Machine Translation System Combination with MANY for ML4HMT

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

Abstract

This paper describes the development of a baseline machine translation system combination framework with the MANY tool for the 2011 ML4HMT shared task. Hypotheses from French–English rule-based, example-based and statistical Machine Translation (MT) systems were combined with MANY, an open source system combination software based on confusion networks decoding currently developed at LIUM. In this baseline framework, the extra information about the MT systems provided for the shared task was not used. The system combination yielded significant improvements in BLEU score when applied on system combination data.

1 Introduction

The "Machine Learning for Hybrid Machine Translation" (ML4HMT) workshop proposed a shared task which objective was to investigate whether system combination or hybrid machine translation techniques could benefit from extra information (linguistically motivated, decoding and runtime) from the different systems involved. Thus the focus was to improve the combination of several types of MT systems (rule-based, example-based and statistical) thanks to the extra information corresponding to each type of system.

The LIUM computer science laboratory participated in this shared task providing a baseline for it, that is a system combination withouth using any of the extra information provided by the organisers about each MT system. The one-best system outputs were combined using the MANY[1] (Barrault, 2010) framework, an open source system combination software based on Confusion Networks (CN).

The MANY toolkit was run with all default options. These options, and more generally the various steps involved in the combination system, are described in Section 2. The data available for the shared task and the results obtained are presented in Section 3.

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; Rosti et al., 2009). MANY can be decomposed in two main modules: an alignment module and a decoder (see Figure 1), which are described in the next sections. A last section deals with parameter tuning.

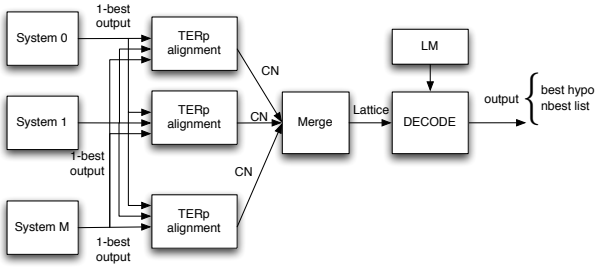

Figure 1: System combination based on confusion network decoding.

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

Alignment Module

The alignment module is actually a version of TERp (Snover et al., 2009) which has been modified to add some functionalities, such as alignment between a sentence and a confusion network. The alignment with TERp uses different costs (which corresponds to an exact match, an insertion, a deletion, a substitution, a shift, a synonym match and a stem match) to compute the best alignment between two sentences. In the case of confusion networks, 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.

The role of the alignment module is to incrementally align the hypotheses against a backbone in order to create a confusion network, as depicted in Figure 2. Each hypothesis acts as backbone, the remaining hypotheses being aligned and merged to it beginning with the nearest in terms of TER and ending with the more distant one. If there are $M + 1$ hypotheses to combine, $M + 1$ confusion networks are generated. Those confusion networks are then connected together into a single lattice by adding a first and last node. The probability of the first arcs (later named priors) must reflect how well such system provides a well structured hypothesis.

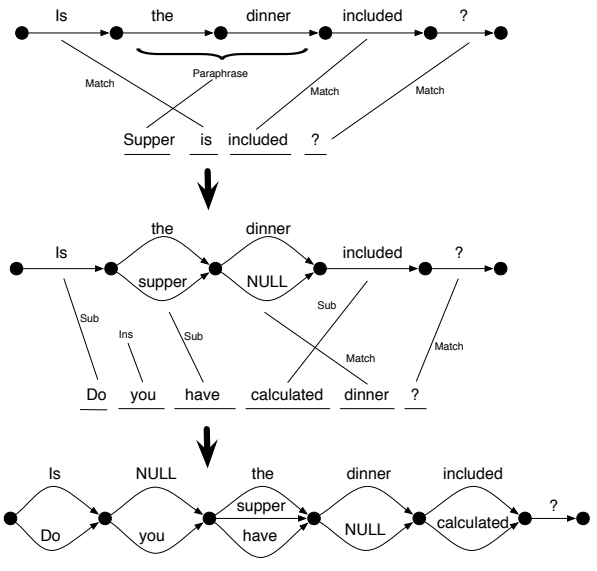

Figure 2: Incremental alignment with TERp resulting in a confusion network.

Decoder

The 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 :

$$log(P_W) \quad = \quad \sum_i \alpha_i \; log\Big(h_i(t)\Big) \qquad (1)$$

where t is the hypothesis, the α_i are the weights of the feature functions h_i. The following features are considered for decoding:

- The language model probability: the probability given by a 4-gram language model.
- The word penalty: penalty depending on the size (in words) of the hypothesis.
- The null-arc penalty: penalty depending on the number of null-arcs crossed in the lattice to obtain the hypothesis.
- System weights: each word receive a weight corresponding to the sum of the weights of all systems which proposed it.

At the beginning, only one token is created at the first node of the lattice. Then this token spreads over the consecutive nodes, accumulating the score on the arc it crosses, the language model probability of the word sequence generated so far and null or length penalty if applicable. The number of tokens can increase really quickly to cover the whole lattice, and, in order to keep it tractable, only the $Nmax$ best tokens are kept (the others are discarded), where $Nmax$ can be set at the start. Other methods to restrict the number of tokens (like pruning based on score or other heuristics) can easily be implemented in this software, but this has not been implemented yet.

Tuning

According to recent experiments (Barrault, 2011), it is better to consider the tuning of the alignment module parameters and the decoder parameters in two distinct steps.

By default, TERp costs are set to 0.0 for match and 1.0 for everything else. These costs are not optimal, since a shift in that case will hardly be possible. However, tuning these costs (with Condor, a numerical optimizer based on Powell's algorithm, (Berghen and Bersini, 2005)) never showed significant improvements so far. Thus the default configuration in the current version of MANY is to keep default TERp weights for alignment.

Decoder feature functions weights were optimized with MERT (Och, 2003). The 300-best list created at each MERT iteration was appended to the n-best lists created at previous iterations. This proved to be a more reliable tuning than previous tuning of decoder weights performed with Condor (Barrault, 2011).

3 Shared Task

The task consisted in combining the outputs of the following five MT systems: Joshua (hierarchical), Lucy (rule-based), Metis (working with a monolingual target corpus and a bilingual dictionary only), Apertium (rule-based) and Matrex (combination of example-based and phrase-based SMT features). Outputs of these MT systems were provided on a development set to tune the combination framework, and on a test set to produce the combination output to be evaluated. We took as input of our combination system the one-best plain text output extracted from the xml file for each MT system. The original case was preserved (lower case for the Joshua output and true case for the rest of systems) and the texts were tokenized. Statistics of the development (dev) and test sets calculated on the reference after tokenization are presented in Table 1.

NAME	#sent.	#words
dev	1025	23908
test	1026	25863

Table 1: ML4HMT shared task corpora : number of sentences and running words (after tokenization) calculated on the reference.

Language model. The English target language model has been trained on the only data set allowed for the shared task, namely the News Commentary corpus provided for the MT shared task of the Sixth Workshop of Statistical Machine Translation (WMT 2011).[2] This corpus contains 180k running words of quality commentary articles about the news. We used the SRILM toolkit (Stolcke, 2002) to train a 4-gram back-off language model with Kneser-Ney (Kneser and Ney, 1995) smoothing.

LM weight	Word penalty	Null penalty
0.032	0.23	0.010

Joshua	Lucy	Metis	Apertium	Matrex
-0.013	-0.27	+0.014	-0.21	-0.22

Table 2: Parameters obtained with tuning decoder parameters with MERT.

System	BLEU	TER	METEOR
Joshua	13.80	67.30	52.71
Lucy	22.70	61.97	57.62
Metis	9.09	80.02	41.36
Apertium	21.61	62.88	55.25
Matrex	20.18	60.18	56.55
MANY	24.36	58.55	56.25

Table 3: Automatic scores on the test set for the single MT hypotheses and their combination with MANY.

Tuning. The alignment module was run on the dev set MT hypotheses without tuning, keeping the default TERp weights (0 for exact match and 1 for the other costs). Decoding of the resulting lattice of confusion networks was tuned using MERT to obtain the set of decoder feature functions weights which provides the best scoring combination output on the dev set. The optimum set of parameters obtained is presented in Table 2. The system thus gave a higher weight to words coming from the hypothese proposed by Lucy, then by Matrex, Apertium, Joshua, and it weighted negatively words proposed by Metis.

Evaluation. The test set hypotheses were incrementally aligned with TERp default costs, a lattice was created with the resulting confusion networks, and decoding was conducted with the weights presented in Table 2. This produced the final combination output, which was evaluated on the test set against the reference, as well as the MT hypotheses.

[2] http://www.statmt.org/wmt11/

The evaluation results are shown in Table3. The combination with MANY improves the best single system BLEU score (Lucy) by 1.6 points, the best single system TER score (Matrex) by 1.6 points, but its METEOR score is 1.3 points below the one of the best single system (Matrex).

Another remark about the results is that the ranking of the systems resulting from the weights obtained during tuning (Table 2), namely Lucy/Matrex/Apertium/Joshua/Metis, is consistent with the METEOR score ranking, and close to the BLEU or TER rankings.

4 Conclusions and perspectives

We ran the MANY system combination toolkit on five MT systems of different types provided for the ML4HMT workshop shared task. The combination achieved a better BLEU score and TER score than the best single system (with a 1.6 point gain in both cases), but a worse METEOR score. We emphasize that in the current version, although MANY can benefit from various information sources, the decision taken by the decoder mainly depends on a target language model. Thus the decision to restrict the size of the authorized monolingual training corpus was a severe limitation. In the future, we want to estimate good confidence measure to use in place of the systems priors. These confidences measures have to be related to the system performances, but also to the complementarity of the systems considered.

Finally, we want to give some ideas of how extra information about the MT systems could be taken into account within MANY. The decoder could benefit from information related to the hypothesis, such as the phrase pairs used and their probabilities, or the language model probabilities of each n-gram. The search space could be extended with synonyms, paraphrases or other types of information.

5 Acknowledgements

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

Loïc Barrault. 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.

Loïc Barrault. 2011. Many improvements for WMT. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 135–139, Edinburgh, Scotland.

Frank Vanden Berghen and Hugues Bersini. 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, September.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 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, June 16-17.

Kneser and Ney. 1995. Improved backing-off for m-gram language modeling. In *IEEE Inte. Conf. on Acoustics, Speech and Signal Processing*, pages 49–52, Detroit, MI, May.

F. Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*, Sapporo, Japan.

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

A.-V.I. Rosti, B. Zhang, S. Matsoukas, , and R. Schwartz. 2009. Incremental hypothesis alignment with flexible matching for building confusion networks: BBN system description for WMT09 system combination task. In *EACL/WMT*, pages 61–65.

Wade Shen, Brian Delaney, Tim Anderson, and Ray Slyh. 2008. The MIT-LL/AFRL IWSLT-2008 MT System. In *International Workshop on Spoken Language Translation*, Hawaii, U.S.A, 69–76.

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

A. Stolcke. 2002. SRILM: an extensible language modeling toolkit. In *Proc. of the Int. Conf. on Spoken Language Processing*, pages 901–904, Denver, CO.