics for the system developed using HMM alignment and CRF when compared to the other system. The difference in metrics between the systems is low because the number of topics tested and the number of named entities in the tested topics is low.

7 Future Work

The selected statistical model for transliteration is based on HMM alignment and CRF. This alignment model is used to get character level alignment (n-gram) of source and target language words. The alignment model uses IBM models, such as Model 4, that resort to heuristic search techniques to approximate forward-backward and Viterbi inference, which sacrifice optimality for tractability. So, we plan to use discriminative model CRF for character level alignment (Phil and Trevor , 2006) of source and target language words. The behaviour of the other discrminative models such as Maximum Entropy models etc., towards the transliteration task

Model	P10	tot_rel	tot_rel_ret	MAP	bpref
HMM	0.3308	13000	3493	0.1347	0.2687
HMM & CRF	0.4154	13000	3687	0.1499	0.2836

Table 3: IR Evaluation of the two systems.

also needs to be verified.

8 Conclusion

We demonstrated a statistical transliteration system using HMM alignment and CRF for CLIR that works better than using HMMs alone. The following are our important observations.

1. With the increase in number of output target language words for a given source language word the efficiency of the system increases.

2. The difference between efficiencies for top n and n-5 where $n > 5$; is decreasing on increasing the n value.

References

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

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.*

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

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.*