Sequence Classification for Machine Translation

Srinivas Bangalore, Patrick Haffner, Stephan Kanthak

AT&T Labs-Research
180 Park Ave
Florham Park, NJ 07932
{srini,haffner,kanthak}@att.research.com

Abstract

Discriminatively trained classification techniques have been shown to out-perform generative techniques on many speech and natural language processing problems. However, most of the research in machine translation has been based on generative modeling techniques. The application of classification techniques to machine translation requires scaling classifiers to deal with very large label sets (the vocabulary of the target language). In this paper, we present a method to scale classifiers to very large label sets and apply it to train classifiers for machine translation. We contrast this approach to a generatively trained machine translation model represented as a weighted finite-state transducer. We show translation accuracy results on spoken language corpora in English to Spanish and English to Japanese translation tasks.

Index Terms: Machine Translation, Disciminant classification, Stochastic finite-state transducer based machine translation.

1. Introduction

Discriminatively trained classification-based techniques have become the dominant approach for resolving ambiguity in speech and natural language processing problems. Although these techniques originated for document routing tasks which use features from the entire document, they have also been successfully applied to word-level disambiguation tasks such as part-of-speech tagging, named-entity tagging, and dependency parsing tasks which rely on features in the local context of a word. Models trained using these approaches have been shown to out-perform generative models as they directly optimize the conditional distribution without modeling the distribution of the independent variables.

However, most of machine translation research has focused on generative modeling techniques. Discriminative training has been used only for model combination [1] but not directly to train the parameters of a model. Applying discriminatively trained classification techniques directly to estimate the parameters of a translation model requires scaling the classifiers to deal with very large label sets, typically the size of the target language vocabulary. In this paper, we present a method for scaling the classifiers to such large label sets and apply it to train machine translation models for spoken language translation tasks.

There have been several attempts at exploiting syntactic information in a generative modeling framework to improve the accuracy of machine translation [2]. However, these approaches have met with only marginal success at best. We believe that the discriminative classification framework is more suitable for exploiting such linguistically rich information as they do not model the distribution of independent variables and hence are not affected by sparseness issues that typically affect generative models.

The outline of the paper is as follows. In Section 2, we review statistical machine translation models and the alignment training method that is needed for this approach. We present the different types of decoders that are used for statistical machine translation in Section 3 and focus on weighted finite-state transducer based decoder in Section 4. In Section 5, we present different sequence classification techniques and in Section 6, we compare the performance of different translation models on spoken language corpora for English to Spanish and English to Japanese translation tasks.

2. Statistical Machine Translation Model

In machine translation, the objective is to map a source symbol sequence $S = s_1, \ldots, s_N$ ($s_i \in L_S$) into a target sequence $T = t_1, \ldots, t_M$ ($t_i \in L_T$). This can be formulated as a search for the best target sequence that maximizes $P(T|S)$. Ideally, $P(T|S)$ should be estimated directly to maximize the conditional likelihood on the training data (discriminant model). However, T corresponds to a sequence with a exponentially large combination of possible labels, and traditional classification approaches cannot be used directly. To overcome this problem, Bayes transformation is applied and generative techniques are adopted as suggested in the *noisy channel* paradigm [3]. The sequence S is thought as a noisy version of T and the best guess T^* is then computed as

$$T^* = \arg\max_T P(T|S) \quad (1)$$

$$= \arg\max_T P(S|T)P(T) \quad (2)$$

The translation probability $P(S|T)$ is estimated from a corpus of alignments between the tokens of S and tokens of T. Although there have been several approaches to alignment – string-based and tree-based alignment, for the purposes of this paper, we use Giza++ [4] to provide an alignment between tokens of the source language and tokens of the target language. Using the same source of alignments there have been several variations on decoders to compute the best T^* given an input source string S. We discuss some of these decoders in the next section.

3. Decoders for Machine Translation

Equations 1 and 2 can be interpreted in different ways which results in different decoder architectures. We outline below these decoder architectures.

3.1. Conditional Probability Model based Decoders

Using conditional probability models as in Equation 2 has the advantage of composing the translation process from multiple knowledge sources that could be trained independently. Kumar and Byrne [5] have shown that the translation process can be further decomposed into five models, namely source language model, source segmentation model, phrase permutation model, template sequence model and phrasal translation model. As all models are trained independently, different data sets may be used for the estimation of each. Other examples for decoders based on conditional probabilities can be found in [3, 4, 6, 7, 8].

3.2. Joint Probability Model based Decoders

The FST-based decoders as illustrated in [9, 10, 11, 12], decode the target string using a joint probability model $P(S,T)$ from the bilanguage corpus. The bilanguage could be in either source word-order or target word-order. This gives rise to two different two-stage decoders. As shown in Equation 3, first the source string is mapped to a target string in the source word-order. The target string is computed as the most likely string based on the target language model from a set of possible reorderings of $\hat{T}$ (Equation 4).

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

$$\hat{T}^* = \arg\max_{T \in \lambda_{\hat{T}}} P_{\lambda_T}(\tilde{T}) \quad (4)$$

In a different version of the decoder, a set of possible reorderings (λ_S) of the source string is decoded, instead of reordering the decoded target string, as shown in Equation 5.

$$T^* = \arg\max_{T} \sum_{\hat{S} \in \lambda_S} P(\hat{S},T) \quad (5)$$

3.3. Sentence-Based Feature Combination

Relaxing the conditional probability approach to also allow for unnormalized models leads to a sentence-based, exponential feature combination approach (also called log-linear model combination):

$$T^* = \arg\max_{T} \sum_{i} \lambda_i \cdot h_i(S,T) \quad (6)$$

The choice of features is virtually unlimited, but using the approach to tune just the exponents of the conditional probability models in use proves to be quite effective (see also [13, 7, 8]). Crego et.al [12] presents a similar system based on joint probabilities.

4. Finite-state Transducer based Machine Translation Model

In this section, we explain the steps to build a finite-state machine translation model. We start with the bilingual alignment constructed using GIZA++, shown in Figure 1. The *Alignment* string provides the position index of a word in the target string for each word in the source string. Source words that are not mapped to any word have an index 0 associated to them. It is straightforward to compile a bilanguage corpus consisting of source-target symbol pair sequences $\mathcal{T} = \ldots (w_i : x_i) \ldots$, where the source word $w_i \in L_S \cup \epsilon$ and its aligned word $x_i \in L_T \cup \epsilon$ (ϵ is the null symbol). Note that the tokens of a bilanguage could be either ordered according to the word order of the source language or ordered according to the word order of the target language. Figure 2 shows an example alignment and the source-word-ordered bilanguage strings corresponding to the alignment shown in Figure 1. From the corpus $\mathcal{T}$, we train a n-gram language model using language modeling tools [14, 15]. The resulting language model is represented as a weighted finite-state automaton ($S \times T \rightarrow [0,1]$). The symbols on the arcs of this automaton ($s_i\_t_i$) are interpreted as having the source and target symbols (s_i:t_i), making it into a weighted finite-state transducer ($S \rightarrow T \times [0,1]$) that provides a weighted string-to-string transduction from S into T (as shown in Equation 7).

$$T^* = argmax_T P(s_i, t_i | s_{i-1}, t_{i-1} \ldots s_{i-n-1}, t_{i-n-1}) \quad (7)$$

English: I need to make a collect call
Japanese: 私は コレクト コールを かける 必要があります
Alignment: 1 5 0 3 0 2 4

Figure 1: Example bilingual texts with alignment information

I:私は need:必要があります to:ϵ make:コールを
a:ϵ collect_コレクト call_かける

Figure 2: Bilanguage strings resulting from alignments shown in Figure 1.

5. Sequence Classification Techniques

As discussed earlier, Equation 1 represents a direct method for transducing the source language string into the target language string. It depends on estimates of $P(T|S)$. Learning would consist in modifying the parameters of the system so that T^* closely matches the target output sequence $\tilde{T}$. Ideally, $P(T|S)$ should be estimated directly to maximize the conditional likelihood on the training data (discriminant model). However, T corresponds to a sequence output with a exponentially large combination of possible labels, and traditional classification approaches cannot be used directly. Although, Conditional Random Fields (CRF) [16] train an exponential model at the sequence level, in translation tasks such as ours the computational requirements of training such models is prohibitively expensive.

We approximate the string level global classification problem, using independence assumptions, to a product of local classification problems as shown in Equation's 8.

$$P(T|S) = \prod_{i}^{N} P(t_i | \Phi(S,i)) \quad (8)$$

where $\Phi(S,i)$ is a set of features extracted from the source string S (shortened as Φ in the rest of the section).

A very general technique to obtain the conditional distribution $P(t_i|\Phi(S,i))$ is to choose the least informative one (with Maxent) that properly estimates the average of each feature over the training data [17]. This gives us the Gibbs distribution parameterized with the weights λ_t where t ranges over the label set and V is the total number of target language vocabulary.

$$P(t_i|\Phi) = \frac{e^{\lambda_{t_i} \cdot \Phi}}{\sum_{t=1}^{V} e^{\lambda_t \cdot \Phi}} \quad (9)$$

The weights are chosen so as to maximize the conditional likelihood $L = \sum_i L(s_i, t_i)$ with

$$L(S,T) = \sum_i \log P(t_i|\Phi) = \sum_i \log \frac{e^{\lambda_{t_i} \cdot \Phi}}{\sum_{t=1}^{V} e^{\lambda_t \cdot \Phi}} \quad (10)$$

The procedures used to find the global maximum of this concave function include two major families of methods: Iterative Scaling (IS) and gradient descent procedures, in particular L-BFGS methods [18], which have been reported to be the fastest. We obtained faster convergence with a new Sequential L1-Regularized Maxent algorithm (SL1-Max) [19], compared to L-BFGS[1]. We have adapted SL1-Max to conditional distributions

[1] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

for our purposes. Another advantage of the SL1-Max algorithm is that it provides L1-regularization as well as efficient heuristics to estimate the regularization meta-parameters. The computational requirements are $O(V)$ and as all the classes need to be trained simultaneously, memory requirements are also $O(V)$. Given that the actual number of non-zero weights is much lower than the total number of features, we use a sparse feature representation which results in a feasible runtime system.

5.1. Frame level discriminant model: Binary Maxent

For the machine translation tasks, even allocating $O(V)$ memory during training exceeds the memory capacity of current computers. To make learning more manageable we factorize the frame-level multi-class classification problem into binary classification sub-problems. This also allows for parallelization during training the parameters. We use here V one-vs-other binary classifiers at each frame. Each output label t is projected into a bit string, with components $b_j(t)$. The probability of each component is estimated independently:

$$P(b_j(t)|\Phi) = 1 - P(\bar{b}_j(t)|\Phi) = \frac{1}{1 + e^{-(\lambda_j - \lambda_{\bar{j}}) \cdot \Phi}} \qquad (11)$$

where $\lambda_{\bar{j}}$ is the parameter vector for $\bar{b}_j(y)$. Assuming the bit vector components to be independent, we have $P(t_i|\Phi) = \prod_j P(b_j(t_i)|\Phi)$. Therefore, we can decouple the likelihood and train the classifiers independently. In this paper, we use the simplest and most commonly studied code, consisting of V one-vs-others binary components. The independence assumption states that the output labels or classes are independent.

5.2. Maximum Entropy Markov Models or MEMMs

The independence assumption in Equation 8 is very strong, and one can add more context, replacing $P(t_i|\Phi(S,i))$ with $P(t_i|t_{i-1}, \Phi(S,i))$ (bigram independence). While MEMMs [20] allow the use of frame-level Maxent classifiers that learn sequence dependencies, they usually multiply by a factor V the actual number of input features (factor which propagates down to both memory and learning time requirements). Also, MEMMs estimate $P(t_i|t_{i-1}, \Phi(S,i))$ by splitting into $|V|$ separate models $P_{t_{i-1}}(t_i|\Phi(S,i))$. This causes a new problem known as labeling bias [21]: important frame-level discriminant decisions can be ignored at the sequence level, resulting in a loss of performance [22].

5.3. Dynamic Context Maximum Entropy Model

We believe that the label bias problem arises due the manner in which $P(t_i|t_{i-1}, \Phi(S,i))$ is estimated. The estimation of $P_{t_{i-1}}(t_i|\Phi(S,i))$ requires splitting the corpus based on the t_{i-1} label. This leads to incompatible event spaces across the label set during estimation. In order to alleviate this problem, we use the dynamic context as part of the feature function and compute $P(t_i|\Phi(S,i,t_{i-1}))$. We call this the dynamic context model since the features are to be computed dynamically during decoding, in contrast to the static context model presented in Section 5.1 where the features can all be computed statically from the input string.

6. Experiments and Results

We evaluate the translation models on two different spoken language corpora. First, the "How May I Help You" (HMIHY) corpus consists of operator-customer conversations related to telephone services. We use the transcriptions of the customer's utterance which were also manually translated into Japanese and Spanish. The corpus statistics for English-Japanese sentence pairs are given in Table 1. 5812 English-Spanish sentence pairs were used for training, and 829 for testing.

		English	Japanese
Train	Sentences	12226	
	Words	83262	68202
	Vocab	2189	4541
Test	Sentences	3253	
	Words	20533	17520
	Vocab	829	1580

Table 1: Corpus Statistics for the HMIHY Corpus

The second corpus, ATIS, consists of inquiries to airline reservations services which have been manually transcribed and translated into Spanish. The corpus statistics are given in Table 2.

		English	Spanish
Train	Sentences	11294	
	Words	116151	126582
	Vocab	1310	1556
Test	Sentences	2369	
	Words	23469	25538
	Vocab	738	841

Table 2: Corpus Statistics for the ATIS Corpus

The accuracy of the translation models are evaluated using the word accuracy metric. Simple accuracy is computed based on the number of insertion (I), deletion (D) and substitutions (S) errors between the target language strings in the test corpus and the strings produced by the translation model.

$$WordAccuracy = (1 - \frac{I + D + S}{R}) * 100 \qquad (12)$$

The word accuracy results of the translation models on the different corpora are shown in Table 6. We show the baseline model of selecting the most frequent target word for a given source word. As can be seen from the table, the FST-based model outperforms the baseline significantly, but the sequence classification based decoder trained using Maxent training performs better than the FST based decoder on all three corpora.

Domain	Baseline	FST	Maxent (static)	SVM linear	SVM poly2
HMIHY Eng-Jap	59.5	68.6	70.2	69.1	69.7
HMIHY Eng-Spanish	58.6	70.4	71.2	70.2	70.6
ATIS Eng-Spanish	54.5	76.5	78.7	78.6	79.1

The classification approach regards the target words, phrases (multi-tokens) and null symbol (epsilon) as labels. For instance, the ATIS training data contains 336 epsilon labels, 503 phrase labels and 2576 word labels. Using contextual maxent rather than static maxent significantly improves the label classification accuracy (from 65% to 67%).

However, in order to evaluate the word accuracy of the translated string, the classified labels are re-transcribed as words by removing epsilon label and expanding out multi-token labels. We observed no significant difference in word accuracy between the translations provided by static context and dynamic context Maxent models after these transformations.

We conjecture that the loss function we use for the classifier does not properly represent the final objective function. Misclassification between two phrase labels has a variable cost, depending

on the number of words which differ from one phrase to the other, and this is not accounted for in our loss function.[2]

Another way to improve performance is to increase the representation power of the static classifier. We first ran linear SVMs which are the same linear classifiers as Maxent with a different training procedure. The lower word accuracy observed with linear SVMs in Table 6 is explained by an over-detection of words against the epsilon model. The recognized class is obtained by comparing one-versus-other models, and their threshold value requires to be more carefully adjusted, for instance using an additional univariate logistic regression [23]. The fact that we observe an improvement from linear to second degree polynomial SVMs shows that the use of kernels can improve performance.

7. Conclusion

In this paper, we have presented an approach to sequence classification for machine translation. In contrast to previous approaches that use generative methods to estimate the translation probability, we employ discriminative techniques that classify each source word and its context into a target word. We address the challenge of scaling classifiers to large label sets by assuming independence among the output label set. We show results on different spoken language corpora for English to Spanish and English to Japanese translation tasks.

8. References

[1] F. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of ACL*, 2002.

[2] K. Yamada and K. Knight, "A syntax-based statistical translation model," in *Proceedings of* 39^{th} *ACL*, 2001.

[3] P. Brown, S.D. Pietra, V.D. Pietra, and R. Mercer, "The Mathematics of Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 16, no. 2, pp. 263–312, 1993.

[4] F.J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[5] S. Kumar and W. Byrne, "A weighted finite state transducer implementation of the alignment template model for statistical machine translation," in *Proceedings of HLT-NAACL 2003*, Edmonton, Canada, May 2003.

[6] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada, May 2003.

[7] N. Bertoldi, R. Cattoni, M. Cettolo, and M. Federico, "The ITC-IRST Statistical Machine Translation System for IWSLT-2004," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Kyoto, Japan, Sept. 2004, pp. 51–58.

[8] R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney, "The RWTH Phrase-based Statistical Machine Translation System.," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, Oct. 2005, pp. 155–162.

[2] To factor out the impact of the dynamic programming, we ran the dynamic context Maxent using the true test label as context (cheating decoding). Even in this case, after labels are transcribed into words, the dynamic context Maxent model performance is not better than the static context Maxent model performance.

[9] S. Bangalore and G. Riccardi, "Stochastic finite-state models for spoken language machine translation," *Machine Translation*, vol. 17, no. 3, 2002.

[10] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30(2):205–225, 2004.

[11] S. Kanthak and H. Ney, "Fsa: An efficient and flexible c++ toolkit for finite state automata using on-demand computation," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 2004, pp. 510–517.

[12] J. M. Crego, J. B. Marino, and A. de Gispert, "An ngram-based statistical machine translation decoder," in *Proc. of the 9th European Conf. on Speech Communication and Technology (Interspeech'05)*, Lisbon, Portugal, Sept. 2005, pp. 3185–3188.

[13] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, July 2002, pp. 295–302.

[14] V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, and M. Saraclar, "The AT&T WATSON Speech Recognizer," in *Proceedings of ICASSP*, Philadelphia, PA, 2005.

[15] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *Proc. Intl. Conf. Spoken Language Processing*, 2002.

[16] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, San Francisco, CA, 2001.

[17] A.L. Berger, Stephen A. D. Pietra, D. Pietra, and J. Vincent, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[18] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proceedings of CoNLL-2002*. 2002, pp. 49–55, Taipei, Taiwan.

[19] M. Dudik, S. Phillips, and R.E. Schapire, "Performance Guarantees for Regularized Maximum Entropy Density Estimation," in *Proceedings of COLT'04*, Banff, Canada, 2004, Springer Verlag.

[20] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Proc. 17th International Conf. on Machine Learning*. 2000, pp. 591–598, Morgan Kaufmann, San Francisco, CA.

[21] L. Bottou, *Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole*, Ph.D. thesis, Université de Paris XI, 91405 Orsay cedex, France, 1991.

[22] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*. 2001, pp. 282–289, Morgan Kaufmann, San Francisco, CA.

[23] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *NIPS*. 1999, MIT Press.