Using Hypothesis Selection Based Features for Confusion Network MT System Combination

Sahar Ghannay
LIUM, University of Le Mans
Le Mans, France
Sahar.Gannay.Etu@univ-lemans.fr

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

Abstract

This paper describes the development operated into MANY, an open source system combination software based on confusion networks developed at LIUM. The hypotheses from Chinese-English MT systems were combined with a new version of the software. MANY has been updated in order to use word confidence score and to boost n-grams occurring in input hypotheses. In this paper we propose either to use an adapted language model or adding some additional features in the decoder to boost certain n-grams probabilities. Experimental results show that the updates yielded significant improvements in terms of BLEU score.

1 Introduction

MANY (Barrault, 2010) is an open source system combination software based on Confusion Networks (CN). The combination by confusion networks generates an exponential number of hypotheses. Most of these hypotheses contain n-grams do not exist in input hypotheses. Some of these new n-grams are ungrammatical, despite the presence of a language model. These novel n-grams are due to errors in hypothesis alignment and the confusion network structure. In section 3 we present two methods used to boost n-grams present in input hypotheses.

Currently, decisions taken by the decoder mainly depend on the language model score, which is deemed insufficient to precisely evaluate the hypotheses. In consequence, it is interesting to estimate a score for better judging their quality. The challenge of our work is to exploit certain parameters defined by (Almut Siljaand and Vogel, 2008) to calculate word confidence score. These features are detailed in section 4. The approach is evaluated on the internal data of the BOLT project. Some experiments have been performed on the Chinese-English system combination task. The experimental results are presented in section 5. Before that, a quick description of MANY, including recent developments can be found in section 2.

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, see *e.g.* (Antti-Veikko I.Rosti and Schwartz, 2007; Damianos Karakos and Dreyer, 2008; Shen et al., 2008; Antti-Veikko I. Rosti and Schw, 2009). MANY can be decomposed in two main modules. The first one is the alignment module which is a modified version of TERp (Matthew G. Snover and Schwartz, 2009). Its role is to incrementally align the hypotheses against a backbone in order to create a confusion network. 1-best hypotheses from all M systems are aligned in order to build M confusion networks (one for each system considered as backbone). These 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, synonym, and stem) costs are considered when the word in the hypothesis matches (is a substitution, a synonym or a stem of) at least one word of the considered confusion sets in the CN. 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 :

Proceedings of the 3rd Workshop on Hybrid Approaches to Translation (HyTra) @ EACL 2014, pages 2–6,
Gothenburg, Sweden, April 27, 2014. ©2014 Association for Computational Linguistics

$$\log(P_w) = \sum_i \alpha_i \log(h_i(t)) \qquad (1)$$

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 system receives a weight according to its importance. Each word receives a weight corresponding to the sum of the weights of all systems which proposed it.

Our goal is to include the following ones:

- Word confidence score: each word is given a score, which is the combination of the three scores described in section 4 (equation 7).
- n-gram count: number of n-grams present in input hypotheses for each combined hypothesis.

In most cases, the new features have best weights according to MERT (*e.g.* the best decoding weights of these features by combining two systems are: lm-weight: 0.049703, word-penalty: 0.0605602, null-penalty: 0.319905, **weight-word-score: -0.378226**, **weight-ngram-count: -0.11687**, priors: 0.0141794#-0.0605561).

3 boost n-grams

We defined two methods to boost n-grams present in input hypotheses. The first one is adding the count of bi or tri-grams like a new feature to the decoder as mentioned in Section 2. The second method is using an adapted language model (LM) to decode the lattice, in order to modify n-grams probabilities, that have been observed in input hypotheses.

Language models

Three 4-gram language models named *LM-Web*, *LM-Tune* and *LM-Test*, are used to interpolate the adapted LM. They were trained respectively on the English web Corpus and the system outputs : development and test sets (except their references) involved in system combination, using the SRILM Toolkit (Stolcke, 2002). The resulting model from the interpolation of *LM-Tune* and *LM-Test* is interpolated linearly with the *LM-Web* to build the adapted LM. These models are tuned to minimize the perplexity on the tune reference.

4 Word confidence score

The best hypothesis selection relies on several features. In (Barrault, 2011) decisions taken by the decoder depend mainly on a n-gram language model, but it is sometimes insufficient to evaluate correctly the quality of the hypotheses. In order to improve these decisions, some additional information should be used. Several researches presented some studies of confidence scores at word and sentence level, such as (Almut Siljaand and Vogel, 2008) and (Ueffing and Ney, 2007). A large set of confidence scores were calculated over the n-best list. (Almut Siljaand and Vogel, 2008) defines several features extracted from n-best lists (at the sentence level) to select the best hypothesis in a combination approach via hypothesis selection. The challenge of our work is to exploit these features to estimate a confidence score at the word level and injecting it into the confusion networks. The following features are considered:

Word agreement score based on a window of size *t* around position *i*

This score represents the relative frequency of hypotheses in the n-best lists containing the word ***e*** in a window of size t around the position i. It is computed as follows:

$$\text{WA}_k(e_{i,t}) = \frac{1}{N_k} \sum_{p=0}^{N_k} f(e_{p,i-t}^{p,i+t}, e) \qquad (2)$$

where N_K is the number of hypotheses in the n-best list for the corresponding source sentence k, t={0, 1 or 2} and $f(S_i^j, w)$ =1 if w appears in the word sequence S_i^j.

When t equals 0, this means that $i = t$, then this score only depends on words at the exact position i. The agreement score is calculated accordingly:

$$\mathrm{WA}_k(e_i) = \frac{1}{N_k} \sum_{p=0}^{N_k} f(e_{p,i}, e) \quad (3)$$

The two equations described above, are handled in our contribution, thus the final word agreement score is the average between them if $\mathrm{WA}_k(e_i) \neq 0$ otherwise it is equal to $\mathrm{WA}_k(e_{i,t})$ score.

Position independent n-best List n-gram Agreement

This score represents the percentage of hypotheses in the n-best lists that contain the n-gram $e^i_{i-(n-1)}$, independently of its position in the sentence, as shown in Equation 4. For each hypothesis the n-gram is counted only once.

$$\mathrm{NA}_k(e^i_{i-(n-1)}) = \frac{1}{N_k} \sum_{p=0}^{N_k} f(e^i_{i-(n-1)}, e^I_{1,p}) \quad (4)$$

where $f(e^i_{i-(n-1)}, e^I_{1,p}) = 1$ if the n-gram $e^i_{i-(n-1)}$ exists in the p^{th} hypothesis of the n-best list. We use n-gram lengths of 2 and 3 as two separate features.

The position independent n-best list word agreement is the average count of n-grams that contain the word e. It is computed as:

$$\mathrm{NA}_k(e_i) = \frac{1}{N_{ng}} \sum_{n=0}^{N_{ng}} \mathrm{NA}_k(e^i_{i-(n-1)}) \quad (5)$$

Were N_{ng} is the number of n-grams of hypothesis k.

N-best list n-gram probability

This score is a traditional n-gram language model probability. The n-gram probability for a target word e_i given its history $e^{i-1}_{i-(n-1)}$ is defined as:

$$\mathrm{NP}_k(e_i | e^{i-1}_{i-(n-1)}) = \frac{C(e^i_{i-(n-1)})}{C(e^{i-1}_{i-(n-1)})} \quad (6)$$

Where $C(e^i_{i-(n-1)})$ is the count of the n-gram $e^i_{i-(n-1)}$ in the n-best list for the hypothesis k.

The n-best list word probability $\mathrm{NP}_k(e_i)$ is the average of the n-grams probabilities that contain the word e.

The word confidence score is computed using these three features as follows:

$$S_k(e_i) = \frac{\mathrm{WA}_k(e_i) + \sum_{j \in NG} \mathrm{NA}_k(e_i)^j + \mathrm{NP}_k(e_i)^j}{1 + 2 * |NG|} \quad (7)$$

where NG is the set of n-gram order, experimentally defined as NG=$\{$2-gram, 3-gram$\}$ and t = 2. Each n-gram order in the set NG is considered as a separate feature.

5 Experiments

During experiments, data from the BOLT project on the Chinese to English translation task are used. The outputs (200-best lists) of eight translation systems were provided by the partners. The best six systems were used for combination. *Syscomtune* is used as development set and *Dev* as internal test, these corpora are described in Table 1:

NAME	#sent.	#words.
Syscomtune	985	28671
Dev	1124	26350

Table 1: BOLT corpora : number of sentences and words calculated on the reference.

To explore the impact of each new feature on the results, they are tested one by one (added one by one in the decoder) then both, given that, the oldest ones are used in all cases. These tests are named respectively **boost-ngram**, **CS-ngram** and **Boost-ngram+CS-ngram** later.

The language model is used to guide the decoding in order to improve translation quality, therefore we evaluated the baseline combination system and each test (described above) with two LMs named *LM-Web* and *LM-ad* and compared their performance in terms of BLEU. By comparing their perplexities, that are respectively *295.43* and *169.923*, we observe a relative reduction of about *42.5%*, that results in an improvement of BLEU score.

Figure 1 shows the results of combining the best systems (up to 6) using these models, that achieved respectively an improvement of *0.85* and *1.17* %BLEU point relatively to the best single system. In the remaining experiments we assume that *MANY-LM-Web* is the baseline.

Figure 2 shows interesting differences in how approaches to boost n-gram estimates behave when the number of input systems is varied. This is due to the fact that results are conditioned by the number and quality of n-grams added to the lattice

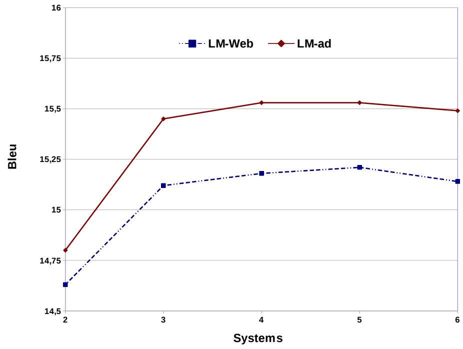

Figure 1: Performance (%BLEU-cased) of MANY after reassessment by LM-Web and LM-ad on the test set.

when the number of systems is varied, that provides varied outputs. In consequence, we observe that using the adapted LM is better than n-gram count feature to boost n-grams, indeed it guarantees n-grams quality.

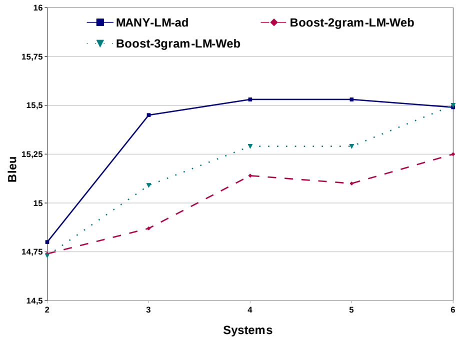

Figure 2: Comparison of n-gram boost approaches.

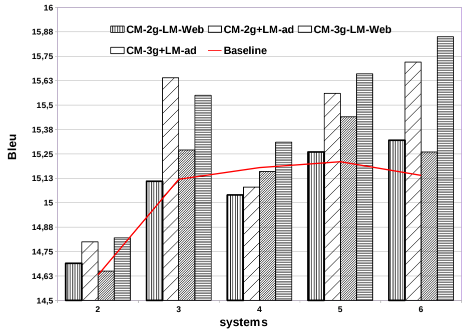

Figure 3: The impact of confidence score on the results when using LM-Web and LM-ad for decoding.

The 200-best lists are operated to estimate the word confidence score that contributes the most to the improvement of results when several (up to 6) systems are combined, as described in Figure 3, whatever the language model used, compared to the baseline. In addition, it seems that the confidence score performs better with the adapted LM than *LM-Web*.

Systems	BLEU
Best single	**14.36**
Sys2	14.21
Sys3	13.76
Sys4	13.52
Sys5	13.36
Sys6	12.99
MANY+LM-Web(baseline)	**15.14**
Boost-2gram+LM-Web	15.25
Boost-3gram+LM-Web	15.50
CS-2gram+LM-Web	15.32
CS-3gram+LM-Web	15.26
Boost-2gram+CS-2gram+LM-Web	15.39
Boost-3gram+CS-3gram+LM-Web	**15.78**
MANY+LM-ad	**15.49**
Boost-2gram+LM-ad	15.24
Boost-3gram+LM-ad	15.32
CS-2gram+LM-ad	15.72
CS-3gram+LM-ad	**15.85**
Boost-2gram+CS-2gram+LM-ad	15.61
Boost-3gram+CS-3gram+LM-ad	15.74

Table 2: Impact of new features and the adapted LM on the combination result of six systems.

Table 2 summarizes the best experiments results by combining the best six systems on the test set. We observe that new features yield significant improvements in term of BLEU score whatever the language model used for decoding. But it is clear that the adapted LM performs relatively well in comparison with *LM-Web*, so the best gains achieved over the best single system and the baseline are respectively *1.49* and *0.71* for *CS-3-gram+LM-ad*.

6 Conclusion

Several technical improvements have been performed into the MT system combination MANY, that are evaluated with the BOLT project data. An adapted LM and new features gave significant gains. Previous experimental results show that using the *adapted* LM in rescoring together with word confidence score and the oldest features improves results in term of BLEU score. This even results in better translations than using a *classical* LM (*LM-Web*) trained on a monolingual training corpus.

References

Hildebrand Almut Siljaand and Stephan Vogel. 2008. Combination of Machine Translation Systems via Hypothesis Selection from Combined N-Best Lists. *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, pages 254–261.

Spyros Matsoukas Antti-Veikko I. Rosti, Bing Zhang and Richard Schw. 2009. Incremental Hypothesis Alignment with Flexible Matching for Building Confusion Networks: BBN System Description for WMT09 System Combination Task. *StatMT '09 Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 61–65.

Spyros Matsoukas Antti-Veikko I.Rosti and Richard Schwartz. 2007. Improved Word-Level System Combination for Machine Translation. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319.

Loïc Barrault. 2010. MANY Open Source Machine Translation System Combination. *The Prague Bulletin of Mathematical Linguistics*, pages 147–155.

Loïc Barrault. 2011. MANY improvements for WMT'11. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 135– 139.

Sanjeev Khudanpur Damianos Karakos, Jason Eisner and Markus Dreyer. 2008. Machine Translation System Combination using ITG-based Alignments. *In 46th Annual Meeting of the Association for Computational Linguistics*, pages 81–84.

Bonnie Dorr Matthew G. Snover, Nitin Madnani and Richard Schwartz. 2009. TER-Plus: Paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation journal*, pages 117–127.

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

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. *In Proceedings International Conference for Spoken Language Processing, Denver, Colorado*.

Nicola Ueffing and Hermann Ney. 2007. Word-Level Confidence Estimation for Machine Translation. *Computational Linguistics journal*, pages 9–40.