Applying boosting to statistical machine translation *

Antonio L. Lagarda and Francisco Casacuberta

Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
Camino de Vera, s/n, 46022 Valencia, Spain
{alagarda,fcn}@iti.upv.es

Abstract. Boosting is a general method for improving the accuracy of a given learning algorithm under certain restrictions. In this work, AdaBoost, one of the most popular boosting algorithms, is adapted and applied to statistical machine translation. The appropriateness of this technique in this scenario is evaluated on a real translation task. Results from preliminary experiments confirm that statistical machine translation can take advantage from this technique, improving the translation quality.

1 Introduction

State-of-the-art statistical machine translation (SMT) techniques are still far from producing high quality translations. This drawback leads us to introduce an alternative approach to the translation problem. In our work, we will propose an adaptation of boosting [1] to SMT.

The purpose of boosting methods is to find a highly accurate rule by combining many weak or base hypotheses. The boosting algorithm generates each one of these hypotheses by iteratively calling a weak learning algorithm. Each iteration takes into account the performance of the previous iterations, trying to concentrate on the instances that have not been correctly learned. All these weak hypotheses are then combined into a final hypothesis.

AdaBoost (Adaptive Boosting) employs a set of importance weights over the training examples [2, 3]. These weights are used by the learning algorithm to produce a new weak hypothesis with lower error with respect to them. In this way, these weights help the algorithm to concentrate on the examples which are hardest to classify.

In machine translation, learning techniques could be considered as weak, due to the low quality of their results. Boosting has previously been applied to machine translation in works like [4] or [5]. In this paper, we propose an adaptation of AdaBoost to a SMT task. Each round, a new translation model

* Work supported by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01, by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), by the i3media Cenit project (CDTI 2007-1012), and by VIDI-UPV under project 20070315.

will be learned, which will produce a new hypothesis. Finally, all these hypotheses will be combined to obtain the final translation.

The next sections introduce machine translation and the AdaBoost algorithm. After that, we discuss our adaptation proposal of AdaBoost to SMT. Experimental results are presented in section 5. Finally, some conclusions and future work are given in section 6.

2 Machine translation

Traditionally, the goal of SMT has been statistically stated as follows [6]. Given a source sentence $f_1^J \equiv f_1...f_j...f_J$, we have to find a target sentence $e_1^I \equiv e_1...e_i...e_I$ that maximizes:

$$\hat{e}_1^{\hat{I}} = \arg\max_{e_1^I} Pr(e_1^I|f_1^J) \tag{1}$$

Using Bayes' Theorem, and taking into account that $Pr(f_1^J)$ does not depend on e_1^I, we arrive at

$$\hat{e}_1^{\hat{I}} = \arg\max_{e_1^I} \{Pr(e_1^I) \cdot Pr(f_1^J|e_1^I)\} \tag{2}$$

Intuitively, this decomposition can be interpreted as follows. The language model probability $Pr(e_1^I)$ ensures that the output e_1^I is a well-formed sentence from the target language. On the other hand, the translation model probability $Pr(f_1^J|e_1^I)$ represents the relationship between the source sentence and its translation, being higher when the former is a good translation of the latter.

Phrase-based models. Phrase-based models [7–11] are translation models that approach probabilistic relationship between a sequence of contiguous words in the source sentence and another sequence of contiguous words in the target sentence. These models are very interesting since they can represent some limited contextual translation information.

All the decisions made are summarized in the hidden variable $\tilde{\mathbf{a}} = \tilde{a}_1^K$ (bilingual segmentations):

$$Pr(f_1^J|e_1^I) = \sum_{\tilde{a}} Pr(\tilde{a}, \tilde{f}_1^K|\tilde{e}_1^K) = \sum_{\tilde{a}} Pr(\tilde{a}|\tilde{e}_1^K) Pr(\tilde{f}_1^K|\tilde{a}, \tilde{e}_1^K) \tag{3}$$

Log-linear models. In practice all of these models (and possibly others) are often combined into a *log-linear model* for $Pr(e_1^I|f_1^J)$ [12]:

$$Pr(e_1^I|f_1^J) = \frac{exp\left(\sum_{m=1}^M \lambda_m b_m(f_1^J, e_1^I)\right)}{\sum_{e'^{I'}_1} exp\left(\sum_{m=1}^M \lambda_m b_m(f_1^J, e'^{I'}_1)\right)} \tag{4}$$

As the denominator does not depend on f_1^J, it can be omitted in the search process:

$$\hat{e}_1^{\hat{I}} = \arg\max_{e_1^I} \sum_{m=1}^{M} \lambda_m b_m(f_1^J, e_1^I) \quad (5)$$

where $b_m(f_1^J, e_1^I)$ can be any model that represents an important feature for the translation, such as $logPr(f_1^J|e_1^I)$, $logPr(e_1^I|f_1^J)$, $logPr(e_1^I)$ or any other. λ_m are the weights of the log-linear combination.

Training and search. To sum up, once the translation models have been chosen, their parameters are estimated in the training phase. After that, for each source sentence, search for the best hypothesis is carried out by the maximization of Equation 2 or, alternatively, of Equation 5 if using a log-linear model combination.

3 Boosting and AdaBoost

Boosting [1] is a bootstrap [13] ensemble method where each model's training set is chosen depending on the performance of the previous ones. In this way, boosting sequentally produces a series of models where each new model tries to focus on the examples that have been so far mislearned. Each resampling of the training set gives more importance to the incorrectly learned examples in the earlier stages.

The AdaBoost algorithm, introduced in 1995 by Freund and Schapire [2], proposes a practical implementation of the boosting technique. Pseudocode of AdaBoost applied to the binary classification task can be found in Figure 1.

4 Adapting AdaBoost to machine translation

In this section, we will discuss a possible adaptation of AdaBoost to SMT.

Training and reweighting. In SMT, particularly when dealing with large corpora, training is a highly expensive process in terms of computing time. A complete training process in each AdaBoost iteration would be prohibitive. Thus, we propose a different approach where a retraining of all the models is not necessary. Instead, we will add another model (b_t) to the log-linear combination (Equation 5). This new model will be the only one that will change as AdaBoost iterates.

There is a main difference between our proposal and the original AdaBoost. While the latter reweights the mislearned training examples, in our case we will reweight phrases (in the sense of sequences of contiguous words, see section 2), instead of the whole training sentences. Once we have translated one of the training examples in step 2, we know which bilingual phrases have been used to generate the hypothesis, since it is a subproduct of the translation process. We can contrast that information with the translation reference of the sentence, so that we can easily find out which phrases have been correctly chosen, and which

Algorithm AdaBoost

Given: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ where $x_i \in X$, $y_i \in \{-1, +1\}$
Initialize: $D_1(\mathbf{z}_i) = 1/m$ for all $i = 1 \ldots m$
For $t = 1, \ldots, T$,

1. Train the base learner using distribution D_t.
2. Obtain hypothesis $h_t : X \rightarrow \{-1, +1\}$
3. Calculate the training error ϵ_t of h_t:

$$\epsilon_t = Pr_{i \sim D_t}[h_t(\mathbf{x}_i) \neq y_i] \ , \tag{6}$$

4. Set

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) \ . \tag{7}$$

5. Update weights:

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \ , \tag{8}$$

where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution.

Output: Final hypothesis

$$H(x) = sign \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right) \tag{9}$$

Fig. 1. The AdaBoost algorithm for the binary classification task [3]

ones not. Thus, we can reweight those phrases in the b_t model so that, hopefully, the next iteration decoding step will not choose the wrong phrases. Similarly, we can also improve the weights in b_t of those correctly chosen phrases.

Obtaining the hypothesis. In addition to the training step, another computationally costly step of the original AdaBoost algorithm is the obtaining of the hypothesis. According to the AdaBoost algorithm, the whole training set should be translated. Nevertheless, in SMT, the large size of the corpora and the complexity of the decoding procedure substantially increase the cost of this step. In our proposal, instead of translating the whole training set, in each iteration we will randomly choose a subset with an affordable size.

Error obtaining. As we have said before, our AdaBoost proposal will work at a phrase level, not at a sentence level as the original AdaBoost does. Thus, error must be calculated over the phrases that have been chosen in the translation step.

Final hypothesis combination. The final hypothesis of the AdaBoost algorithm is a combination of each iteration hypothesis, weighted by α_t, as shown in equation 9 in Figure 1. Instead of just voting, we will combine them by taking the most *centered* hypothesis, i.e. obtaining for each h_t its error with respect to the others. The final hypothesis will be that h_t with the lowest average error with respect to the rest of hypotheses.

5 Experiments

In this section, we will show some translation experiments carried out by applying our AdaBoost proposal.

Experimental framework. Training and translation steps from AdaBoost algorithm were performed using the Moses toolkit [14]. This toolkit estimates four different translation models, which are combined in a log-linear model. Weights were adjusted by means of the MERT [15] procedure over a *development* subset. Translation was carried out with monotone reordering.

Corpus features. We employed two different corpora in our experiments: the Xerox corpus [16], and the Europarl corpus [17].

The Xerox corpus involves the translation of technical Xerox manuals from English (En) to Spanish (Es), French (Fr) and German (De) and vice-versa. In our experiments, we have chosen the Spanish and English sets in their simplified (tokenized, lowercased and categorized) version.

We also used a second larger parallel corpus, the French to English Europarl corpus. This corpus is a collection of transcripts of the European parliamentary proceedings. For our experiments, we chose the second version of this corpus, which was used in the 2006 Workshop on Machine Translation of the NAACL [18]. This corpus is divided into four separate sets: one for training, one for development, one for test (called *DevTest*) and another test set which was the one used in the workshop for the final evaluation. In our case, we present our translation results with the *DevTest* set.

Some statistics of these corpora are shown in Table 1. Perplexity is a measure from information theory that is useful to evaluate the complexity of a corpus [19].

		Xerox		Europarl	
		English	Spanish	English	French
Training	Sentences	56K		688K	
	Running words	665K	753K	15.6M	13.8M
	Vocabulary	8K	11K	80K	62K
	5-gram Perplexity	14.4	13.6	42.5	31.7
Dev	Sentences	1K		2K	
	Running words	14K	16K	67K	59K
	5-gram Perplexity	28.7	24.3	72.4	49.6
DevTest	Sentences	1K		2K	
	Running words	8K	10K	66K	58K
	5-gram Perplexity	51.1	35.3	71.6	49.6

Table 1. Features of Xerox and Europarl corpora (*K* denotes *thousand* and *M million*)

Evaluation metrics. The assessment of the translation quality has been carried out using the *BiLingual Evaluation Understudy* (BLEU) [20]. BLEU is a function (the weighted geometric mean) of the k-substrings ($k \leq 4$) that co-occur in both the hypothesized target sentence and in the reference target sentence, with a penaly for too short sentences. With this measure, higher figures imply better translation quality.

Significance tests. Finally, significance of our results has been assessed by the *paired boostrap resampling* method, described in [21, 22]. In this way, we compared our results with a baseline system, estimating whether our system improvement was statistically significant.

Results. Figure 2 shows the performance of our AdaBoost adaptation in a translation task from English to Spanish and vice versa with the Xerox corpus, and from French to English in the case of Europarl. Baseline error rate in terms of BLEU is shown in both cases with a horizontal line. For each iteration, the figure plots the quality of the final hypothesis, which is a combination of all the previous h_t. All improvements with respect to the baseline system are significant according to the *paired boostrap resampling* method.

Apart from the automatic quality assessment, the final hypotheses can be manually evaluated to analyse in which way AdaBoost improves the translation quality. In general, AdaBoost amends those phrases that were almost perfectly translated in the first hypothesis by proposing different synonyms, the presence or absence of articles, or the inclusion of new words. In a similar way, AdaBoost iterations can deteriorate the translation quality, as they can choose worse phrases than in previous iterations.

However, when moving to a more complex task, the Europarl corpus between French and English, our results are not so good, as shown in Figure 2. The achieved improvement with respect to the baseline is smaller than that obtained with the Xerox corpus. In addition, most of these improvements are not significant.

6 Conclusions and future work

In this paper, an adaptation of AdaBoost algorithm to machine translation has been proposed. This AdaBoost version has been implemented and applied in some experiments.

Our results show that our proposal can achieve statistically significant improvements of translation quality in some corpora. Particularly, we present an important BLEU improvement when translating the Xerox corpus. Nevertheless, when working with the Europarl corpus the improvement is less important.

These results are quite appealing, and they encourage us to study in depth the possibilities that AdaBoost can bring to machine translation.

Another adaptations of AdaBoost should be analysed, especially in the reweighting step. With respect to the final hypothesis combination, some other

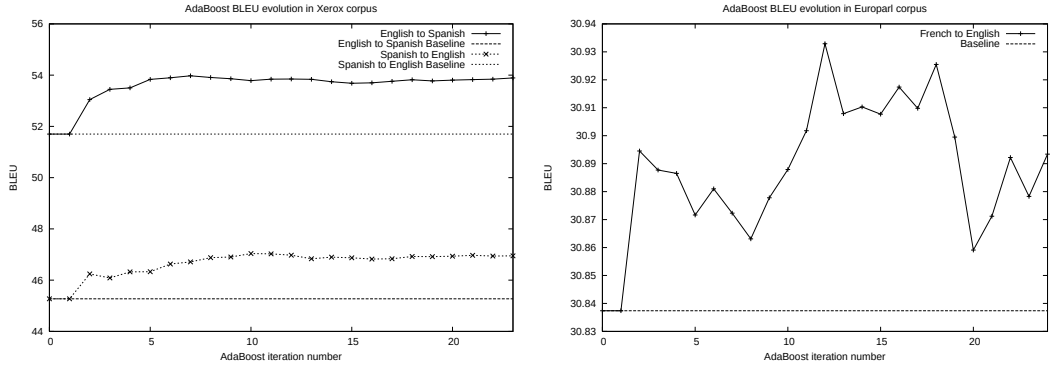

Fig. 2. *Evolution of translation quality (in terms of BLEU) when increasing the number of iterations of AdaBoost. Left plot shows results when translating the DevTest set of the Xerox corpus, from Spanish to English and vice versa. Translation quality of every iteration outperforms the baseline system (shown by a horizontal line in each case). Right plot shows results when working with the Europarl corpus, DevTest set, from French to English. In this case, results are not so good and more random per iteration number.*

more sophisticated alternatives can be considered. For instance, the creation of a lattice representation of the hypotheses and posterior extraction of the path with the lowest expected error [23]; the *ROVER* approach [24]; or other combination strategies [25].

Finally, an interesting property of AdaBoost is its ability to identify outliers, examples that are hard to learn, ambiguous or mislabeled [3]. Other boosting algorithms such as *BrownBoost* or *Gentle Adaboost* take advantage of this ability. They might be adapted to SMT in a similar way as AdaBoost.

References

1. R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
2. Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.
3. R. E. Schapire. The boosting approach to machine learning: an overview. In MSRI Workshop on Nonlinear Estimation and Classification, Berkeley, CA, USA, 2001.
4. R. Zhang and E. Sumita. Boosting statistical machine translation by lemmatization and linear interpolation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 181–184, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
5. H. Wu, H. Wang, and Z. Liu. Boosting statistical word alignment using labeled and unlabeled data. In *Proceedings of the COLING-ACL on Main conference poster*

sessions, pages 913–920, Sydney, Australia, 2006. Association for Computational Linguistics.

6. P. F. Brown, S. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, 1993.
7. J. Tomás and F. Casacuberta. Monotone statistical translation using word groups. In *Proceedings of the Machine Translation Summit VIII*, pages 357–361, Santiago de Compostela, Spain, 2001.
8. D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In *EMNLP02*, pages 133–139, Philadelphia, PA, USA, July 6-7 2002.
9. R. Zens, F.J. Och, and H. Ney. Phrase-based statistical machine translation. In *Advances in artificial intelligence. 25th Annual German Conference on Artificial Intelligence (KI 02), Aachen, Germany, September 16–22, Proceedings*, volume 2479 of *Lecture Notes on Artificial Intelligence*, pages 18–32. Springer Verlag, 2002.
10. R. Zens and H. Ney. Improvements in phrase-based statistical machine translation. In *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting*, pages 257–264, Boston, MA, May 2004.
11. P. Koehn, F.J. Och, and D. Marcu. Statistical phrase-based translation. Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL), May 27-June 1, Edmonton, Canada, 2003.
12. F. J. Och and H. Ney. Discriminative training and maximum entropy models for statistical machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Philadelphia, PA, USA, 2001.
13. B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall/CRC, May 1994.
14. P. 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. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Assoc. for Computational Linguistics*, pages 177–180, Prague, Czech Republic, June 2007.
15. F. J. Och. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, 2003.
16. SchlumbergerSema S.A, Instituto Tecnológico de Informática, Rheinisch Westfälische Technische Hochschule Aachen Lehrstul für atik VI, Recherche Appliquée en Linguistique Informatique Laboratory University of Montreal, Celer Soluciones, Société Gamma, and Xerox Research Centre Europe. TT2. TransType2 - computer assisted translation. Project technical annex., 2001.
17. P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, Phuket, Thailand, September 2005.
18. P. Koehn and C. Monz. Manual and automatic evaluation of machine translation between european languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City, NY, USA, June 2006.
19. R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? In *Proceedings of the IEEE*, volume 88, pages 1270–1278, 2000.

20. K. Papineni, S. Roukos, T. Ward, and W.-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA, USA, 2002.
21. P. Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, Barcelona, Spain, July 2004.
22. M. Bisani and H. Ney. Bootstrap estimates for confidence intervals in asr performance evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 409–412, Montréal, Canada, May 2004.
23. A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error ization in N-best list rescoring. In *Proc. Eurospeech '97*, pages 163–166, Rhodes, Greece, 1997.
24. J. Fiscus. A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (rover). In *Proceedings 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–352, Santa Barbara, CA, USA, 1997.
25. F. Huang and K. Papineni. Hierarchical system combination for machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 277–286, Prague, Czech Republic, June 2007.