MANY : Open Source MT System Combination at WMT'10

Loïc Barrault
LIUM, University of Le Mans
Le Mans, France.
FirstName.LastName@lium.univ-lemans.fr

Abstract

LIUM participated in the System Combination task of the Fifth Workshop on Statistical Machine Translation (WMT 2010). Hypotheses from 5 French/English MT systems were combined with MANY, an open source system combination software based on confusion networks currently developed at LIUM.

The system combination yielded significant improvements in BLEU score when applied on WMT'09 data. The same behavior has been observed when tuning is performed on development data of this year evaluation.

1 Introduction

This year, the LIUM computer science laboratory has participated in the French-English system combination task at WMT'10 evaluation campaign. The system used for this task is MANY[1] (Barrault, 2010), an open source system combination software based on Confusion Networks (CN). Several improvements have been made in order to being able to combine many systems outputs in a decent time.

The focus has been put on the tuning step, and more precisely how to perform system parameter tuning. Two methods have been experimented corresponding to two different representations of system combination. In the first one, system combination is considered as a whole : fed by system hypotheses as input and generating a new hypothesis as output. The second method considers that the alignment module is independent from the decoder, so that the parameters from each module can be tuned separately.

Those tuning approaches are described in section 3. Before that, a quick description of MANY, including recent developments, can be found in section 2. Results on WMT'09 data are presented in section 4 along results of tuning on newssyscombtune2010.

2 System description

MANY is a system combination software (Barrault, 2010) based on the decoding of a lattice made of several Confusion Networks (CN). This is a widespread approach in MT system combination (Rosti et al., 2007); (Shen et al., 2008); (Karakos et al., 2008). MANY can be decomposed in two main modules. The first one is the alignment module which actually is a modified version of TERp (Snover et al., 2009). Its role is to incrementally align the hypotheses against a backbone in order to create a confusion network. Those confusion networks are then connected together to create a lattice. This module uses different costs (which corresponds to a match, an insertion, a deletion, a substitution, a shift, a synonym and a stem) to compute the best alignment and incrementally build a confusion network. In the case of confusion network, the match (substitution, synonyms, and stems) costs are considered when the word in the hypothesis matches (is a substitution, a synonyms or a stems of) at least one word of the considered confusion sets in the CN, as shown in Figure 1.

The second module is the decoder. This decoder is based on the token pass algorithm and it accepts as input the lattice previously created. The probabilities computed in the decoder can be expressed as follow :

$$\begin{aligned} log(P_W) \quad = \quad & \sum_{n=0}^{Len(W)} \Big\{ \alpha_1 log P_{ws}(n) + \alpha_2 log P_{lm}(n) \\ & + \alpha_3 L_{pen}(n) + \alpha_4 N_{pen}(n) \Big\} \end{aligned} \tag{1}$$

where $Len(W)$ is the length of the hypothesis,

[1] MANY is available at the following address http://www-lium.univ-lemans.fr/~barrault/MANY

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 271–275,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

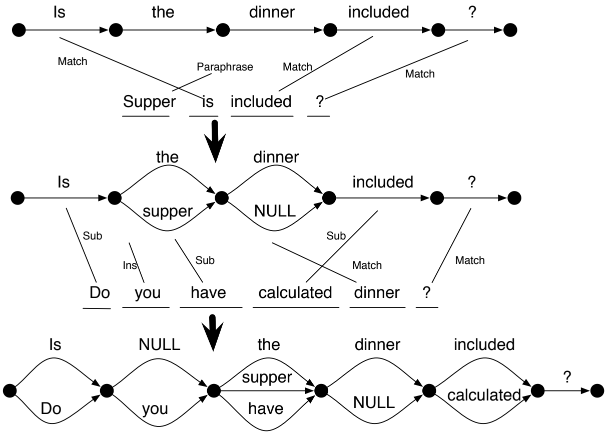

Figure 1: Incremental alignment with TERp resulting in a confusion network.

$P_{ws}(n)$ is the score of the n^{th} word in the lattice, $P_{lm}(n)$ is its LM probability, $L_{pen}(n)$ is the length penalty (which apply when W_n is not a null-arc), $N_{pen}(n)$ is the penalty applied when crossing a null-arc, and the α_i are the features weights.

Multithreading

One major issue with system combination concerns scaling. Indeed, in order to not lose information about word order, all system hypotheses are considered as backbone and all other hypotheses are aligned to it to create a CN. Consequently, if we consider N system outputs, then to build N confusion networks, $N * (N - 1)$ alignments with modified TERp have to be performed. Moreover, in order to get better results, the TERp costs have to be optimized, which requires a lot of iterations, all of which calculate $N * (N - 1)$ alignments. However, the building of a CN with system i as backbone does not depend on the building of CN with other system as backbone. Therefore multithreading has been integrated into MANY so that multiple CNs can be created in parallel. From now on, the number of thread can be specified in the configuration file.

3 Tuning

As mentioned before, MANY is made of two main modules : the alignment module based on a modified version of TERp and the decoder. Considering 10 systems, 19 parameters in total have to be optimized in order to get better results. By default, TERp costs are set to 0.0 for match and 1.0 for everything else. These costs are not correct, since a shift in that case will hardly be possible. TERp costs, system priors, fudge factor, null-arc penalty, length penalty are tuned with Condor (a global optimizer based on the Powell's algorithm, (Berghen and Bersini, 2005)).

Two ways of tuning have been experimented. The first one consists in optimizing the whole set of parameters together (see section 3.1). The second one rely on the (maybe likely) independence of the TERp parameters towards those of the decoder and consists in tuning TERp parameters in a first step and then using the optimized TERp costs when tuning the decoder parameters (see section 3.2).

3.1 Tuning all parameters together

Condor is an optimizer which aims at minimizing a certain objective function. In our case, the objective function is the whole system combination. As input, it takes the whole set of parameters (*i.e.* TERp costs except match costs (which is always set to 0), system priors, the fudge factor, and null-arc and length penalty) and outputs -BLEU score. The BLEU score is one of the most robust metrics as presented in (Leusch et al., 2009), which is consequently an obvious target for optimization.

Such a tuning protocol has the disadvantage to be slower as all the confusion networks have to be regenerated at each step because the TERp costs provided by the optimizer will hardly be the same for two iterations (thus, confusion networks computed during previous iterations can hardly be reused). Another issue with this approach is that it is hard to converge when the parameter set is that large. This is mainly due to the fact that we cannot guarantee the convexity of the problem. However, one advantage is that the possible correlation between all parameters are taken into account during the optimization process, which is not the case when optimizing in several steps.

3.2 Two-step tuning

Tuning TERp parameters : In order to optimize TERp parameters (*i.e.* del, ins, sub, shift, stem and syn costs), we have to determine which measure to use to evaluate a certain configuration. We naturally considered the minimization of the TERp score. To do so, the confusion networks are built using the set of parameters given by the optimizer. TERp scores are then calculated between the reference and each CN, and summed up.

The goal of this step is to guide the confusion networks generation process to produce sentences similar to the reference. Consequently, if the confusion networks generated at this step have a lower TERp score, then this means that the decoder is more likely to find a better hypothesis inside.

Tuning decoder parameters : Based on the TERp configuration determined at the previous step, this step aims at finding good parameter values. Those parameters control the final hypothesis size and the importance given to the language model probabilities compared to the translation scores (occurring on words). The metric which is minimized is -BLEU for the same reasons mentioned in section 3.1.

4 Experiments and Results

During experiments, data from last year evaluation campaign are used for testing the tuning approach. news-dev2009a is used as development set, and news-dev2009b as internal test, these corpora are described in Table 1.

NAME	#sent.	#words	#tok
news-dev2009a	1025	21583	24595
news-dev2009b	1026	21837	24940

Table 1: WMT'09 corpora : number of sentences, words and tokens calculated on the reference.

For the sake of speed and simplicity, the five best systems (ranking given by score on dev) are considered only. Baseline systems performances on dev and test are presented in Table 2.

Corpus	Sys0	Sys1	Sys2	Sys3	Sys4
Dev	18.20	17.83	20.14	21.06	17.72
Test	18.53	18.33	20.43	21.35	18.15

Table 2: Baseline systems performance on WMT'09 data (%BLEU).

When tuning all parameters together, the set obtained is presented in Table 3. The 2-step tuning protocol applied on news-dev2009a provides the set of parameters presented in Table 4.

Costs :	Del	Stem	Syn	Ins	Sub	Shift
	0.89	0.94	1.04	0.98	0.94	0.94
Dec. :	Fudge		Null_{pen}		Len_{pen}	
	0.01		0.25		1.46	
Weights :	Sys0	Sys1	Sys2	Sys3	Sys4	
	0.04	0.04	0.16	0.26	0.04	

Table 3: Parameters obtained with 1-step tuning.

Costs :	Del	Stem	Syn	Ins	Sub	Shift
	9e-6	0.89	1.22	0.26	0.44	1.76
Dec. :	Fudge		Null_{pen}		Len_{pen}	
	0.1		0.27		2.1	
Weights :	Sys0	Sys1	Sys2	Sys3	Sys4	
	0.07	0.09	0.09	0.09	0.11	

Table 4: Parameters obtained with 2-step tuning.

Results on development corpus of WMT'09 (used as test set) are presented in Table 5. We can observe that 2-step tuning provides almost 0.9 BLEU point improvement on development corpus which is well reflected on test set with a gain of more than 0.7 BLEU. The best results are obtain when tuning all parameters together, which give more than 1 BLEU point improvement on dev and more than 0.9 on test.

System	Dev	Test
Best single	21.06	21.35
MANY	**22.08**	**22.28**
MANY-2steps	**21.94**	**22.09**

Table 5: System Combination results on WMT'09 data.

4.1 Discussion

Choosing a measure to optimize the TERp costs is not something easy. One important remark is that default (equal) costs are not suitable to get good confusion networks. The goal of the confusion networks is to make possible the generation of a new hypothesis which can be different from those provided by each individual system.

In these experiments, TERp calculated between the CNs and the reference is used as the distance to be minimized by the optimizer. We can notice that for the 2-step optimization, the deletion cost is very small. This is probably not a value which is expected, because in this case, this means that deletions can occur in an hypothesis without penalizing it a lot. However, this parameter set has a beneficial impact on the system combination performance. Another comment is that the system weights are not directly proportional to the results. This suggests that some phrases proposed by weaker systems can have a higher importance for system combination.

By contrast, optimizing parameters all together provides more fair weights, according to the results of the single systems.

4.2 2010 evaluation campaign

For this year system combination tasks, a development corpus (syscombtune) and the test (syscombtest), described in Table 6, were provided to participants.

NAME	#sentences	#words	#words tok
syscombtune	455	9348	10755
syscombtest	2034	-	-

Table 6: Description of WMT'10 corpora.

Language model : The English target language models has been trained on all monolingual data provided for the translation tasks. In addition, LDC's Gigaword collection was used for both languages. Data corresponding to the development and test periods were removed from the Gigaword collections.

Tuning on syscombdev2010 corpus produced the parameter set presented in Table 7

Costs :	Del	Stem	Syn	Ins	Sub	Shift
Dec. :	Fudge		Null_{pen}		Len_{pen}	
	0.01		0.33		1.6	
Weights :	Sys0	Sys1	Sys2	Sys3	Sys4	
	0.11	0.21	0.04	0.15	0.15	

Table 7: Parameters obtained with tuning.

The result provided by the system with this configuration can be compared to the single systems in Table 8.

System	newssyscombtune2010
Sys0	27.74
Sys1	27.26
Sys2	27.15
Sys3	27.06
Sys4	27.04
MANY	**28.63**

Table 8: Baseline systems performance on WMT'10 development data (%BLEU).

A behavior comparable to WMT'09 evaluation campaign is observed, which suggests that the approach is correct.

5 Conclusion and future work

We have shown that tuning all parameters together is better than 2-step tuning. However, the second method has not been fully explored. Tuning TERp parameters targeting minimum TERp score is not satisfying. Therefore, an alternative measure, like ngram agreement which would be more related to BLEU, can be considered in order to obtain better parameters.

Further improvement for MANY will be considered like case insensitive combination then recasing the output using majority vote on the confusion networks. This is currently a work in progress.

6 Acknowledgement

This work has been partially funded by the European Union under the EuroMatrix Plus project (http://www.euromatrixplus.net, IST-2007.2.2-FP7-231720)

References

Barrault, L. (2010). MANY : Open source machine translation system combination. *Prague Bulletin of Mathematical Linguistics, Special Issue on Open Source Tools for Machine Translation*, 93:147–155.

Berghen, F. V. and Bersini, H. (2005). CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175.

Karakos, D., Eisner, J., Khudanpur, S., and Dreyer, M. (2008). Machine translation system combination using ITG-based alignments. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.*, pages 81–84, Columbus, Ohio, USA.

Leusch, G., Matusov, E., and Ney, H. (2009). The RWTH system combination system for WMT 2009. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 61–65, Athens, Greece.

Rosti, A.-V., Matsoukas, S., and Schwartz, R. (2007). Improved word-level system combination for machine translation. In *Association for Computational Linguistics*, pages 312–319.

Shen, W., Delaney, B., Anderson, T., and Slyh, R. (2008). The MIT-LL/AFRL IWSLT-2008 MT System. In *International Workshop on Spoken Language Translation*, Hawaii, U.S.A.

Snover, M., Madnani, N., Dorr, B., and Schwartz, R. (2009). TER-Plus: Paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation Journal*.