DCU Confusion Network-based System Combination for ML4HMT

Tsuyoshi Okita
Dublin City University
Glasnevin, Dublin 9, Ireland
tokita@computing.dcu.ie

Josef van Genabith
Dublin City University
Glasnevin, Dublin 9, Ireland
josef@computing.dcu.ie

Abstract

This paper describes a system combination module in the MaTrEx (Machine Translation using Examples) MT system developed at Dublin City University. We deployed this module to the evaluation campaign for the ML4HMT task, achieving an improvement of 2.16 BLEU points absolute and 9.2% relative compared to the best single system.

1 Introduction

This paper describes a system combination module in the MaTrEx (Machine Translation using Examples) MT system (Du et al., 2009; Okita et al., 2010b) developed at Dublin City University. We deployed this module to the evaluation campaign for the ML4HMT task.

System combination techniques often rely on the Minimum Bayes Risk decoder (MBR decoder) (Kumar and Byrne, 2002) with and without confusion network. Our system combination approach uses the MBR decoder with the confusion network (Bangalore et al., 2001; Matusov et al., 2006; Du et al., 2009). One notable addition in this paper is in the optimization procedure (presented in Section 2) which considers all the possible combinations of given inputs and may result in excluding the outputs of some of the systems participating in system combination architecture. As far as we know, there is no paper yet which discusses in detail how to best select from the provided set of single best translations. This paper also seeks to explain the mechanism why this selection works.

The alternative approach which does not use the confusion network tends to address the problem when the MBR decoder has to handle larger n in its n-best lists (Tromble et al., 2008; DeNero et al., 2009).

The remainder of this paper is organized as follows. Section 2 describes the system combination strategy we used in this evaluation campaign. In Section 3, our experimental results are presented. In Section 4, we discuss why one inferior system is better removed in the overall system combination strategy. We conclude in Section 5.

2 Our System Combination Strategy

Let E be the target language, F be the source language, and $M(\cdot)$ be an MT system which maps some sequence in the source language F into some sequence in the target language E. Let $\mathcal{E}$ be the translation outputs of all the MT systems. For a given reference translation E, the decoder performance can be measured by the loss function $L(E, M(F))$. Given such a loss function $L(E, E')$ between an automatic translation E' and the reference E, a set of translation outputs $\mathcal{E}$, and an underlying probability model $P(E|F)$, a MBR decoder is defined as in (1) (Kumar and Byrne, 2002):

$$\begin{aligned} \hat{E} &= \arg\min_{E' \in \mathcal{E}} R(E') \\ &= \arg\min_{E' \in \mathcal{E}} \sum_{E' \in \mathcal{E}} L(E, E') P(E|F) \end{aligned} \quad (1)$$

where $R(E')$ denotes the Bayes risk of candidate translation E' under the loss function L. We use BLEU (Papineni et al., 2002) as this loss function L.

system	MT output seqs				prob	expected matches
1	a	a	a	c	0.30	expected-matches(aaac)=0.3*4+0.2*0+0.2*0+0.2*0+0.1*0=1.2
2	b	b	c	d	0.20	expected-matches(bbcd)=0.3*0+0.2*4+0.2*3+0.2*3+0.1*1=2.1
3	b	b	b	d	0.20	expected-matches(bbbd)=0.3*0+0.2*3+0.2*4+0.2*2+0.1*2=2.0
4	b	b	c	f	0.20	expected-matches(bbcf)=0.3*0+0.2*3+0.2*2+0.2*4+0.1*0=1.8
5	f	f	b	d	0.10	expected-matches(ffbd)=0.3*0+0.2*1+0.2*2+0.2*0+0.1*4=1.0

system	MT output seqs				prob	expected matches
1	a	a	a	c	0.33	expected-matches(aaac)=0.33*4+0.22*0+0.22*0+0.22*0+0.00*0=1.32
2	b	b	c	d	0.22	expected-matches(bbcd)=0.33*0+0.22*4+0.22*3+0.22*3+0.00*1=**2.20**
3	b	b	b	d	0.22	expected-matches(bbbd)=0.33*0+0.22*3+0.22*4+0.22*2+0.00*2=1.98
4	b	b	c	f	0.22	expected-matches(bbcf)=0.33*0+0.22*3+0.22*2+0.22*4+0.00*0=1.98
5	-	-	-	-	0.00	

Table 1: Motivating examples. MBR decoding can be schematically described as the expectation of the number of matching between the MT output sequence and some sequence, as is described in this table. The upper row shows the MT output sequences consisting of 5 systems, while the lower row shows the MT output sequences consisting of 4 systems. In this case, the expected matches of “bbcd” for 4 systems (lower row) are better than those for 5 systems (upper row). This suggests that it may be better to remove extremely bad MT output from the inputs of system combination.

We now introduce the idea of searching for the optimal subset $\mathcal{E}_0$ among $\mathcal{E}$ (where $\mathcal{E}$ is the translation outputs of all the MT systems participating in the system combination). The motivating example is shown in Table 1. In this example, five MT output sequences “aaac”,”bbcd”,”bbbd”,”bbcf”, and “ffbd” are given. Suppose that we calculate the expected matches of “bbcd”, which constitute the negative quantity in Bayes risk. If we use all the given MT outputs consisting of 5 systems, the expected matches sum to 2.1. If we discard the system producing ”ffbd” and only use 4 systems, the expected matches improve to 2.20. As a conclusion, it is not always the best solution to use the full set of given MT outputs, but to remove some MT output can be a good strategy. This suggests to consider all the possible subsets of the full set of MT outputs, as is shown in (2):

$$\hat{E} = \arg\min_{\mathcal{E}_i \subseteq \mathcal{E}} \sum_{E' \in \mathcal{E}_i} L(E, E')P(E|F) \quad (2)$$

where $\mathcal{E}_0 \subseteq \mathcal{E}$ indicates that we choose $\mathcal{E}_0$ from all the possible subsets of $\mathcal{E}$ (or a power set of $\mathcal{E}$). [1]

We now move on to obtain each value of $\arg\min_{E' \in \mathcal{E}_i} \sum_{E' \in \mathcal{E}_i} L(E, E')P(E|F)$ and consider a confusion network which enables us to combine several fragments from MT outputs. In the first step, we select the sentence-based best single system via a MBR decoder. Note that single system outputs are often used as the backbone of the confusion network. For example in Table 2, system t1 is selected as the backbone. Note that the backbone determines the general word order of the confusion network.

In the second step, based on the backbone which is selected in the first step, we build the confusion network by aligning the hypotheses with the backbone. In this process, we used the TER distance (Snover et al., 2006) between the backbone and the hypotheses. We do this for all the hypotheses sentence by sentence. Note that in this process, deleted words are substituted as NULL words (or ϵ-arcs). For example in Table 2, the lower half shows an example of a confusion network. hyp(t2), . . . , hyp(t5) are aligned according to the backbone(t1). Note that $*$ denotes ϵ-arcs, (D) denotes deletion, (I) denotes insertion, and (S) denotes substitution following the terminology in the TER distance literature. The right most column in Table 2 in the rows of the confusion network, that is 57.14, 71.43, and so forth, shows the TER score for this example.

In the third step, the consensus translation is extracted as the best path in the confusion network. The most primitive approach (Matusov et al., 2006) is to select the best word $\hat{e}_k$ by the word posterior probability via voting at each position k in the con-

[1] A power set of $\mathcal{E} = \{1, 2\}$ is $\{\{1, 2\}, \{1\}, \{2\}, \emptyset\}$.

segment 782		
Input t1	since the a team of almost 1000 policemen is in charge of security .	
Input t2	since the previous day an equipment of almost 1000 policewomen is being in charge of the safety .	
Input t3	from the previous day a team from almost 1000 police officer himself is using of the security	
Input t4	from the previous day a team of almost 1000 police is occupying of the security .	
Input t5	since the day before a team of almost 1 policemen is pursuing security .	
backbone(t1)	since the a team of almost 1000 policemen is in charge of security .	
hyp(t2)	since the previous(I) day(I) an(S) equipment(S) of almost 1000 policewomen(S) is being(I) in charge of the(I) safety(S) .	57.14
hyp(t3)	from(S) the(I) previous(I) day(S) a team from(S) almost 1000 police(I) officer(I) himself(S) is using(S) the(S) of security .	71.43
hyp(t4)	from(S) the previous(I) day(I) a team of almost 1000 police(S) is occupying(S) the(S) of security .	50.00
hyp(t5)	since the day(I) before(I) a team of almost 1(S) policemen is *(D) *(D) pursuing(S) security .	42.86
output	since the previous day a team of almost 1000 policemen is in charge of security .	

Table 2: Example from the 782th sentence from the testset. First we choose the first input as the backbone. Second, we make the confusion network measuring the performance by TER. Then, the consensus translation of ”since the previous day a team of almost 1000 policemen is in charge of security .” is obtained as an output.

fusion network, as in (3):

$$\hat{E}_k = \arg\max_{e \in \mathcal{E}} p_k(e|F) \qquad (3)$$

Note that this word posterior probability can be used as a measure how confident the model is about this particular word translation (Koehn, 2010), as defined in (4):

$$p_i(e|F) = \sum_j \delta(e, e_{j,i}) p(e_j|F) \qquad (4)$$

where $e_{j,i}$ denotes the i-th word and $\delta(e, e_{j,i})$ denotes the indicator function which is 1 if the i-th word is e, otherwise 0. However, in practice as is shown by (Du et al., 2009; Leusch et al., 2009), the incorporation of a language model in this voting process will improve the quality further. Hence, we use the following features in this voting process: word posterior probability, 4-gram and 5-gram target language model, word length penalty, and NULL word length penalty. Note that Minimum Error-Rate Training (MERT) is used to tune the weights of the confusion network. In Table 2, “since the previous day a team of almost 1000 policemen is in charge of security .” is selected in this voting process. In the final step, we remove the ϵ-arcs if existed.

3 Experiments

We use MERT (Och, 2003) internally to tune the weights and language modeling is provided by SRILM (Stolcke, 2002). We did not use any external language data resources.

Our results as obtained by the system described in Section 2 (which automatically selects and discards translations provided by the component MT systems) are shown in the results line in Table 3. Although the organizers provide the reference set for the testset, the decision that we make in the following is based on the results obtained on the development set performance since we cannot access the reference set in “real life” situations. Due to the performance on the development set, we tuned the parameters in our system as is described in Section 2.

The improvement in BLEU was 2.16 points absolute and 9.2% relative compared to the performance of system t2, the single best performing system (we optimized according to BLEU). Except for METEOR, we achieved the best performance in NIST (0.14 points absolute and 2.1% relative), WER (0.71 points absolute and 1.1% relative) and PER (0.64 points absolute and 1.3% relative) as well.

In order to shed further light on the intermediate results, we sampled three combinations of single best translation outputs, which are shown in Table 3 as well. Combination 1 includes all of the five single best translation outputs. Combination 2 includes t1, t2, t4, and t5 which eliminates system t2 which performed worst in terms of development set perfor-

	NIST	BLEU	METEOR	WER	PER
system t1	6.3934	0.1968/0.1289*	0.5022487	62.3685	47.3074
system t2	6.3818	0.2337/0.1498*	**0.5732194**	64.7816	49.2348
system t3	4.5648	0.1262/0.0837*	0.4073446	77.6184	63.0546
system t4	6.2136	0.2230/0.1343*	0.5544878	64.9050	50.2139
system t5	6.7082	0.2315/0.1453*	0.5412563	60.6646	45.1949
results	**6.8419**	**0.2553**	0.5683086	**59.9591**	**44.5357**
combination 1 (t1,t2,t3,t4,t5)	6.7151	0.2505	0.5701207	60.6993	45.5148
combination 2 (t1,t2,t4,t5)	6.8419	0.2553	0.5683086	59.9591	44.5357
combination 3 (t2,t4,t5)	6.7722	0.2498	0.5687383	60.6723	45.2257

Table 3: We do experiments and obtained the results as above (See the results line). All the scores are on testset except those marked * (which are on devset). On comparison, we did sampling of three combinations of the single systems, which shows that our results are equivalent to the combination 2. These exeprimental results validate our motivating results: it is often the case that some radically bad translation output may harm the final output by system combination. In this case, system t3 whose BLEU score is 12.62 has a negative effect on the results of system combination. The best performance was achieved by removing this system, i.e. the combination of systems t1, t2, t4, and t5.

mance. Combination 3 includes t2, t4, and t5 which eliminates the two worst systems in terms of the development set performance.

It is evident that our overall result is equivalent to Combination 2. Combination 2 achieved the best performance among these three combinations in NIST (0.13 points absolute and 2% relative), WER (0.70 points absolute and 1.1% relative) and PER (0.66 points absolute and 1.4% relative) as well. Combination 1 is second best in terms of BLEU scores. The improvement in BLEU was 1.68 points absolute and 7.1% relative. Combination 3 achieves 1.61 points improvement absolute and 6.9% relative.

4 Discussion

In Statistical Machine Learning (Vapnik, 1998), the term Bayes risk refers to the minimum risk over all possible measurable functions. This strategy leads to find the best hypothesis under the worst case analysis which is called agnostic learning (Kearns et al., 1994). In agnostic learning, with probability 1-δ, the number of training samples sufficient to ensure that every hypothesis H having zero training error will have a true error m of at most ϵ, is investigated as is shown in (5):

$$m \quad \geq \quad \frac{1}{\epsilon}(\ln|H| + \ln|\frac{1}{\delta}|) \tag{5}$$

In Support Vector Machines (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000), this strategy is called the empirical risk minimization or the structural risk minimization. For example, in the case of an (independent) regression problem,[2] Bayes risk is defined as in (6):

$$R(t) \quad = \quad \inf_g R(g) \tag{6}$$

where t is a target function and g is a true function. Bayes risk can be further rewritten as in (7):

$$R(g) \quad = \quad P(g(X) \neq Y) = \mathbb{E}(\mathbf{1}_{g(X)\neq Y}) \tag{7}$$

where **1** denotes an indicator function. As we cannot measure this risk since P is unknown, we use the following empirical risk (8) to measure the performance:

$$R_n(g) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{g(X_i)\neq Y_i} \tag{8}$$

This leads to the theory of worst case analysis taken by Support Vector Machines. To seek minimal risk is equivalent to seeking high probability mass in the hypothesis space since Eq (8) counts how many $g(X_i)$ and Y_i disagree with each other. We seek high counts of disagreement.

[2] Let us consider an input space $\mathcal{X}$ and output space $\mathcal{Y}$. We assume that a set of n IID pairs (X_i, Y_i) sampled according to an unknown but fixed distribution P. Suppose that our task is to predict a function $g : \mathcal{X} \rightarrow \mathcal{Y}$ where we call g a true function. Now, let t be a target function $t(x) = \text{sgn}\eta(x)$ where $\eta(x) = \mathbb{E}[Y|X = x] = 2\mathbb{P}[Y = 1|X = x] - 1$.

In the case of Machine Translation, this analogy can be extended. As is shown in Eq (1), MBR decoding seeks to obtain the translations whose probability mass are concentrated (Koehn, 2010) where each word is split as in Eq (4) if we take the confusion network-based approach of system combination. Hence, if the same words appear in the same word position, such words may occupy the high probability mass in Eq (4). If we include incorrect translation output among candidate translation outputs in the same word position, incorrect words may occupy the high probability mass. Then, the resulting output may include such bad words, causing the overall BLEU score to be low. Although this is not a conclusive explanation, this explains the possibility in a qualitative way why our combination 1 can be worse than our combination 2 in Table 3.

5 Conclusion and Further Studies

This paper describes the system combination module in the MT system MaTrEx developed at Dublin City University. We deployed the system combination module to this evaluation campaign. In this paper, we introduce a new input selection mechanism which removes some radically bad systems for the sake of achieving final better overall performance. Although this phenomenon was observed between JP-EN (Okita et al., 2010b), we implemented this mechanism in the procedure in this paper and showed the same to hold between ES-EN. Improvement was 2.16 BLEU points absolute and 9.2% relative compared to the best single system.

Further study will investigate the effect of bad translation inputs in system combination. Currently our implementation of Eq (2) is somewhat naive, in that the approach considers all subsets of translations contributed by the individual MT systems. We will work on a strategy how to select translation inputs optimally. In particular such a discussion will be fruitful if our inputs are the 1000-best list as in the case of Tromble et al. (Tromble et al., 2008) and DeNero et al. (DeNero et al., 2009). Their improvements are in general quite small compared to the confusion network-based approach. As is shown in Figure 1, the 100-best list and the 1000-best list produced by Moses (Koehn et al., 2007) tend not to be sufficiently different and do not produce meaningful translation alternatives. As a result, their BLEU score tends to be low compared to the (nearly best) single systems. This means that in our strategy those MT inputs may be better removed rather than employed as a useful source in system combination.

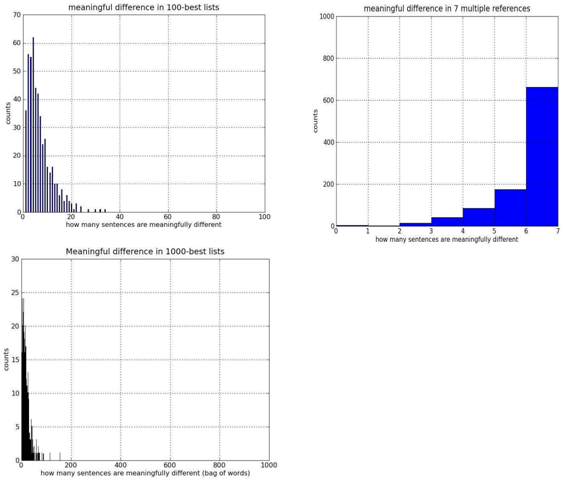

Figure 1: The upper left figure shows the count of exact matches among the translation outputs of Moses as a 100-best list after stop-word removal and sorting; We project each sentence in a 100-best list onto vector space model and count the number of points. The lower left figure shows the same quantity for a 1000-best list. The upper right figure shows the same quantity for a 7-multiple reference (human translation). We use the parallel data of IWSLT 07 JP-EN where we use devset5 (500 sentence pairs) as a development set and devset4 (489 sentence pairs) as a test set; 7-multiple references consist of devset4 and devset5 (989 sentence pairs). For example, the upper left figure shows that 7% of sentences produce only one meaningful sentence in a 100-best list and the other 99 sentences in a 100-best list is just a reordered version. In contrast, the upper right figure of human translation shows that more than 70% of sentences in 7 multiple references are meaningfully different.

Yet another avenue for further study is to provide prior knowledge into the system combination module. In word alignment, one successful strategy is to embed prior knowledge about alignment links (Okita et al., 2010a; Okita, 2011; Okita and Way, 2011), which work as the link between statistical learning and linguistic resources. We have shown that the selection of MT input sentences is an effective strategy in this paper. Similarly, it would be interesting to incorporate some prior knowledge about system combination, for example, (in)correct words or phrases in some particular translation output.

6 Acknowledgements

This research is supported by the the 7th Framework Programme and the ICT Policy Support Programme of the European Commission through the T4ME project (grant agreement no. 249119).

References

Srinivas Bangalore, G. Bordel, and G. Riccardi. 2001. Computing consensus translation from multiple machine translation systems. *In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 350–354.

Nello Cristianini and John Shawe-Taylor. 2000. Introduction to Support Vector Machines. *Cambridge University Press*.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. *In proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 567–575.

Jinhua Du, Yifan He, Sergio Penkale, and Andy Way. 2009. MaTrEx: the DCU MT System for WMT 2009. *In Proceedings of the Third EACL Workshop on Statistical Machine Translation*, pages 95–99.

Michael J. Kearns, Robert Schapire, and Linda Sellie. 1994. Towards efficient agnostic learning. *Machine Learning*, 17:115–141.

Philipp Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for Statistical Machine Translation. *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Philipp Koehn. 2010. Statistical machine translation. *Cambridge University Press*.

Sanjiv Kumar and William Byrne. 2002. Minimum Bayes-Risk word alignment of bilingual texts. *In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147.

Gregor Leusch, Eugene Matusov, and Hermann Ney. 2009. The rwth system combination system for wmt 2009. *In Fourth EACL Workshop on Statistical Machine Translation (WMT 2009)*, pages 56–60.

Eugene Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. *In Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 33–40.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167.

Tsuyoshi Okita and Andy Way. 2011. Given bilingual terminology in statistical machine translation: Mwe-sensitve word alignment and hierarchical pitman-yor process-based translation model smoothing. *In Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference (FLAIRS-24)*.

Tsuyoshi Okita, Alfredo Maldonado Guerra, Yvette Graham, and Andy Way. 2010a. Multi-Word Expression sensitive word alignment. *In Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA2010, collocated with COLING2010), Beijing, China.*, pages 1–8.

Tsuyoshi Okita, Jie Jiang, Rejwanul Haque, Hala Al-Maghout, Jinhua Du, Sudip Kumar Naskar, and Andy Way. 2010b. MaTrEx: the DCU MT System for NTCIR-8. *In Proceedings of the MII Test Collection for IR Systems-8 Meeting (NTCIR-8), Tokyo.*, pages 377–383.

Tsuyoshi Okita. 2011. Word alignment and smoothing method in statistical machine translation: Noise, prior knowledge and overfitting. *PhD thesis. Dublin City University*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method For Automatic Evaluation of Machine Translation. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 311–318.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Andreas Stolcke. 2002. SRILM – An extensible language modeling toolkit. *In Proceedings of the International Conference on Spoken Language Processing*, pages 901–904.

Roy Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice minimum bayes-risk decoding for statistical machine translation. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 620–629.

Vladimir Vapnik. 1998. Statistical learning theory. *Wiley and Sons*.