Conditional Random Fields for Machine Translation System Combination

Tian Xia, Shandian Zhe, Jinsong Su, Qun Liu
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{xiatian,sujinsong, 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 regarding to the training goal and it is hard to tackle for MERT, that is different parameters may lead to the same minimum error rate in training but greatly different performances in test data. We propose a novel training objective as the unique goal for training towards, namely partial reference translation, and by use of conditional random fields 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;

1 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 construction (Matusov et al., 2006; Rosti et al., 2007; Rosti et al., 2007; He et al., 2008; He and Toutanova 2009), and decoding an optimal path evaluated with a set of features. In Table 1, hypotheses are aligned to *h0*, and corresponding confusion network refers to Figure 1.

h0	He	feels	to	apples
h1	He	prefer	ε	apples
h2	He	ε	like	apples
h3	Him	prefer	to	apples

Table 1. Suppose *h0* is skeleton hypothesis, to which others be aligned pair-wisely.

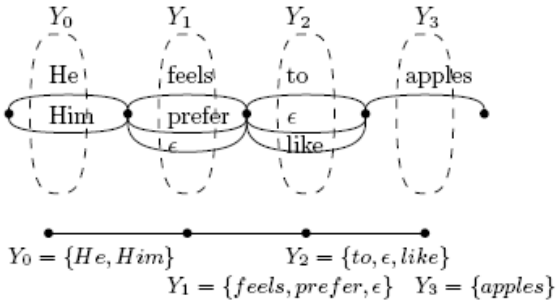

Figure1. The above graph is about a confusion network, and to be casted as a sequence labeling problem shown in the below graph.

Training algorithm on confusion networks following Minimum Error Rate Training (MERT) (Och 2003; Koehn et al., 2003) 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 possible 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 (Blunsom et al.,2008), utilizes the reachable references[1] for CRF training. However, it is impossible to choose reachable confusion networks to train, because most confusion networks does not generate results fully matching the reference translations, so the available number of confusion networks is too poor to waste. Thus, we propose a novel objective, *partial reference translation*, as the unique objective for each confusion network to train towards. The *partial reference translation* is defined as the optimal sub-string of reference translations, which 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 their respective span, a simple graphical model is naturally generated.

[1] The ``Reachable" means reference translations could be generated by a model regardless of parameters. In our application, reachable references should be decoded from a confusion network.

We adopt conditional random fields to train our model on uni-confusion network due to an important reason, CRF model could train a global optimal solution (Sutton and McCallum 2006; Lafferty et al., 2001). 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 make comparisons on the task of multi-confusion network based system combination. Our method is firstly to collect the *n*-best hypotheses from CRF-based systems, then to feed a common multi-confusion network based system to complete a full system combination procedure.

2 Background

2.1 Confusion Network and MERT

Formally, confusion network is a directed, acyclic graph with unique source vertex and sink vertex. On each edge of graph, there is one alternative word attached to, including a special place-holder ε denoting no concrete word.

The skeleton hypothesis (also called backbone hypothesis) determines the words order in final translation, eg. *h0* in Table 1. Constructing a confusion network, all hypotheses are aligned to the skeleton (Rosti et al., 2007; He et al., 2008; Matusov et al., 2006) or to partially constructed confusion network (Li et al., 2009; Rosti et al., 2008).

In order to reduce the risk of mis-choosing skeleton hypothesis, multi-confusion network based system combination was developed, which choose respective skeleton for each candidate system. Multi-confusion network based system combination may generate potential better-quality translations than uni-confusion network based system combination.

Training parameters in system combination follows Minimum Error Rate Training (MERT) firstly proposed by Och (Och 2003). The whole produce bases on iteration and does not stop until predefined times or system converges. In each round, decoder searches an *n*-best hypothesis list for each sentence in the development data set, and MERT predicts the optimal parameters having the minimum error rate.

2.2 Conditional Random Fields

Conditional random fields (CRFs) are undirected graphical models trained to maximize a conditional probability (Lafferty et al., 2001; Sutton and McCallum 2006). A common special-case graph structure is a linear chain, which corresponds to a finite state machine, and is suitable for sequence labeling. A linear-chain CRF with parameters $\Lambda = \{\lambda_1 \ldots\}$ defines a conditional probability for a state (label) sequence $Y = y_1..y_N$ (for example, POS labels) given an input sequence $X = x_1..x_N$ (for example, the characters of a Chinese sentence) to be

$$P_{\Lambda}(Y \mid X) = \frac{1}{Z_x} \exp\{\sum_{t=1}^{N} \sum_{k} \lambda_k f_k(y_{t-1}, y_t, X, t)\}$$

where Zx is the per-input normalization that makes the probability of all state sequences sum to one; $f_k(y_{t-1}, y_t, X, t)$ is a feature function which is often binary-valued, also can be real-valued, and λ_k is a learned weight associated with feature f_k. The feature functions

measure any aspect of a state transition, from y_{t-1} *to* y_t, and the entire observation sequence, X, centered at the current time step, t.

The parameters can be estimated by maximum likelihood principle, maximizing the conditional probability of a set of label sequences, each given their corresponding input sequences. The log-likelihood of training set $\{X, Y\}$ is written

$$\begin{aligned} \ell_{\wedge} &= \sum_{X,Y} \log P_{\lambda}(Y \mid X) \\ &= \sum_{X,Y} (\sum_{t} \sum_{k} \lambda_k f_k(y_{t-1}, y_t, X, t)) \end{aligned}$$

As the log-likelihood function is convex, this guarantees that every local maximum is a global maximum. Our implementation uses a quasi-Newton gradient-climber BFGS for optimization, which has been shown to converge much faster. The gradient of feature weight λ_k is

$$\begin{aligned} &\sum_{X,Y,t} f_k(y_{t-1}, y_t, X, t) \\ &- \sum_{X,Y,t} P(M \mid X) f_k(z_{t-1}, z_t, X, t) \end{aligned}$$

where M is one of possible state sequences generated in input data X.

The most probable label sequence for an input X can be efficiently searched using Viterbi algorithm (Rabiner 1990).

$$Y^* = \underset{Y}{\operatorname{argmax}}\ P_{\wedge}(Y \mid X)$$

To control over-fitting, we regularize the parameters with a Gaussian prior of 1, which is also be viewed as *L2* regularization.

2.3 Features

Features used in our work and baseline systems are nearly the same as (Rosti et al., 2007a; He et al., 2008), 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 is assigned as *1/(k+1)*, and then it is normalized by the sum from the same sys and span.

2) logarithm of language model score, *Lm*.

3) ε value number, *Num(ε)*.

4) words number, *Num(w)*.

$$\log(h) = \sum_{span} \log(\sum_{sys} \lambda_{sys}\, p(w \mid sys, span)) + w_0 Lm(h) + w_1 Num(\varepsilon) + w_2 Num(w)$$

3 CRF-based Training on System combination

MERT aims to minimize errors in a development data set (or maximize BLEU metric in machine translation and system combination area). Our intuition is, good translations should have a high BLEU score, like reference translations, but high-BLEU translations may not be regarded as good ones judged by the people. In the introduction part, our interesting trials show that two sets of parameters may lead to the same BLEU in training, but greatly different performance in test. Since MERT does not discriminate this case and, as an approximation algorithm, can not find the global optimal solution, we conjecture empirically that training towards better objectives using a stable model, like CRF, may lead to better results. We also guess it is one of reasons for Blunsom (Blunsom et al., 2008) to get a success by selecting reachable reference translations exclusively. Moreover, MERT is very time-consuming because it iterates for several rounds, in each round the decoder is called to decode all the sentences.

The biggest barrier for a probabilistic model to be used here is there are no determined and unique objective hypotheses like other NLP tasks, like parsing, POS tagging. The language model feature is also a challenge for exact inference in the probabilistic models. Our following subsections address these problems.

3.1 Partial Reference Translation

We enumerate all the configurations of a confusion network to match the n-gram in reference translations. The partial confusion network, including the optimal fragments of reference translations, is kept for training, and the remnant are thrown away. Note that, there are usually four reference translations for each source sentence, while our model merely choose as the training goal one optimal fragments of them.

Since any variable *Yi* might take a value ε, it is important to decide whether our model should encourage to generate more ε or less in partial references. There are several alternative rules.

1) Treating ε with others value with no difference, find the longest fragments.

2) Make fragments as longer as possible, requiring no ε in two ends.

3) Considering the second rule preferentially, then permit as many as possible ε in two ends.

Table 2 describe three examples respective to above rules. Suppose both “a b” and “a b c” are part of reference translations.

h1	ε	ε	a	b	ε	ε	ε	ε	ε
h2			a	b	ε	c			
h3		ε	a	c	ε	c	ε	ε	

Table 2. Three valid hypotheses on a confusion network, one of them is expected as our training objective.

3.2 Feature Decomposition

Let *Nj* be the length of a confusion network, *N*s be the number of candidate translation systems, a full hypothesis is defined as $Y = y_1..y_N$. We define a lower case letter *y* as a taken value of a special variable in *Y*.

Any feature f worked on Y could be decomposed into the summation of sub-features $f^i(Y)$ on *i*-th variable (or position).

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

$$f^{i}{}_{sys}(Y) = \begin{cases} \log\ f_{sys}(y^{sys}{}_{i}) & \text{if } y^{i}{}_{sys} \text{ exist} \\ \text{None} & \text{otherwise} \end{cases}$$

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

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

Then the feature fired on Y_i is defined as

$$f_{lm}{}^{i}(Y) = \begin{cases} \log P(y_i \mid ..y_{i-1}) & \text{if } y_i \neq \varepsilon \\ \text{None} & \text{otherwise} \end{cases}$$

Where $P(y_i| ..y_{i-1})$ 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 $P(y_i|..y_{i-1})$ 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 $Y = a_0\ \varepsilon\ b_2\ c_3\ \varepsilon\ \varepsilon\ d_6$, 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, only c_3 can be available in m_c windows respective to d_6, with 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.

$$is\_lost^{i}(Y) = |\{y \in \{y_{i-mc+1}...y_i\} \mid y \neq \varepsilon\}| < m_l$$

$$f_{plm}{}^{i}(Y_{mc}) = \begin{cases} 1 & \text{if } is\_losti(Y) \text{ and } y_i \neq \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

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.

$$f_{wc}{}^{i}(Y) = \begin{cases} 0 & \text{if } y_i \neq \varepsilon \\ 1 & \text{otherwise} \end{cases}$$

And let f_{nc} denote the count of ε, then

$$f_{nc}(Y) = N_s - f_{wc}(Y) = N_s - \sum_i f_{wc}{}^{i}(Y)$$

4 Related Work

Liang (Liang et al., 2006) used a perceptron-style discriminative approach to machine translation. Liang tried different training objective, local update and global update. The latter also face the problem of being unreachable for the reference translations. A large number of n-gram features contributes to final translation in his work, while we only use basic feature set.

Blunsom (Blunsom et al., 2008) utilized a tree-like CRF to challenge MERT framework. His work is demanding in data scale, reachable reference translations.

5 Evaluation

The candidate systems participating in the system combination are as listed in Table 3: 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.

Two different data sets are used in the experiments. 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 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-confusion network based and one multi-confusion network based) ,all re-implemented following (Rosti et al., 2007a 2007b), and be measured with case-sensitive NIST BLEU score.

In our trials, the second rule of selecting partial reference translation brings a bit of better and consistent results over other two, so our following comparisons are under this configuration.

5.1 Comparisons with MERT-based Decoding

Our comparisons consist of two parts, uni-confusion network based and multi-confusion network based system combination. In the first part, we choose skeleton from different candidate systems to construct uni-confusion network 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-confusion network based system, 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 3, three CRF-based systems outperform respective baseline systems significantly, and one is a bit worse than B_A. Especially, the MERT-based B_B does not obtain a consistent result, while CRF-based CB does. Our final system C_{mul} overpass a classic multi-confusion network based baseline system by 0.63 points. Note $C_{A,B,C,D}$ only utilize the partial training data instead of the full development set, thus we do not 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 3. 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.

In the second data set, Table 4, 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.

SYSTEM	MT06(news, dev, %)	MT08(news, %)
A	31.83	29.13
B	31.82	29.55
C	31.55	27.69
D	34.41	**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 4. Experiments on news portion of mt06 and mt08. ++significance at 0.01 level, and +significance at 0.05 level.

5.2 Available Partial Reference Translations

In practice, it would be demanding of requiring reference translation to be reachable in each test sentence.

From Table 5, there are only 12 and 8 out of 878 and 616 sentences for fully matched reference translations, which make necessary for using partial reference translations.

	MT02	MT06(news)
total number	878	616
fully matched	12	8
6	602	474
8	339	311
10	172	184
12	69	80

Table 5. Available number of partial reference translations with different minimal length.

5.3 Effect of Minimum Length of Partial Reference Translations

We hope to set suitable minimum length for partial reference translations. On one hand, the limitation is relaxed enough, so many scrap-like objectives may do harm to the training and be a waste of time. On another hand, there would no sufficient data to ensure efficient training. Table 6 lists the performances with different limitations.

length	A	B	C	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 6. Fluctuation of bleu of crf-based decoding with the different minimal partial references length.

Adjusting the minimum length from 4 to 12, the differences between maximum BLEU and minimum BLEU for four single training are 0.33%, 0.35%, 0.44%, 0.58%. We conclude this factor does not cause great fluctuation to the translation quality measured by BLEU score. In practice, we set a value of 10 in order to get a balance between program efficiency and translation quality.

5.4 Effect of Penalty for Language Model

As decomposing language model onto each variable Y_i would causes inaccuracy inevitably if there are plenty of ε value in Y, thus we try to introduce the penalty feature to supplement the losses

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

Table 7. $-f_{plm}$ 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 Table 7, without the feature f_{plm}, CRF greatly suffers from the losses of language model caused by ε values, and this feature would contribute as high as 4% average improvements to the final translations in BLEU score. A step further, we conjecture CRF model may work better in other applications of machine translation area where language model feature can be computed exactly.

5.5 Effect of Window Size m_c

This parameter causes great influence to the computing of language model feature f_{plm}. As our experiments use 4-gram language model, m_c is set no smaller than 4. Due the inaccuracy on language model score brought by ε, we should consider moderately bigger setting to leverage depends on window size of context, m_c. Considering more context, there may be more accurate in calculating language model, as well as taking more time. We tune this parameter to leverage final quality and time for training parameters.

m_c	BLEU(MT05)	time
baseline	0.3345	1.8h
4	0.3010	1m 10s
5	0.3270	2m 23s
6	0.3336	4m 21s
7	0.3340	> 20m

Table 8. When mc be set no less than 5, our model acquire similar quality, but with less time for training.

6 Details and Conclusion

We re-implement a CRF code to support real-value features (like language model score), and make no modification to CRF itself. Compared to classic applications of CRF with millions of features, our application only use several features, those are similar to the baseline systems, four system-specified word posterior probabilities, one language model, words number, ε number, and a penalty feature for language model. We find taking maximum likelihood and pseudo-likelihood as graphical inference principle acquires similar

performance in BLEU metric, while the latter behaves better in training speed by several folds.

Machine translation is a special problem in natural language processing area, no clear and definite reference goals, very hard to measure the translation quality, quite huge for the solution space. As a result, it is not a trivial thing to bring sophisticated machine learning models into this area. This paper attempts to solve the objective ambiguity in MERT frame. We propose a novel objective, partial reference translation, and cast decoding a confusion network as a sequence labeling problem, then borrow classic graphical model CRF to train optimal parameters. Our CRF-based systems obtain better or similar translation quality compared to MERT based systems in different data sets, and take less time for training uni-confusion network based systems.

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.

7 REFERENCES

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

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.

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.

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

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

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, Ben Taskar. 2006. An End-to-End Discriminative Approach to Machine Translation. In Proc. of ACL.

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

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

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proc. of ICML.

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.

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

Fuchun Peng, Fangfang Feng, Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In Proc. of COLING.

L. Rabiner. 1990. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In Alex Weibel and Kay-Fu Lee, editors, Readings in Speech Recognition.

Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007a. Improved word-level system combination for machine translation. In Proc. of ACL.

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.

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.

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. In MIT press.