Conditional Random Fields for Machine Translation System Combination

Tian Xia, Shandian Zhe, Qun Liu
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{xiatian,liuqun}@ict.ac.cn, zsdlightning@gmail.com

***Abstract*—Minimum Error Rate Training (MERT) as an effective parameters learning algorithm is widely applied in machine translation and system combination area. However, there exists an ambiguity problem in respect to the training goal and it is hard for MERT to tackle, that is different parameters may lead to the same minimum error rate in training but greatly different performances in testing. We propose a novel training objective as the unique goal for training towards, namely partial references, and by use of conditional random fields (CRF) to cast the decoding procedure in system combination as a sequence labeling problem. Experiments on Chinese-English translation test sets show that our approach significantly outperforms the MERT-based baselines with less training time.**

***Keywords*-machine translation; conditional random fields; system combination;Minimum Error Rate Training;**

I. Introduction

The mechanism of combining outputs from multiple machine translation systems has shown the great power in machine translation (MT) area. Generally, the framework consists of two independent steps, confusion network (CN) construction [8], [10], [11], [4], [3], and decoding an optimal path evaluated with a set of features. In Table I, hypotheses are aligned to h_0, and corresponding confusion network refers to Figure 1.

h_0	:He	feels	to	apples
h_1	:He	prefer	ε	apples
h_2	:He	ε	like	apples
h_3	:Him	prefer	to	apples

Table I
SUPPOSE h_0 IS SKELETON HYPOTHESIS, TO WHICH OTHERS BE ALIGNED PAIR-WISELY.

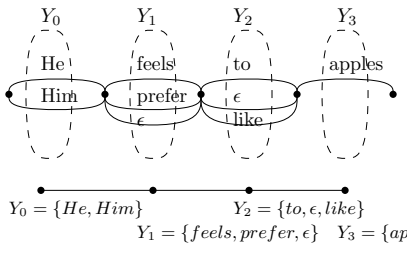

Figure 1. The above graph is about a confusion network, and to be casted as a sequence labeling problem shown in the below graph.

Training algorithm on confusion networks following Minimum Error Rate Training (MERT) [9], [6] aims to learn optimal parameters that could reach the minimum error (or maximum BLEU metric in machine translation) in a development set. Nevertheless, how to define the better one if two completely different parameters cause the same errors? We design an interesting experiment to demonstrate this possibile case.

We train a hierarchical phrase-based translation system for twice. The first time is to let MT02 data set for training and MT05 for testing, and the second is vice verse. We compare all the intermediate data and find two different set of parameters, both are 8-dimension vector, those conduct a similar performance in MT02, whose BLEU score is 0.292, but act obviously differently in MT05, 0.264 and 0.312 in case-sensitive BLEU.

It would be ideal for training parameters towards reference translations. One successful work [1], utilizes the reachable references [1] for CRF training. However, it is impossible to choose reachable confusion networks to train, because most ones could not generate the reference translations, and the available number of confusion network is too poor to waste. Thus, we propose a novel objective, *partial reference*, as the unique objective for each confusion network to train. The *partial reference* is defined as the longest sub-string of the reference translations, and in the meantime could be potentially decoded from a confusion network.

In another view, shown in Figure 1, decoding a confusion network is simply to choose for each span one edge to construct a full translation. If we consider choice for every span as a variable Y, whose values are edges in current span, a simple graphical model is naturally generated.

We adopt conditional random fields (CRF) to train our model on uni-CN due to an important reason, that is CRF model could train a global optimal solution [13]. In the first part experiments, we conduct several experiments to compare the efficiency of parameters training between CRF-based and MERT-based. In the second part, we will make comparison on the task of multi-CN based system combination. Our method is firstly collect the n-best lists from CRF-based systems, and feed a common multi-CN based system to complete a full system combination procedure.

[1]The "Reachable" means reference translations could be potentially generated by the model. In our example, reachable references should be decoded from a confusion network.

978-0-7695-4288-1/10 $26.00 © 2010 IEEE
DOI 10.1109/IALP.2010.91

II. Background

A. Confusion Network

Formally, confusion network (CN) is a directed, acyclic graph, owing the unique source vertex and sink vertex, with each edge attached to one alternative word. A special place-holder ε denotes no concrete word on the arc.

The skeleton hypothesis (also called backbone hypothesis) would determine the words order in final translations, eg. h_0 in Table I. To construct a CN, the remaining hypotheses would be aligned to skeleton [10], [3], [8] or to partially constructed CN [7], [12].

In order to reduce the risk of mischoosing skeleton hypothesis, we will choose separate skeleton for each candidate system to construct several CNs. Considering the solutions from all the CNs to generate the optimal translation. As a result, multi-CN based system combination may generate potential better-quality translations than uni-CN based system combination.

B. Features in MERT

Features used in our work and baseline systems are nearly the same to [10], [3], which are modeled in a log-liner fashion. Four class features are defined as follows.

1) word posterior probabilities. $p(w|sys, span)$. If the word w comes from k-th hypothesis of sys-th system, the raw score should be $\frac{1}{k+1}$, and then it should be normalized by the sum from the same sys and $span$.
2) logarithm of language model score, Lm.
3) number of ϵ edges, Num_ϵ.
4) number of words, Num_w.

$$\begin{aligned} log(h) &= \sum_{span} \log(\sum_{sys} \lambda_{sys} p(w|sys, span)) \\ &+ w_0 Lm(h) + w_1 Num_\epsilon + w_2 Num_w \end{aligned}$$

III. CRF-based Training on Confusion Network

A. Partial References

We enumerate all the configurations of a CN to search the longest sub-string of reference translations. The part of a CN, capable of generating expected partial references, would be kept for training, and the remnant are thrown away. Note that, there are usually four reference translations for each source sentence, while our model only take use of one partial reference as the training goal.

Since any variable Y_i might be taken as ε, it is important to decide whether it is encouraging to generate more ε or less in partial references. Here, a tricky standard proves to work best.

- making sub-string as longer as possible conditioned on no value ε in two ends.

Table II describe several alternatives of partial references, in which h_3 seems to be longer than h_2 while h_2 is the better objective.

h_1 :	ε	ε	A	B	ε	ε	ε	ε	ε
h_2:			A	B	ε	C			
h_3:		ε	A	B	ε	C	ε	ε	

Table II
SUPPOSE BOTH "A B" AND "A B C" ARE THE SUB-STRING OF ONE OF 4 REFERENCE TRANSLATIONS, AND h_1,h_2,h_3 ARE THE POSSIBLE PARTIAL REFERENCES, IN WHICH h_2 IS OUR CHOICE.

B. Feature Decomposition

Let N_j be the length of a CN, N_s be the number of candidate translation systems, a full hypothesis is defined as $\overrightarrow{Y} = y_1 \dots y_{N_j}$. We define a single upper case letter like Y as a variable, and define a lower case letter y as a taken value of variable Y.

Any feature f worked on $\overrightarrow{Y}$ could be decomposed into the summation of sub-features $f^i(\overrightarrow{Y})$ on i-th variable.

word posterior probability

One value y_i, namely one edge, may include a word w coming from different candidate translation systems. We assign an extra attribute to denote the word represented by value y_i from sys-th system as $y_i = \{y_i^{sys}\}$.

We define N_s features of word posterior probability as $f_1 \dots f_{Ns}$, and their corresponding weights as $\lambda_1 \dots \lambda_{N_s}$, each of which could be computed as

$$f^i_{sys}(\overrightarrow{Y}) = \begin{cases} \log f_{sys}(y_i^{sys}) & \text{if } y_i^{sys} \text{ exist} \\ None & \text{otherwise} \end{cases}$$

The $f_{sys}(y_i^{sys})$ is equivalent to word posterior probability $p(w|sys, i)$ mentioned in the background section.

Language Model

Take a string $\overrightarrow{Y} = s_0 s_1 s_2$ for example, suppose the language model order is 2, and there exist no value ε, then the expected feature score is as follows.

$$\begin{aligned} & f_{lm}(\overrightarrow{Y} = s_0 s_1 s_2) = \log P(s_0 s_1 s_2) \\ = \; & \log P(s_0) + \log P(s_1|s_0) + \log P(s_2|s_1) \\ = \; & f^0_{lm}(\overrightarrow{Y}) + f^1_{lm}(\overrightarrow{Y}) + f^2_{lm}(\overrightarrow{Y}) \end{aligned}$$

Then the feature fired on Y_i is defined as

$$f^i_{lm}(\overrightarrow{Y}) = \begin{cases} \log P(y_i| \dots y_{i-1}) & \text{if } y_i \neq \varepsilon \\ None & \text{otherwise} \end{cases}$$

Where $P(y_i| \dots y_{i-1})$ means taking enough context to compute language model score, where at most m_c windows including current position are considered.

Obviously, to ensure the accuracy of language model score, the language model order m_l is required no smaller than m_c, and in computing $P(y_i| \dots y_{i-1})$ there should be efficient context. One trick is enlarging the m_c.

Penalty for Loss of Language Model

Plenty of value ε would lead to errors in computing LM. Suppose $\overrightarrow{Y} = a_0\, \varepsilon\, b_2\, c_3\, \varepsilon\, \varepsilon\, d_6$, the language model order $m_l = 4$, the windows size $m_c = 4$. There are no losses for a_0,b_2 and c_3, but d_6. On 6-th position, mere c_3 can be

available in m_c windows, b_2 being out of the scope, thus the real score $\log(d_6|a_0 b_2 c_3)$ would be lost.

Since larger the m_c is, more computing is required. We simply add a penalty feature to supplement the losses.

$$is\_lost^i(\overrightarrow{Y}) = |\{y \in \{y_{i-m_c+1} \dots y_i\} | y \neq \epsilon\}| < m_l$$

$$f^i_{plm}(\overrightarrow{Y_{m_c}}) = \begin{cases} 1 & \text{if } is\_lost^i(\overrightarrow{Y}) \text{ and } y_i \neq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

word number and ε number

Let f_{wc} be the word count, namely none-ϵ value for any y_i, we have definition as follows.

$$f^i_{wc}(\overrightarrow{Y}) = \begin{cases} 0 & \text{if } y_i = \epsilon \\ 1 & \text{otherwise} \end{cases}$$

And let f_{nc} denote the ε edge count, then $f_{nc}(\overrightarrow{Y}) = N_s - f_{wc}(\overrightarrow{Y}) = N_s - \sum_i f^i_{wc}(\overrightarrow{Y})$.

IV. Evaluation

The candidate systems participating in the system combination are as listed in Table III: System A is a BTG-based system using a MaxEnt-based reordering model; System B is a hierarchical phrase-based system; System C is a Moses decoder; System D is a syntax-based system. 10-best hypotheses from each candidate system on the development and test sets are collected as the input of the system combination.

In our experiments, two different data sets are used. The first is to use NIST MT02 Chinese-to-English as the development set, and to use NIST MT05 for a test. The second is to use news portion in NIST MT06 Chinese-to-English as development set, and to use news portion in NIST MT2008 as a test. A 4-gram language model trained on Xinhua portion of Gigaword corpus are used. On two data sets, we used five baselines (four uni-CN based and one multi-CN based) ,all re-implemented following [11], [10], and be measured with case-sensitive BLEU score.

A. Comparison with MERT-based decoding

Our comparison consists of two parts, uni-CN based and multi-CN based system combination. In the first part, we choose skeleton from different candidate systems to construct uni-CN in turn, on which four baseline systems are trained, named as $B_{A,B,C,D}$ respectively. By contrast, four CRF-based systems are named as $C_{A,B,C,D}$. In the second part, baseline B_{mul} is a multi-CN based system [10], and our final system C_{mul} is to simply feed B_{mul} with four n-best lists from $C_{A,B,C,D}$ systems to complete a new system combination.

In the first data set, Table III, three CRF-based systems outperform respective baseline systems significantly, and one is a bit worse than B_A. Especally, the MERT-based B_B don't acquire a consistent result, but our C_B does. Our final system C_{mul} overpass a classic multi-CN based baseline system by 0.63 points. Note $C_{[ABCD]}$ only utilize the partial references instead of the full development set for training, thus we don't compare the BLEU with baselines in MT02.

SYSTEM	MT02(dev,%)	MT05(test,%)
A	31.85	30.25
B	32.16	**32.07**
C	32.11	31.71
D	33.37	31.26
B_A/C_A	34.69/-	**33.45**/33.36$^-$
B_B/C_B	34.57/-	33.19/**33.68**$^+$
B_C/C_C	30.85/-	29.17/**32.82**$^{++}$
B_D/C_D	34.00/-	32.34/**33.26**$^{++}$
B_{mul}/C_{mul}	35.48/36.25	34.04/ **34.67**$^+$

Table III
EXPERIMENTS ON MT02 AND MT05. ALL B_* ARE BASELINE SYSTEMS, AND C_* ARE OUR CRF-BASED SYSTEMS. $^{++}$SIGNIFICANCE AT 0.01 LEVEL, AND $^+$SIGNIFICANCE AT 0.05 LEVEL.

SYSTEM	MT06(news,dev,%)	MT08(news, test,%)
A	31.83	29.13
B	31.82	29.55
C	31.55	27.69
D	32.41	**30.16**
B_A/C_A	33.98/-	31.70/**32.07**$^+$
B_B/C_B	33.70/-	**31.83**/31.52$^-$
B_C/C_C	33.60/-	**30.02**/29.57$^-$
B_D/C_D	34.21/-	**31.75**/31.43$^-$
B_{mul}/C_{mul}	34.70/34.61	32.25/**32.37**

Table IV
EXPERIMENTS ON NEWS PORTION OF MT06 AND MT08. $^{++}$SIGNIFICANCE AT 0.01 LEVEL, AND $^+$SIGNIFICANCE AT 0.05 LEVEL.

In the second data set, Table IV, our CRF-based decoder don't go beyond the most results compared to baselines, but it delivers the similar performance, and would cost less training time shown in the next sub-section.

Our parameter settings are as follows, the minimal partial references length is 10, window size $m_c = 6$. The following content would demonstrate more experiments conducted on the first data set.

B. Effect of Minimal Partial References Length

We set the minimal length for partial references, because too small would lead to too much scrap-like objectives, and another extreme would not find efficient partial references. We adjust the minimal length, and Table V lists the different performances.

length	sys_A	sys_B	sys_C	sys_D
4	0.3303	0.3341	0.3259	0.3320
6	0.3314	0.3330	0.3285	0.3337
8	0.3310	0.3329	**0.3293**	**0.3341**
10	**0.3336**	0.3360	0.3282	0.3326
12	0.3304	**0.3365**	0.3249	0.3283

Table V
FLUCTUATION OF BLEU OF CRF-BASED DECODING WITH THE DIFFERENT MINIMAL PARTIAL REFERENCES LENGTH.

We show that different minimal length limitation does not cause to great fluctuation to the final quality measured by BLEU score.

C. Effect of Penalty for Language Model

As decomposing language model feature onto each variable Y_i would inevitably causes inaccuracy if there are plenty of ϵ value in $\overrightarrow{Y}$, we try to introduce the penalty feature. This experiment is to test the influence brought by this problem.

length	$-f_{plm}$	$+f_{plm}$(MT05, C_A)
8	0.2913	0.3310
10	0.2940	0.3336
12	0.2900	0.3304

Table VI
$-f_{plm}$ MEANS USING FEATURES EXCEPT f_{plm}, $+f_{plm}$ IS TO USE FULL FEATURES. WE USE CRF-BASED SYSTEM C_A AS OUR TEST TOOL.

From the data, we can see that, without the feature f_{plm}, CRF greatly suffers from the losses of language model caused by ε values. A step further, we conjecture CRF model may work better in other fields of machine translation in which circumstances there does not exist such a problem.

D. Effect of Window Size m_c

The language model feature f_{lm} relies on window size of context, m_c. Considering more context, there may be more accurate in calculating language model, the same time costing more. We tune this parameter to leverage final quality and time for training parameters.

m_c	BLEU(MT05, sys_A)	time
baseline B_A	0.3345	1.8 h
4	0.3010	1m 10s
5	0.3270	2m 23s
6	0.3336	4 m 21s
7	0.3340	$\geq$20m

Table VII
WHEN m_c BE SET NO LESS THAN 5, OUR MODEL ACQUIRE SIMILAR QUALITY, BUT WITH LESS TIME FOR TRAINING.

V. DETAILS AND CONCLUSION

We re-implement the CRF code to support real-value features, and make no modification to CRF itself. Our model use the similar features set as baseline systems, four system-specified word posterior probabilities, one language model, words number, ϵ number, in addition to a penalty feature for language model, compared to classic applications of CRF with millions of features. We find taking maximum likelihood and pseudo-likelihood as graphical inference principle acquire the similar performance, and *the* latter lead a more quick training speed for several folds. Due to the page limitation, readers could refer to [13] to learn details about CRF training.

As general machine translation tasks are explored more as a search-based problem, it is not a trivial thing to bring sophisticated machine learning models into this area. This paper attempts to solve the objective ambiguity problem in MERT frame by proposing a novel objective, partial reference, and casting decoding a confusion network as a sequence labeling problem, then borrow traditional graphical model CRF to train optimal parameters. More, in uni-CN based system combination tasks, our CRF-based systems could acquire better or similar results, and with less training time. Our work show a promise of introducing more sophisticated machine learning techniques into MT field to improve translation quality a step further.

ACKNOWLEDGMENT

The authors are supported by National Natural Science Foundation of China, Contracts 60873167 and 60736014. We would like to thank Yang Liu for valuable comments, and the anonymous reviewers for helpful suggestions.

REFERENCES

[1] Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. *A discriminative latent variable model for statistical machine translation*. In *Proc. of ACL*.

[2] Khe Chai Sim, William J. Byrne, Mark J.F. Gales, Hichem Sahbi, and Phil C. Woodland. *Consensus network decoding for statistical machine translation system combination*. 2007, In *Proc. of ICASSP*.

[3] Xiaodong He, Mei Yang, Jangfeng Gao, Patrick Nguyen and Robert Moore. 2008. *Indirect-HMM-based Hypothesis Alignment for Computing Outputs from Machine Translation Systems. Proc. of EMNLP.*

[4] Xiaodong He and Kristina Toutanova. 2009. *Joint optimization for machine translation system combination*. In *Proc. of EMNLP*.

[5] Fei Huang and Kishore Papineni. 2007. *Hierarchical System Combination for Machine Translation*. In *Proc. of EMNLP*.

[6] Phillip Koehn, Franz Josef Och, and Daniel Marcu. 2003. *Statistical phrase-based translation*. In *Proc. of NAACL.*

[7] Chi-Ho Li, Xiaodong He, Yupeng Liu and Ning Xi. 2009. *Incremental HMM Alignment for MT System Combination*. In *Proc. of ACL.*

[8] Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. *Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment*. In *Proc. of IEEE EACL.*

[9] Franz Josef Och. 2003. *Minimum error rate training in statistical machine translation*. In *Proc. of ACL.*

[10] Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007a. *Improved word-level system com- bination for machine translation*. In *Proc. of ACL.*

[11] Antti-Veikko I. Rosti, Bing Xiang, Spyros Matsoukas, Richard Schwartz, Necip Fazil Ayan, and Bonnie J. Dorr. 2007b. *Combining outputs from multiple machine translation systems*. In *Proc. of NAACL-HLT*.

[12] Antti-Veikko I. Rosti, Bing Zhang, Spyros Matsoukas, and Richard Schwartz. 2008. *Incremental hypothesis alignment for building confusion networks with application to machine translation system combination*. In *Proc. of the Third ACL WorkShop on Statistical Machine Translation*.

[13] Charles Sutton and Andrew McCallum. *An introduction to conditional random fields for relational learning*. In *MIT press.*