Combining a Two-step Conditional Random Field Model and a Joint Source Channel Model for Machine Transliteration

Dong Yang, Paul Dixon, Yi-Cheng Pan, Tasuku Oonishi
Masanobu Nakamura and Sadaoki Furui
Department of Computer Science
Tokyo Institute of Techonology
{raymond,dixonp,thomas,oonishi,masa,furui}@furui.cs.titech.ac.jp

Abstract

This paper describes our system for "NEWS 2009 Machine Transliteration Shared Task" (NEWS 2009). We only participated in the standard run, which is a direct orthographical mapping (DOP) between two languages without using any intermediate phonemic mapping. We propose a new two-step conditional random field (CRF) model for DOP machine transliteration, in which the first CRF segments a source word into chunks and the second CRF maps the chunks to a word in the target language. The two-step CRF model obtains a slightly lower top-1 accuracy when compared to a state-of-the-art n-gram joint source-channel model. The combination of the CRF model with the joint source-channel leads to improvements in all the tasks. The official result of our system in the NEWS 2009 shared task confirms the effectiveness of our system; where we achieved 0.627 top-1 accuracy for Japanese transliterated to Japanese Kanji(JJ), 0.713 for English-to-Chinese(E2C) and 0.510 for English-to-Japanese Katakana(E2J) .

1 Introduction

With the increasing demand for machine translation, the out-of-vocabulary (OOV) problem caused by named entities is becoming more serious.

The translation of named entities from an alphabetic language (like English, French and Spanish) to a non-alphabetic language (like Chinese and Japanese) is usually performed through transliteration, which tries to preserve the pronunciation in the source language.

For example, in Japanese, foreign words imported from other languages are usually written in a special syllabary called *Katakana*; in Chinese, foreign words accepted to Chinese are always written by Chinese characters; examples are given in Figure 1.

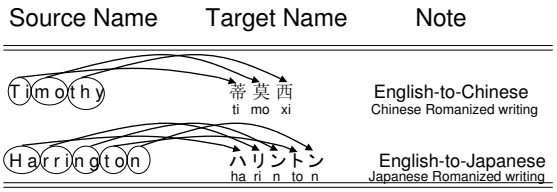

Figure 1: Transliteration examples

An intuitive transliteration method is to first convert a source word into phonemes, then find the corresponding phonemes in the target language, and finally convert to the target language's writing system (Knight and Graehl, 1998; Oh et al., 2006). One major limitation of this method is that the named entities are usually OOVs with diverse origins and this makes the grapheme-to-phoneme conversion very difficult.

DOP is gaining more attention in the transliteration research community which is also the standard evaluation of NEWS 2009.

The source channel and joint source-channel models (Li et al., 2004) have been proposed for DOP, which try to model $P(T|S)$ and $P(T,S)$ respectively, where T and S denotes the words in the target and source languages. (Ekbal et al., 2006) modified the joint source-channel model to incorporate different context information into the model for the Indian languages. Here we propose a two-step CRF model for transliteration, and the idea is to make use of the discriminative ability of CRF. For example, in E2C transliteration, the first step is to segment an English name into alphabet chunks and after this step the number of Chinese characters is decided. The second step is to perform a context-dependent mapping from each English chunk into one Chinese character. Figure 1 shows that this method is applicable to many other

Proceedings of the 2009 Named Entities Workshop, ACL-IJCNLP 2009, pages 72–75,
Suntec, Singapore, 7 August 2009.

transliteration tasks including E2C and E2J.

Our CRF method and the n-gram joint source-channel model use different information in predicting the corresponding Chinese characters and therefore in combination better results are expected. We interpolate the two models linearly and use this as our final system for NEWS 2009. The rest of the paper is organized as follows: Section 2 introduces our system in detail including the alignment and decoding modules, Section 3 explains our experiments and finally Section 4 describes conclusions and future work.

2 System Description

Our system starts from a joint source channel alignment to train the CRF segmenter. The CRF is used to re-segment and align the training data, and from this alignment we create a Weighted Finite State Transducer (WFST) based n-gram joint source-channel decoder and a CRF E2C converter. The following subsections explain the structure of our system shown in Figure 2.

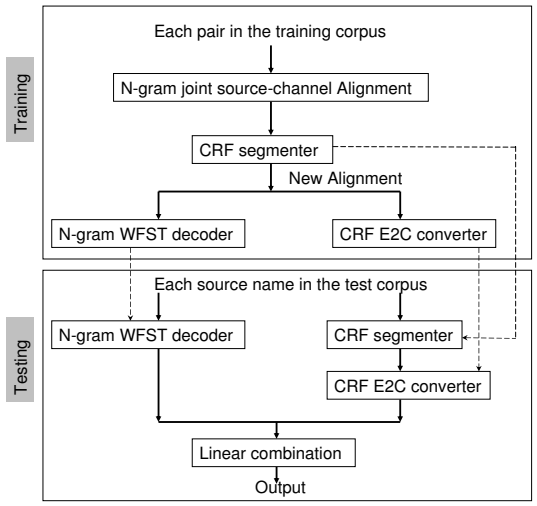

Figure 2: System structure

2.1 Theoretical background

2.1.1 Joint source channel model

The source channel model represents the conditional probability of target names given a source name $P(T|S)$. The joint source channel model calculates how the source words and target names are generated simultaneously (Li et al., 2004):

$$\begin{aligned} & P(S,T) = P(s_1, s_2, ..., s_k, t_1, t_2, ..., t_k) \\ = \; & P(< s,t >_1, < s,t >_2, ..., < s,t >_k) \\ = \; & \prod_{k=1}^{K} P(< s,t >_k \mid < s,t >_1^{k-1}) \end{aligned} \quad (1)$$

where, $S = (s_1, s_2, ..., s_k)$ and $T = (t_1, t_2, ..., t_k)$.

2.1.2 CRF

A CRF (Lafferty et al., 2001) is an undirected graphical model which assigns a probability to a label sequence $L = l_1 l_2 \dots l_T$, given an input sequence $C = c_1 c_2 \dots c_T$,

$$P(L|C) = \frac{1}{Z(C)} exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(l_t, l_{t-1}, C, t)) \quad (2)$$

For the k^{th} feature, f_k denotes the feature function and λ_k is the parameter which controls the weighting. $Z(C)$ is a normalization term that ensure the distribution sums to one. CRF training is usually performed through the L-BFGS algorithm (Wallach, 2002) and decoding is performed by Viterbi algorithm (Viterbi, 1967). In this paper, we use an open source toolkit "crf++"[1].

2.2 N-gram joint source-channel alignment

To calculate the probability in Equation 1, the training corpus needs to be aligned first. We use the Expectation-Maximization(EM) algorithm to optimize the alignment A between the source S and target T pairs, that is:

$$\widetilde{A} = \arg\max_{A} P(S, T, A) \quad (3)$$

The procedure is summarized as follows:

1. Initialize a random alignment
2. E-step: update n-gram probability
3. M-step: apply the n-gram model to realign each entry in corpus
4. Go to step 2 until the alignment converges

2.3 CRF alignment & segmentation

The performance of EM algorithm is often affected by the initialization. Fortunately, we can correct mis-alignments by using the discriminative ability of the CRF. The alignment problem is converted into a tagging problem that doesn't require the use of the target words at all. Figure 3 is an example of a segmentation and alignment, where the labels B and N indicate whether the character is in the starting position of the chunk or not.

In the CRF method the feature function describes a co-occurrence relation, and it is formally

[1] crfpp.sourceforge.net

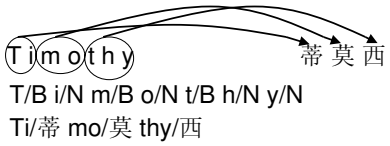

Figure 3: An example of the CRF segmenter format and E2C converter

defined as $f_k(l_t, l_{t-1}, C, t)$ (Eq. 2). f_k is usually a binary function, and takes the value 1 when both observation c_t and transition $l_{t-1} \rightarrow l_t$ are observed. In our segmentation tool, we use the following features

- 1. Unigram features: $C_{-2}, C_{-1}, C_0, C_1, C_2$
- 2. Bigram features:$C_{-1}C_0, C_0C_1$

Here, C_0 is the current character, C_{-1} and C_1 denote the previous and next characters and C_{-2} and C_2 are the characters two positions to the left and right of C_0.

In the alignment process, we use the CRF segmenter to split each English word into chunks. Sometimes a problem occurs in which the number of chunks in the segmented output will not be equal to the number of Chinese characters. In such cases our solution is to choose from the n-best list the top scoring segmentation which contains the correct number of chunks.

In the testing process, we use the segmenter in the similar way, but only take top-1 output segmented English chunks for use in the following CRF E2C conversion.

2.4 CRF E2C converter

Similar to the CRF segmenter, the CRF E2C converter has the format shown in Figure 3. For this CRF, we use the following features:

- 1. Unigram features: C_{-1}, C_0, C_1
- 2. Bigram features:$C_{-1}C_0, C_0C_1$

where C represents the English chunks and the subscript notation is the same as the CRF segmenter.

2.5 N-gram WFST decoder for joint source channel model

Our decoding approach makes use of WFSTs to represent the models and simplify the development by utilizing standard operations such as composition and shortest path algorithms.

After the alignments are generated, the first step is to build a *corpus* to train the transliteration WFST. Each aligned word is converted to a sequence of transliteration alignment pairs $\langle s,t\rangle_1, \langle s,t\rangle_2, ... \langle s,t\rangle_k$, where each s can be a chunk of one or more characters and t is assumed to be a single character. Each of the pairs is treated as a word and the entire set of alignments is used to train an n-gram language model. In these evaluations we used the MITLM toolkit (Hsu and Glass, 2008) to build a trigram model with modified Kneser-Ney smoothing.

We then use the procedure described in (Caseiro et al., 2002) and convert the n-gram to a weighted acceptor representation where each input label belongs to the set of transliteration alignment pairs. Next the pairs labels are broken down into the input and output parts and the acceptor is converted to a transducer M. To allow transliteration from a sequence of individual characters, a second WFST T is constructed. T has a single state and for each s a path is added to allow a mapping from the string of individual characters.

To perform the actual transliteration, the input word is converted to an acceptor I which has one arc for each of the characters in the word. I is then combined with T and M according to $O = I \circ T \circ M$ where $\circ$ denotes the composition operator. The n–best paths are extracted from O by projecting the output, removing the epsilon labels and applying the n-shortest paths algorithm with determinization from the OpenFst Toolkit(Allauzen et al., 2007).

2.6 Linear combination

We notice that there is a significant difference between the correct answers of the n-gram WFST and CRF decoders. The reason may be due to the different information utilized in the two decoding methods. Since their performance levels are similar, the overall performance is expected to be improved by the combination. From the CRF we compute the probability $P_{CRF}(T|S)$ and from the list of scores output from the n-gram decoder we calculate the conditional probability of $P_{n-gram}(T|S)$. These are used in our combination method according to:

$$P(T|S) = \lambda P_{CRF}(T|S) + (1-\lambda) P_{n-gram}(T|S) \quad (4)$$

where λ denotes the interpolation weight (0.3 in this paper).

3 Experiments

We use the training and development sets of NEWS 2009 data in our experiments as detailed in Table 1[2]. There are several measure metrics in the shared task and due to limited space in this paper we provide the results for top-1 accuracy.

Task	Training data size	Test data size
E2C	31961	2896
E2J	23808	1509

Table 1: Corpus introduction

Task	n-gram+CRF Alignment		interpolation
	WFST	CRF	
E2C	70.3	67.3	71.5
E2J	44.9	44.8	46.7

Table 2: Top-1 accuracies(%)

The results are listed in Table 2. For E2C task the top-1 accuracy of the joint source-channel model is 70.3% and 67.3% for the two-step CRF model. After combining the two results together the top-1 accuracy increases to 71.5% corresponding to a 1.2% absolute improvement over the state-of-the-art joint source-channel model. Similarly, we get 1.8% absolute improvement for E2J task.

4 Conclusions and future work

In this paper we have presented our new hybrid method for machine transliteration which combines a new two-step CRF model with a state-of-the-art joint source-channel model. In comparison to the joint source-channel model the combination approach achieved 1.2% and 1.8% absolute improvements for E2C and E2J task respectively.

In the first step of the CRF method we only use the top-1 segmentation, which may propagate transliteration errors to the following step. In future work we would like to optimize the 2-step CRF jointly. Currently, we are also investigating minimum classification error (MCE) discriminant training as a method to further improve the joint source channel model.

[2]For the JJ task the submitted results are only based on the joint source channel model. Unfortunately, we were unable to submit a combination result because the training time for the CRF was too long.

Acknowledgments

The corpora used in this paper are from "NEWS 2009 Machine Transliteration Shared Task" (Li et al., 2004; CJK, website)

References

Kevin Knight and Jonathan Graehl. 1998. *Machine Transliteration*, 1998 Association for Computational Linguistics.

Li Haizhou, Zhang Min and Su Jian. 2004. *A joint source-channel model for machine transliteration*, 2004 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.

Asif Ekbal, Sudip Kumar Naskar and Sivaji Bandyopadhyay. 2006. *A modified joint source-channel model for transliteration*, Proceedings of the COLING/ACL, pages 191-198.

Jong-Hoon Oh, Key-Sun Choi and Hitoshi Isahara. 2006. *A comparison of different machine transliteration models* , Journal of Artificial Intelligence Research, 27, pages 119-151.

John Lafferty, Andrew McCallum, and Fernando Pereira 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.*, Proceedings of International Conference on Machine Learning, 2001, pages 282-289.

Hanna Wallach 2002. *Efficient Training of Conditional Random Fields*. M. Thesis, University of Edinburgh, 2002.

Andrew J. Viterbi 1967. *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*. IEEE Transactions on Information Theory, Volume IT-13, 1967,pages 260-269.

Bo-June Hsu and James Glass 2008. *Iterative Language Model Estimation: Efficient Data Structure & Algorithms.* Proceedings Interspeech, pages 841-844.

Diamantino Caseiro, Isabel Trancosoo, Luis Oliveira and Ceu Viana 2002. *Grapheme-to-phone using finite state transducers*. Proceedings 2002 IEEE Workshop on Speech Synthesis.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut and Mehryar Mohri 2002. *OpenFst: A General and Efficient Weighted Finite-State Transducer Library*. Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007), pages 11-23.

http://www.cjk.org