Synchronous Learning of Chinese Word Segmentation and Word Alignment

Chapter · January 2011

CITATIONS
0

READS
32

4 authors:

Jia Xu
Stevens Institute of Technology
64 PUBLICATIONS **1,049** CITATIONS
SEE PROFILE

Jianfeng Gao
Chinese Academy of Sciences
598 PUBLICATIONS **59,893** CITATIONS
SEE PROFILE

Kristina Toutanova
Google Inc.
131 PUBLICATIONS **16,898** CITATIONS
SEE PROFILE

Hermann Ney
RWTH Aachen University
1,041 PUBLICATIONS **50,834** CITATIONS
SEE PROFILE

improvement in TER, but no improvement in BLEU. The total improvement over the word-based baseline is 0.5 in BLEU and 0.9 in TER. The improvements are in line with those described in the previous section.

2.2.3.7 Discussion

In our experiments, we concatenated the aligned training data from different segmenters in rule extraction. Obviously, not all segmenters are equally useful. A refinement is to weight the alignments from different segmenters (or exclude some segmenters altogether) based on performances on the development set. While performing such an experiment is straightforward, due to the limitation of time we leave it for future work.

While character-based decoding is similar to decoding a segmentation lattice (Dyer *et al*. 2008), it has two advantages. First, it incurs a smaller overhead in translation table size, since rules with the same unsegmented source string are merged. In our experiments, the character-based phrase translation table (extracted from the combined alignments) was only 40% larger than that of the word-based translation table (left to right word segmentation). Due to a compact translation table, the overhead in decoding speed is also small: Character-based decoding was only 45% slower than the word-based baseline. We expect lattice-based methods will result in a much larger translation table and a significant reduction in decoding speed. A disadvantage with character-based decoding is that it cannot assign weights to different segmentations of the input in scoring translation theories, while lattice-based methods do not have this problem.

2.2.4. Synchronous Learning of Chinese Word Segmentation and Word Alignment for Statistical Machine Translation

Authors: Jia Xu, Jianfeng Gao, Kristina Toutanova and Hermann Ney

2.2.4.1 Introduction

Chinese sentences are written in the form of a sequence of Chinese characters; words are not separated by white spaces. This is different from most European languages and poses difficulty in many natural language processing tasks, such as machine translation.

It is difficult to define "correct" Chinese word segmentation (CWS) and various definitions have been proposed. The common solution in Chinese-to-English translation has been to segment the Chinese text using an off-the-shelf CWS method and to apply a standard translation model given the fixed segmentation. The most widely applied method for MT is unigram segmentation, such as segmentation using the LDC (LDC 2003) tool, which requires a manual lexicon containing a list of Chinese words and their frequencies. The lexicon and

frequencies are obtained using manually annotated data. This method is sub-optimal for MT, because words out of the manual lexicon cannot be generated. In addition to unigram segmentation, other methods have been proposed. For example, (Gao *et al*. 2005) described an adaptive CWS system and Andrew (2006) and Chang et. al. (2008) employed a conditional random field model for word segmentation. However, these methods are not specifically developed for the MT application, where Chinese word segmentation and translation model training are separate steps although they influence each other.

In the work of Xu *et al*. (2004), word segmentations are learned from word alignments. We refine this method by integrating the Chinese word segmentation into the word alignment training so that the word segmentation and alignment can be learned synchronously and their effects on each other can be considered in the training. We present a log-linear model derived from a generative model which consists of a word model and two alignment models, representing the monolingual and bilingual information, respectively. The model is trained using Gibbs sampling. Alternative segmentation boundaries and realignments of words due to the change of these boundaries are taken into account in the sampling process. New Chinese words are generated using Dirichlet Process and the lexicon is updated dynamically. In this way, two problems are solved: adaptation to the parallel training corpus, and out of vocabulary words.

Our experiments on both large (GALE) and small (IWSLT) data tracks of Chinese-to-English translation show that our method improves the performance of state-of-the-art machine translation systems.

2.2.4.2 Review of the Baseline System

In statistical machine translation, we are given a Chinese sentence in characters $c_1^K = c_1 \dots c_K$ which is to be translated into an English sentence $e_1^I = e_1 \dots e_I$. In order to obtain a more adequate mapping between Chinese and English words, c_1^K is usually segmented into words $f_1^J = f_1 \dots f_J$ in preprocessing.

In our baseline system, we apply the commonly used unigram model to generate the segmentation. Given a manually compiled lexicon containing words and their relative frequencies, the best segmentation is the one that maximizes the joint probability of all words in the sentence, under the assumption that words are independent of each other. However, a human collected lexicon can hardly cover all Chinese words in various domains. Words out of the lexicon list are dropped during word segmentation and might not be able to contribute to the translation any more. Inaccurate word distributions can also result in sub-optimal segmentation.

Once we have segmented the Chinese sentences into words, we train standard alignment models in both directions with GIZA++ (Och and Ney 2002) using models of IBM-1 (Brown *et al*. 1993), HMM (Vogel *et al*. 1996) and IBM-4 (Brown *et al*. 1993). The translation system uses a phrase-based decoder with a log-linear model described by Zens and Ney (2004). The feature weights are

tuned on the development set using a downhill simplex algorithm (Press *et al*. 2002). The language model is a statistical *n*-gram model estimated using modified Kneser-Ney smoothing.

2.2.4.3 Semi-supervised Word Segmentation

We introduce a semi-supervised approach to perform Chinese word segmentation as illustrated in Figure 2.2. The inputs to the system are the bilingual training data, including a set of Chinese sentences in characters and its English translations, a manual Chinese word lexicon, such as LDC lexicon, as well as the test corpus on character level. First, we segment the Chinese training corpus with a unigram segmenter using the manual lexicon and get an initialized training corpus in words. Then, we perform the synchronous training of Chinese word segmentations and word alignments to maximize the likelihood of a log-linear model. Optimal word segmentations and alignments are generated as outputs. By counting the Chinese word frequencies of the generated training corpus, we obtain a lexicon. To combine this lexicon with a manual lexicon, we interpolate the probabilities of each word entry in both lexicons linearly. This combined lexicon is applied to segment the test corpus using unigram segmentation. The optimal Chinese word segmentations of the training and test data, as well as, the alignments of the training data, are the system outputs, which will be used further into the decoder.

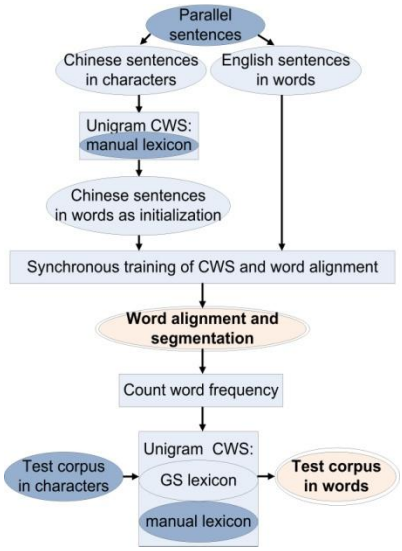

Figure 2.2: Semi-supervised CWS process.

2.2.4.4 Generative and Log-linear Model

Observations		

Chinese characters	c_1^K	小孩玩纸牌
English sentence	e_1^I	Children play cards
Hidden variables		
Alignment normal	a_1^J	e.g. (cards, 纸),(cards, 牌)
Alignment inverse	b_1^I	e.g. (纸,cards),(牌, cards)
Chinese words	f_1^J	e.g. 小孩玩纸牌

Table 2.16: Observations and hidden variables of the generative model.

As a solution to the problems with the conventional approach to CWS mentioned in Section 2.2.4, we propose a generative model for CWS in this section and then extend the model to a more general, but deficient model, a log-linear model in which most features are derived from the sub-models of the generative model.

As shown in Table 2.16, the generative model assumes that a corpus of parallel sentences (c_1^K, e_1^I) is generated along with a hidden sequence of Chinese words f_1^J and a hidden word alignment b_1^I in the inverse direction for every sentence. The joint probability of the observations (c_1^K, e_1^I) can be obtained by summing over all possible values of the hidden variables f_1^J and b_1^I and each value is computed as following:

$$Pr(c_1^K, e_1^I, f_1^J, b_1^I) = Pr(f_1^J)\delta(f_1^J, c_1^K)Pr(e_1^I, b_1^I|f_1^J) \tag{2.1}$$

$$\approx \frac{1}{z}P(f_1^J)^{\lambda_1}P(e_1^I, b_1^I|f_1^J)^{\lambda_2}P(f_1^J, a_1^J|e_1^I)^{\lambda_3} \tag{2.2}$$

$\delta(f_1^J, c_1^K)$ is 1 if the characters of the sequence of words f_1^J are c_1^K and to 0 otherwise, Z is the normalization factor. We can drop the conditioning on c_1^K in $Pr(e_1^I, b_1^I|f_1^J)$, because the characters are deterministic given the words.

In Equation 2.2, we put the monolingual model and the translation models in both directions together into a single model, where each of the component models is weighted by a scaling factor. This model can be viewed as a weighted linear combination of the log probabilities of sub-models. The weights, which are optimized on a development dataset, have empirical justifications. Since different sub-models are trained on different datasets, their dynamic value ranges can be so different that it is inappropriate to combine their log probabilities through simple addition. Moreover, some models may be poorly estimated due to for example the lack of large amount of training data. Therefore, empirical results have demonstrated that the use of scaling factors that reflect the relative contribution of different sub-models often improves the performance.

In practice, we do not renormalize probabilities and our model is thus deficient because it does not sum to 1 over observations. However, the model works very well in our experiments. Similar deficient models have been used successfully before, such as in IBM models (Och and Ney 2002).

Monolingual Chinese sentence model

We use the Dirichlet Process unigram word model (Xu *et al*. 2008) to introduce new Chinese word types and to learn word distributions automatically from unlabeled data, where the parameters of a distribution over words G are first drawn from the Dirichlet prior $DP(\alpha, P_0)$. Words are then indepen dently generated according to G. The probability of a sequence of Chinese words in a sentence is thus:

$$Pr(f_1^J) = \prod_{j=1}^{J} P_G(f_j) \tag{2.3}$$

Translation model

We employ the Dirichlet Process inverse IBM model 1 to generate English words and alignments given the Chinese words. In this model, for every Chinese word f (including the null word), a distribution over English words G_f is first drawn from a Dirichlet Process prior $DP(\alpha, P_0(e))$, where for $P_0(e)$ we use the empirical distribution over English words in the parallel data. Then the probability of an English sentence and alignment given a Chinese sentence in words is given by

$$Pr(e_1^I, b_1^I | f_1^J) = \prod_{i=1}^{I} \frac{1}{J+1} P_{G_{f_{b_i}}}(e_i | f_{b_i}) \tag{2.4}$$

where the probability of e_i is distributed according to $G_{f_{b_i}}$. This is the same model form as inverse IBM model 1, except we have placed Dirichlet Process priors on the Chinese word specific distributions over English words.[7]

In practice, we observed that using a word alignment model in one direction is not sufficient then added a factor to our model which includes the word alignment in the other direction, i.e., a Dirichlet Process IBM model 1. We ignore the detailed description here, because the calculation is the same as that of the inverse IBM model 1. According to this model, for every English word e (including the null word), a distribution over Chinese words G_e is first drawn from a Dirichlet Process prior $DP(\alpha, P_0(f))$. Here, for the base distribution $P_0(f)$ we used the same spelling model as for the monolingual unigram Dirichlet Process prior as described by Xu *et al*. (2008). The probability of a sequence of Chinese words f_1^J and a word alignment a_1^J given a sequence of English words e_1^I is then computed in the same way.

[7] f_{b_i} is the Chinese word aligned to e_i and $G_{f_{b_i}}$ is the distribution over English words conditioned on the word f_{b_i}. Similarly, e_{a_j} is the English word aligned to f_j in the other direction and $G_{e_{a_j}}$ is the distribution over Chinese words conditioned on e_{a_j}.

Gibbs Sampling Training

It is generally impossible to find the most likely segmentation according to our Bayesian model using exact inference, because the hidden variables do not allow exact computation of the integrals. Nonetheless, it is possible to define algorithms using Markov chain Monte Carlo (MCMC) that produce a stream of samples from the posterior distribution of the hidden variables given the observations. We applied the Gibbs sampler (Geman and Geman 1984), one of the simplest MCMC methods, in which transitions between states of the Markov chain result from sampling each component of the state conditioned on the current value of all other variables. For a complete discussion of Gibbs sampling training and the word segmentation and realignment algorithm used in our experiment see (Xu et. al. 2008).

The Gibbs sampler for our model works as follows: For each iteration we sample on each character position by fixing other segmentations and alignments, then we compare hypotheses considering the segmentation and the related alignments of this position. Each position has two alternative segmentations: a boundary exists, or not. The change of a segmentation boundary causes relinking alignment points to parts or groups of the original words. In the work of Xu et. al. (2008) all alignment alternatives are discussed in detail. Together with the boundary versus no-boundary state at each character position, we sample a set of alignment links between English words and any of the Chinese words related to this position given all other word alignments and segmentations in the parallel corpus fixed. After sampling by using the posterior probabilities of each candidate, we choose one of these candidates and perform the same operation for the next position. This process is usually terminated until the result is converged. Since we only implemented the IBM model 1 in both directions for computational efficiency, more advanced word alignment models are applied by repeatedly aligning the corpus using GIZA++.

2.2.4.5 Translation Experiments

We performed experiments using our models on a large and a small data track. We evaluated performance by measuring WER (word error rate), PER (position-independent word error rate), BLEU (Papineni *et al*. 2002) and TER (translation error rate) (Snover *et al*. 2006) using multiple references.

Translation Task: Large Track GALE Translation

We first report the experiments on the GALE machine translation task (GALE 2008). The bilingual training corpus is a superset of corpora in the news, conversation domains collected from different sources provided under the GALE program. As shown in Table 2.17, the training corpus in each language contains more than seven million sentences after the bilingual sentence segmentation (Xu

et al. 2005b). We took LDC (LDC 2003) as baseline method to compare. The word segmentation using Gibbs Sampling (GS) and baseline method generated 92.8 and 93.9 million Chinese running words respectively.

		Chinese		English
		LDC	GS	
Train	Sentences[M]	7.57		
	Running Words[M]	93.9	92.8	102
	Vocabulary[K]	112	121	347
	Singletons[K]	38.1	38.3	152
Test	Sentences	1943		
	Running Words[K]	44.3	44.3	53.2
	Vocabulary[K]	6.78	6.60	6.15
	OOVs (R. W.)	15	17	246
	OOVs (in voc.)	13	14	158

Table 2.17: Statistics of corpora in task GALE.

The CWS model parameters are not optimized but fixed as applied in the IWSLT task because of the computational complexity. The log-linear model scaling factors in the decoder as mentioned in Section 2.2.4.2 are neither optimized and we took the values optimized on the baseline system for convenience. The resulting systems were evaluated on the test corpus in 2008 including all domains with 1943 sentences. We only list the statistics of the first English reference.

Starting from the unigram segmentation as initial word segmentation, we performed Gibbs sampling with only one iteration, which takes several hours, on the training corpus because of the large computational requirement. After that, we merged the GS generated lexicon with a weight of 0.4 and the manual LDC lexicon with a weight of 0.6 using linear combination. Then we performed the unigram segmentation on the test corpus using the combined lexicon.

As shown in Table 2.18, on the test data, the BLEU score was improved by 0.5% absolutely or more than 1.8% relatively using GS with combined lexicon. The TER score is also enhanced significantly, i.e., 0.8% absolutely and 0.9% relatively.

Method	WER	PER	BLEU	TER
LDC	73.0	49.5	28.2	67.1
Unigram	73.0	49.7	28.4	67.2
GS with combined lexicon	72.5	48.6	28.7	66.3

Table 2.18: Translation performance [%] with the baselines (LDC, unigram) and GS method on GALE.

We can see that although the semi-supervised word segmentation is not yet converged, it can still outperform a supervised one in MT. One of the reasons is probably the training and test corpora contain many words and words have

different frequencies in our MT data from they do in the manually labeled CWS data.

Task: Small Track IWSLT

The Chinese training corpus of the IWSLT task was segmented using the unigram segmenter as baseline method (Baseline) and our GS method. The parameter optimizations were performed on the Dev2 data with 500 sentences and evaluations were done on both the Dev3 and the Eval data, i.e., the evaluation corpus of (IWSLT 2007).

The model weights of GS were optimized using the Powell (Press *et al*. 2002) algorithm with respect to the BLEU score. We obtained the optimal number of iterations of the GIZA++ word alignment update as four.

Test	Method	WER	PER	BLEU	TER
Dev2	Unigram (Baseline)	38.2	31.2	55.4	37.0
	GS	36.8	30.0	56.6	35.5
Dev3	Unigram (Baseline)	33.5	27.5	60.4	32.1
	GS	32.3	26.6	61.0	31.4
Eval	Characters	49.3	41.8	35.4	47.5
	LDC	46.2	40.0	39.2	45.0
	ICT	45.9	40.4	40.1	44.9
	Unigram (Baseline)	46.8	40.2	41.6	45.6
	9-gram	46.9	40.4	40.1	45.4
	GS	45.9	40.0	41.6	44.8

Table 2.19: Translation performance with different CWS methods on IWSLT[%].

For a fair comparison, we evaluated on various CWS methods including translation on characters, LDC (LDC 2003), ICT (Zhang *et al*. 2003), unigram, 9-gram and GS. Improvements using GS can be seen in Table 2.19. Under all test sets and evaluation criteria, GS outperforms the baseline method. The absolute WER decreases with 1.2% on the Dev3 and with 0.9% on the Eval data over the baseline. In the BLEU score there is no change on the Eval set between the baseline and the GS, because the Eval data has a lower vocabulary coverage with the Dev2 than the other test sets such as the Dev3 do. The optimization of many parameters leads to a slight over-fitting of the model, so that the parameters may not be optimal for the Eval translation.

a) Baseline	有近路吗？
	do you have a ?
GS	有近路吗？
	do you have a shorter way ?
REF	is there a shorter route ?
b) Baseline	请告诉我总金额
	please show me the in .
GS	请告诉我总金额
	please show me the total price .

REF	can you tell me the total amount?

Table 2.20: Segmentation and translation outputs with baseline and GS methods.

We compared the translation outputs using the GS with those using the baseline method. On the Eval data, 196 sentences have different translations out of 489 lines, where 64 sentences from the GS are better, 33 sentences are worse and the rests have similar translation qualities. Table 2.20 shows two examples from the Eval corpus. We list segmentations produced by the baseline and the GS methods, as well as the translations generated using these segmentations. The GS method generates better translation results than the baseline method in these cases.

2.2.4.6 Conclusion and Future Work

We showed that it is possible to learn Chinese word boundaries during the word alignment training so that the translation performance of Chinese-English MT systems is improved.

We presented a Bayesian generative model for parallel Chinese-English sentences, which uses word segmentation and alignment as hidden variables and incorporates both monolingual and bilingual information to derive word segmentation and alignment for MT.

Starting with initial word segmentation, our method learns both new Chinese words and word distributions using the Dirichlet Process. In a small data environment and a large data environment, our method outperformed the standard Chinese word segmentation approach in terms of the Chinese-English translation quality. In future work, we plan to enrich our models to better represent the true distribution of the data.

Chapter 2.3 Word Alignment

2.3.1. Word Alignment Revisited

Authors: Francisco Guzman, Qin Gao, Jan Niehues and Stephan Vogel

2.3.1.1 Introduction

Word alignment can be considered the backbone of Statistical Machine Translation. Even when Statistical Machine Translation (SMT) shifted from a word-based to a phrase-based paradigm, word alignment remained the base for most phrase-based (Koehn *et al*. 2003), hierarchical (Chiang 2007) and syntactic SMT systems (Zollmann and Venugopal 2006; Marcu *et al*. 2006). Generative models have the advantage that they are well suited for a noisy-channel approach. Unsupervised training can be used to align a large amount of unlabeled parallel corpora. Nonetheless, they have a major disadvantage, because these models are completely unsupervised, they can hardly make use of the