Integrating Models Derived from non-Parametric Bayesian Co-segmentation into a Statistical Machine Transliteration System

Andrew Finch
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
andrew.finch@nict.go.jp

Paul Dixon
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
paul.dixon@nict.go.jp

Eiichiro Sumita
NICT
3-5 Hikaridai
Keihanna Science City
619-0289 JAPAN
eiichiro.sumita@nict.go.jp

Abstract

The system presented in this paper is based upon a phrase-based statistical machine transliteration (SMT) framework. The SMT system's log-linear model is augmented with a set of features specifically suited to the task of transliteration. In particular our model utilizes a feature based on a joint source-channel model, and a feature based on a maximum entropy model that predicts target grapheme sequences using the local context of graphemes and grapheme sequences in both source and target languages. The segmentation for our approach was performed using a non-parametric Bayesian co-segmentation model, and in this paper we present experiments comparing the effectiveness of this segmentation relative to the publicly available state-of-the-art m2m alignment tool. In all our experiments we have taken a strictly language independent approach. Each of the language pairs were processed automatically with no special treatment.

1 Introduction

In the NEWS2010 workshop, (Finch and Sumita, 2010b) reported that the performance of a phrase-based statistical machine transliteration system (Finch and Sumita, 2008; Rama and Gali, 2009) could be improved significantly by combining it with a model based on the n-gram context of source-target grapheme sequence pairs: a joint source-channel model similar to that of (Li et al., 2004). Their system integrated the two approaches by using a re-scoring step at the end of the decoding process. Our system goes one step further and integrates a joint source-channel model directly into the SMT decoder to allow the probabilities from it to be taken into account within a single search process in the similar manner to (Banchs et al., 2005).

2 System Description

2.1 Bayesian Co-segmentation

The typical method of deriving a translation-model for a machine translation is to use GIZA++ (Och and Ney, 2003) to perform word alignment and a set of heuristics for phrase-pair extraction. A commonly used set of heuristics is known as grow-diag-final-and. This type of approach was taken by (Finch and Sumita, 2010b; Rama and Gali, 2009) to train their models.

An alternative approach is to use a non-parametric Bayesian technique to co-segment both source and target in a single step (Finch and Sumita, 2010a; Huang et al., 2011). This approach has the advantage of being symmetric with respect to source and target languages, and furthermore Bayesian techniques tend to give rise to models with few parameters that do not overfit the data in the same way as traditional maximum likelihood training. In experiments on an English-Japanese transliteration task, (Finch and Sumita, 2010a) showed that that a Bayesian approach offered higher performance than using GIZA++ together with heuristic phrase-pair extraction. Their approach unfortunately required a simple set of agglomeration heuristics in order get good performance from the system. Similarly, (Huang et al., 2011) show that their Bayesian system is able to outperform a baseline based on EM alignment, by removing the need to align to a single grapheme in one language to avoid over-fitting.

In our approach, we adopt the same Bayesian co-segmentation (bilingual alignment) framework as (Finch and Sumita, 2010a), and replace the agglomeration heuristics by incorporating a joint source-channel model directly into the decoder as an additional feature. Our motivation for this was simply that the phrase-based translation model lacks contextual information, and in the experiments of (Finch and Sumita, 2010a), the model gained this contextual information implicitly by the use of agglomerated phrases. In other words,

Proceedings of the 2011 Named Entities Workshop, IJCNLP 2011, pages 23–27,
Chiang Mai, Thailand, November 12, 2011.

the longer phrases carried with them their own built-in context. In our model these contextual dependencies are made explicit and modeled directly by the joint source-channel model.

The termination condition for our Bayesian co-segmentation algorithm was set based on pilot experiments that showed very little gain in system performance after iteration 10, and no loss in performance by continuing the training. We arbitrarily chose iteration 30 in all our experiments as the final iteration.

2.2 Phrase-based SMT Models

The decoding was performed using a specially modified version of the CLEOPATRA decoder (Finch et al., 2007), an in-house multi-stack phrase-based decoder that operates on the same principles as the MOSES decoder (Koehn et al., 2007). The system we used in this shared task is a log-linear combination of 5 different models, the following sections describe each of these models in detail. Due to the small size of many of the data sets in the shared tasks, we used all of the data to build models for the final systems.

2.2.1 Joint source-channel model

The joint source-channel model was trained from the Viterbi co-segmentation arising from the final iteration of the Bayesian segmentation process on the training data (for model used in parameter tuning), and the training data added to the development data (for the model used to decode the test data). We used the MIT language modeling toolkit (Bo-june et al., 2008) with modified Knesser-Ney smoothing to build this model. In all experiments we used a language model of order 5.

2.2.2 Target Language model

The target model was trained from target side of the training data (for model used in parameter tuning), and the training data added to the development data (for the model used to decode the test data). We used the MIT language modeling toolkit with Knesser-Ney smoothing to build this model. In all experiments we used a language model of order 5.

2.2.3 Insertion penalty models

Both grapheme based and grapheme-sequence-based insertion penalty models are simple models that add a constant value to their score each time a grapheme (or grapheme sequence) is added to the target hypotheses. These models control the tendency both of the joint source-channel model and the target language model to generate derivations that are too short.

2.2.4 Maximum-entropy model

In a typical phrase-based SMT system, the translation model contains a context-independent probability of the target grapheme sequence (phrase) given the source. Our system replaces this with a more sophisticated maximum entropy model that takes the local context of source and target graphemes and grapheme sequences into account. The features can be partitioned into two classes: grapheme-based features and grapheme sequence-based features. In both cases we use a context of 2 to the left and right for the source, and 2 to the left for the target. Sequence begin and end markers are added to both source and target and are used in the context. The features used in the ME model consist of all possible bigrams of contiguous elements in the context. We do not mix features at the grapheme level and grapheme sequence level, so for example, a grapheme sequence bigram can only consist of grapheme sequences (including sequences of length 1).

2.3 Parameter Tuning

The exponential log-linear model weights of our system are set by tuning the system on development data using the MERT procedure (Och, 2003) by means of the publicly available ZMERT toolkit [1] (Zaidan, 2009). The systems reported in this paper used a metric based on the word-level F-score, an official evaluation metric for the shared tasks, which measures the relationship of the longest common subsequence of the transliteration pair to the lengths of both source and target sequences.

2.4 Official Results

The official scores for our system are given in Table 1. Some of the data tracks will benefit from a language-dependent treatment (for example in Korean it is advantageous to decompose the characters), and in these tracks our language-independent approach was not competitive. Our system typically gave a strong relative performance on those tracks with larger amounts of training data.

3 Segmentation Experiments

A novel feature of our system is the Bayesian co-segmentation approach used to bilingually segment the data in order to yield training data from which to train the models in our system. It has been

[1] http://www.cs.jhu.edu/∼ozaidan/zmert/

	En-Ch	Ch-En	En-Th	Th-En	En-Hi	En-Ta	En-Ka
Acc.	0.348	0.145	0.338	0.296	0.478	0.441	0.419
F-score	0.700	0.765	0.853	0.854	0.879	0.900	0.885

	En-Ja	En-Ko	Jn-Jk	Ar-En	En-Ba	En-Pe	En-He
Acc.	0.394	0.356	0.454	0.447	0.478	0.615	0.600
F-score	0.803	0.680	0.641	0.911	0.892	0.938	0.929

Table 1: The Evaluation Results on the 2011 Shared Task for our System in terms of the official F-score and Top-1 accuracy metrics.

shown (Finch and Sumita, 2010a) that in transliteration, this Bayesian approach can give rise to a smaller and more useful phrase-table than that derived by using GIZA++ for alignment and the grow-diag-final-and heuristics which have been shown to be effective for transliteration (Rama and Gali, 2009). In these experiments we compare the Bayesian segmenter to a similar state-of-the-art segmentation tool that is capable of many-to-many alignments: the publicly available m2m alignment tool [2] (Jiampojamarn et al., 2007) that is trained using the EM algorithm and is based on the principles set out in (Ristad and Yianilos, 1998).

We used a similar system to that in the shared task, but without the maximum entropy model. The experiments were run in the same way using the same script, the only difference being the choice of aligner used. We used data from the 2009 NEWS workshop for our experiments, and evaluated using the F-score metric used for the shared task evaluation. The aligners were run with their default settings, and with the same limits for source and target segment size. It may have been possible to obtain better performance from the aligners by adjusting specific parameters, but no attempt was made to do this. The results are shown in Table 2. In all experiments, the Bayesian segmenter gave the best performance, and the largest improvement was on language pairs that have large grapheme set sizes on the target side. The grapheme set size is shown in Table 2 in the 'Target Types' column. The source grapheme set sizes were very similar and small (around 27) for all experiments, as the source language was either English or in the case of Jn-Jk, a romanized form of Japanese. Looking at the n-gram statistics in Table 2, for languages with large grapheme sets the number of unigrams in the Bayesian model is less than half that used by the m2m model. Learning a compact model is one of the signature characteristics of the Bayesian model we use; adding a new parameter to the model is extremely costly, and the algorithm will therefore strongly prefer to learn a model in which the parameters are re-used.

Initially we considered the hypotheses that the difference in performance between these two approaches came from differences in the sparseness of the language models. Surprisingly however, the numbers of bi-grams and tri-grams in the joint language models are quite similar.

Another explanation is that the smaller number of unigrams indicates that the segmentation is more self-consistent and therefore makes the generation task less ambiguous. This is supported by looking at the development set perplexity. On the Jn-Jk task where the differences between the systems are the largest, we found that a joint language model trained on the Bayesian segmentation had 1-, 2-, and 3-gram perplexities of 218.3, 88.4 and 87.5 respectively, whereas the corresponding m2m model's perplexities were 321.8, 120.5 and 119.3. The number of segments used to segment the corpus was the same for both systems in this experiment.

Table 3 gives an example from the data of the differences in segmentation consistency. The Bayesian segmentation is strongly self-consistent. The source sequence 'ara' has been segmented identically as a single unit in all cases. The m2m system also shows self-consistency, but uses a few different strategies to segment the start of the sequence. Interestingly the Bayesian method in this example has segmented according to the correct linguistic readings of the kanji. We investigate this further in the next section.

3.1 Linguistic Agreement

In this experiment, we attempt to assess the ability of each segmentation scheme to discover the underlying linguistic segmentation of the data. We took a random sample of 100 word-pairs from the Japanese romaji to Japanese Kanji training corpus. The segmentation of this sample using both systems was then labeled as either 'correct' or 'incorrect' by a human judge using a Japanese

[2] http://code.google.com/p/m2m-aligner/

Language Pairs	Target Types	m2m F-score	Bayesian F-score	m2m			Bayesian		
				1-grams	2-grams	3-grams	1-grams	2-grams	3-grams
En-Ch	372	0.858	0.880	9379	44003	75513	4706	38647	72905
En-Hi	84	0.874	0.884	3114	15209	30195	1867	20218	34657
En-Ko	687	0.623	0.651	4337	11891	14112	2968	11233	14729
En-Ru	66	0.919	0.922	1638	6351	14869	1105	12607	23250
En-Ta	64	0.885	0.892	2852	14696	27869	1561	17195	30244
Jn-Jk	1514	0.669	0.767	7942	27286	38365	3532	22717	37560

Table 2: System performance in terms of F-score, by using alternative segmentation schemes together with statistics relating to be number of parameters in the models derived from the segmentations.

m2m			Bayesian		
arad↦荒	a↦田		ara↦荒	da↦田	
ar↦新	ae↦江		ara↦新	e↦江	
ar↦荒	ahori↦堀		ara↦荒	hori↦堀	
ar↦新	ai↦井		ara↦新	i↦井	
ar↦新	ai↦居		ara↦新	i↦居	
ar↦荒	ai↦井		ara↦荒	i↦井	
ar↦荒	ai↦居		ara↦荒	i↦居	
araj↦荒	ima↦島		ara↦荒	jima↦島	
arak↦新	i↦木		ara↦新	ki↦木	
arak↦荒	i↦木		ara↦荒	ki↦木	
ar↦荒	akid↦木	a↦田	ara↦荒	ki↦木	da↦田
ar↦荒	ao↦尾		ara↦荒	o↦尾	
ar↦荒	ao↦生		ara↦荒	o↦生	
ar↦荒	aoka↦岡		ara↦荒	oka↦岡	
arasa↦荒	wa↦沢		ara↦荒	sawa↦沢	
ar↦荒	aseki↦関		ara↦荒	seki↦関	

Table 3: Example segmentations from the m2m segmenter and the Bayesian segmenter, taken from a long contiguous section of the training set where both techniques disagree on the segmentation.

name reading dictionary as a reference. We found that Bayesian segmentation agreed with the human segmentation in 96% of the test cases, and whereas the m2m system agreed in 42% of cases.

4 Conclusion

The system entered in the year's shared task is built within a statistical machine translation framework, but has been augmented by adding features specifically suited to transliteration. In particular, a joint source-channel model and a maximum entropy model were integrated into the decoder to enhance the translation model of the SMT system by contributing local contextual information. Our system uses a novel Bayesian co-segmentation technique to perform a many-to-many source-target sequence alignment of the corpus. The models of our system are trained directly from this co-segmentation. We have shown that this technique is very effective for producing training data for a joint source-channel model, and is able to accurately induce the linguistic segmentation of Japanese names, building a compact model based on a self-consistent segmentation of the data. In the future we would like to develop more sophisticated Bayesian models, and investigate methods for identifying and dealing with different source languages. We would also like to measure the utility of training the language model component of our system independently on large amounts of monolingual data, which is often much more readily available than aligned bilingual corpora.

Acknowledgements

For the English-Japanese, English-Korean and Arabic-English datasets, the reader is referred to the CJK website: http://www.cjk.org. For English-Hindi, English-Tamil, and English-Kannada, and English-Bangla the data sets originated from the work of (Kumaran and Kellner, 2007).

References

Rafael E. Banchs, Josep Maria Crego, Adria Degispert, Patrik Lambert, Marta Ruiz, and Jose A. R. Fonollosa. 2005. Bilingual n-gram statistical machine translation. In *Proc. of Machine Translation Summit X*, pages 275–282.

Bo-june, Paul Hsu, and James Glass. 2008. Iterative language model estimation: Efficient data structure and algorithms. In *Proc. Interspeech*.

Andrew Finch and Eiichiro Sumita. 2008. Phrase-based machine transliteration. In *Proc. 3rd International Joint Conference on NLP*, volume 1, Hyderabad, India.

Andrew Finch and Eiichiro Sumita. 2010a. A Bayesian Model of Bilingual Segmentation for Transliteration. In Marcello Federico, Ian Lane, Michael Paul, and François Yvon, editors, *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 259–266.

Andrew Finch and Eiichiro Sumita. 2010b. Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In *Proceedings of the 2010 Named Entities Workshop*, NEWS '10, pages 48–52, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew Finch, Etienne Denoual, Hideo Okuma, Michael Paul, Hirofumi Yamamoto, Keiji Yasuda, Ruiqiang Zhang, and Eiichiro Sumita. 2007. The NICT/ATR speech translation system for IWSLT 2007. In *Proceedings of the IWSLT*, Trento, Italy.

Yun Huang, Min Zhang, and Chew Lim Tan. 2011. Nonparametric Bayesian Machine Transliteration with Synchronous Adaptor Grammars. In *ACL (Short Papers)*, pages 534–539.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379, Rochester, New York, April. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowa, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL 2007: proceedings of demo and poster sessions*, pages 177–180, Prague, Czeck Republic, June.

A. Kumaran and Tobias Kellner. 2007. A generic framework for machine transliteration. In *SIGIR'07*, pages 721–722.

Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 159, Morristown, NJ, USA. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proceedings of the ACL*.

Taraka Rama and Karthik Gali. 2009. Modeling machine transliteration as a phrase based statistical machine translation problem. In *NEWS '09: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 124–127, Morristown, NJ, USA. Association for Computational Linguistics.

Eric Sven Ristad and Peter N. Yianilos. 1998. Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532, May.

Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.