LIMSI Submission for the WMT'13 Quality Estimation Task: an Experiment with n-gram Posteriors

Anil Kumar Singh
LIMSI
Orsay, France
anil@limsi.fr

Guillaume Wisniewski
Université Paris Sud
LIMSI
Orsay, France
wisniews@limsi.fr

François Yvon
Université Paris Sud
LIMSI
Orsay, France
yvon@limsi.fr

Abstract

This paper describes the machine learning algorithm and the features used by LIMSI for the Quality Estimation Shared Task. Our submission mainly aims at evaluating the usefulness for quality estimation of n-gram posterior probabilities that quantify the probability for a given n-gram to be part of the system output.

1 Introduction

The dissemination of statistical machine translation (SMT) systems in the professional translation industry is still limited by the lack of reliability of SMT outputs, the quality of which varies to a great extent. In this context, a critical piece of information would be for MT systems to assess their output translations with automatically derived quality measures. This problem is the focus of a shared task, the aim of which is to predict the quality of a translation without knowing any human reference(s).

To the best of our knowledge, all approaches so far have tackled quality estimation as a supervised learning problem (He et al., 2010; Soricut and Echihabi, 2010; Specia et al., 2010; Specia, 2011). A wide variety of features have been proposed, most of which can be described as loosely 'linguistic' features that describe the source sentence, the target sentence and the association between them (Callison-Burch et al., 2012). Surprisingly enough, information used by the decoder to choose the best translation in the search space, such as its internal scores, have hardly been considered and never proved to be useful. Indeed, it is well-known that these scores are hard to interpret and to compare across hypotheses. Furthermore, mapping scores of a linear classifier (such as the scores estimated by MERT) into consistent probabilities is a difficult task (Platt, 2000; Lin et al., 2007).

This work aims at assessing whether information extracted from the decoder search space can help to predict the quality of a translation. Rather than using directly the decoder score, we propose to consider a finer level of information, the n-*gram posterior probabilities* that quantifies the probability for a given n-gram to be part of the system output. These probabilities can be directly interpreted as the confidence the system has for a given n-gram to be part of the translation. As they are directly derived from the number of hypotheses in the search space that contains this n-gram, these probabilities might be more reliable than the ones estimated from the decoder scores.

We first quickly review, in Section 2, the n-gram posteriors introduced by (Gispert et al., 2013) and explain how they can be used in the QE task; we then describe, in Section 3 the different systems that have developed for our participation in the WMT'13 shared task on Quality Estimation and assess their performance in Section 4.

2 n-gram Posterior Probabilities in SMT

Our contribution to the WMT'13 shared task on quality estimation relies on n-gram posteriors. For the sake of completeness, we will quickly formalize this notion and summarize the method proposed by (Gispert et al., 2013) to efficiently compute them. We will then describe preliminary experiments to assess their usefulness for predicting the quality of a translation hypothesis.

2.1 Computing n-gram Posteriors

For a given source sentence F, the n-gram posterior probabilities quantifies the probability for a given n-gram to be part of the system output. Their computation relies on all the hypotheses considered by a SMT system during decoding: intuitively, the more hypotheses a n-gram appears in, the more confident the system is that this n-gram is part of the 'correct' translation, and the

higher its posterior probability is. Formally, the posterior of a given n-gram u is defined as:

$$P(u|\mathcal{E}) = \sum_{(A,E)\in\mathcal{E}} \delta_u(E) \cdot P(E, A|F)$$

where the sum runs over the translation hypotheses contained in the search space $\mathcal{E}$ (generally represented as a lattice); $\delta_u(E)$ has the value 1 if u occurs in the translation hypothesis E and 0 otherwise and $P(E, A|F)$ is the probability that the source sentence F is translated by the hypothesis E using a derivation A. Following (Gispert et al., 2013), this probability is estimated by applying a soft-max function to the score of the decoder:

$$P(A, E|F) = \frac{\exp(\alpha \times H(E, A, F))}{\sum_{(A',E')\in\mathcal{E}} \exp(H(E', A', F))}$$

where the decoder score $H(E, A, F)$ is typically a linear combination of a handful of features, the weights of which are estimated by MERT (Och, 2003).

n-gram posteriors therefore aggregate two pieces of information: first, the number of paths in the lattice (i.e. the number of translation hypotheses of the search path) the n-gram appears in; second, the decoder scores of these paths that can be roughly interpreted as a quality of the path.

Computing $P(u|\mathcal{E})$ requires to enumerate all n-gram contained in $\mathcal{E}$ and to count the number of paths in which this n-gram appears at least once. An efficient method to perform this computation in a single traversal of the lattice is described in (Gispert et al., 2013). This algorithm has been reimplemented[1] to generate the posteriors used in this work.

2.2 Analysis of n-gram Posteriors

Figure 1 represents the distribution of n-gram posteriors on the training set of the task 1-1. This distribution is similar to the ones observed for task 1-3 and for higher n-gram orders. It appears that, the distribution is quite irregular and has two modes. The minor modes corresponds to n-grams that appear in almost every translation hypotheses and have posterior probability close to 1. Further analyses show that these n-grams are mainly made of stop words and of out-of-vocabulary words. The major mode corresponds to very small n-gram posteriors (less than 10^{-1}) that the system has only

a very small confidence in producing. The number of n-grams that have such a small posterior suggests that most n-grams occur only in a small number of paths.
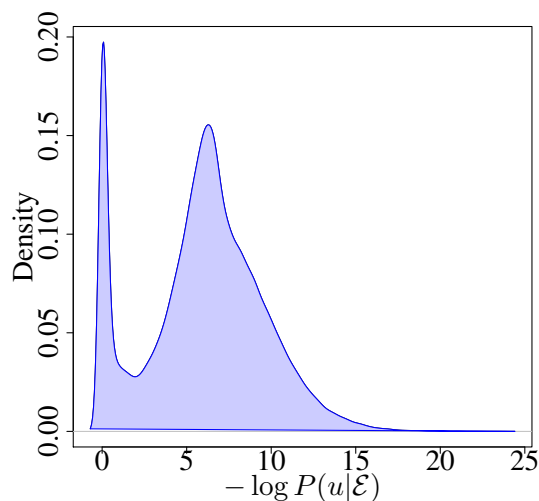

Figure 1: Distribution of the unigram posteriors observed on the training set of the task 1-1

Using n-gram posteriors to predict the quality of translation raises a representation issue: the number of n-grams contained in a sentence varies with the sentence length (and hence with the number of posteriors) but this information needs to be represented in a fixed-length vector describing the sentence. Similarly to what is usually done in the quality estimation task, we chose to represent posteriors probability by their histogram: for a given n-gram order, each posterior is mapped to a bin; each bin is then represented by a feature equal to the number of n-gram posteriors it contains. To account for the irregular distribution of posteriors, bin breaks are chosen on the training set so as to ensure that each bin contains the same number of examples. In our experiments, we considered a partition of the training data into 20 bins.

3 Systems Description

LIMSI has participated to the tasks 1-1 (prediction of the hTER) and 1-3 (prediction of the post-edition time). Similar features and learning algorithms have been considered for the two tasks. We will first quickly describe them before discussing the specific development made for task 1-3.

3.1 Features

In addition to the features described in the previous section, 176 'standard' features for quality estimation have been considered. The full list of features we have considered is given in (Wisniewski et al., 2013) and the features set can be downloaded from our website.[2] These features can be classified into four broad categories:

- **Association Features**: Measures of the quality of the 'association' between the source and the target sentences like, for instance, features derived from the IBM model 1 scores;

- **Fluency Features**: Measures of the 'fluency' or the 'grammaticality' of the target sentence such as features based on language model scores;

- **Surface Features**: Surface features extracted mainly from the source sentence such as the number of words, the number of out-of-vocabulary words or words that are not aligned;

- **Syntactic Features**: some simple syntactic features like the number of nouns, modifiers, verbs, function words, WH-words, number words, etc., in a sentence;

These features sets differ, in several ways, from the baseline feature set provided by the shared task organizers. First, in addition to features derived from a language model, it also includes several features based on large span continuous space language models (Le et al., 2011). Such language models have already proved their efficiency both for the translation task (Le et al., 2012) and the quality estimation task (Wisniewski et al., 2013). Second, each feature was expanded into two 'normalized forms' in which their value was divided either by the source length or the target length and, when relevant, into a 'ratio form' in which the feature value computed on the target sentence is divided by its value computed in the source sentence. At the end, when all possible feature expansions are considered, each example is described by 395 features.

3.2 Learning Methods

The main focus of this work is to study the relevance of features for quality estimation; therefore, only very standard learning methods were used in our work. For this year submission both random forests (Breiman, 2001) and elastic net regression (Zou and Hastie, 2005) have been used. The capacity of random forests to take into account complex interactions between features has proved to be a key element in the results achieved in our experiments with last year campaign datasets (Zhuang et al., 2012). As we are considering a larger features set this year and the number of examples is comparatively quite small, we also considered elastic regression, a linear model trained with L_1 and L_2 priors as regularizers, hoping that training a sparse model would reduce the risk of overfitting.

In this study, we have used the implementation provided by `scikit-learn` (Pedregosa et al., 2011). As detailed in Section 4.1, cross-validation has been used to choose the hyper-parameters of all regressors, namely the number of estimators, the maximal depth of a tree and the minimum number of examples in a leaf for the random forests and the importance of the L_1 and the L_2 regularizers for the elastic net regressor.

3.3 System for Task 1-3

Like task 1-1, task 1-3 is a regression task that aims at predicting the time needed to post-edit a translation hypothesis. From a machine learning point of view, this task differs from task 1-1 in three aspects. First, the distributed training set is much smaller: it is made of only 803 examples, which increases the risk of overfitting. Second, contrary to hTER scores, post-edition time is not normalized and the label of this task can take any positive value. Finally and most importantly, as shown in Figure 2, the label distributions estimated on the training set has a long tail which indicates the presence of several outliers: in the worse case, it took more than 18 minutes to correct a single sentence made of 35 words! Such a long post-edition time most certainly indicates that the corrector has been distracted when post-editing the sentence rather than a true difficulty in the post-edition.

These outliers have a large impact on training and on testing, as their contributions to both MAE
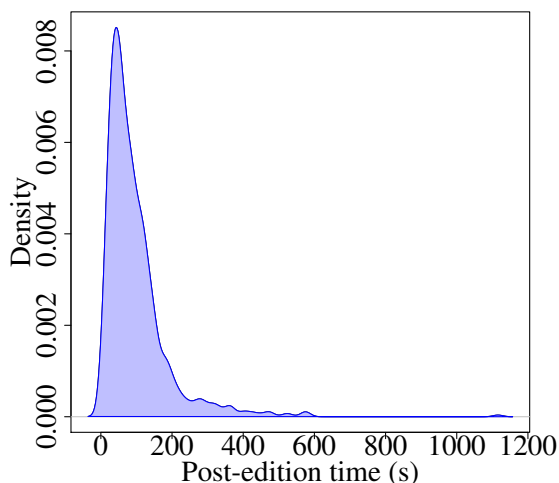
Figure 2: Kernel density estimate of the post-edition time distribution used as label in task 1-3.

and MSE,[3] directly depends on label values and can therefore be very large in the case of outliers. For instance, a simple ridge regression with the baseline features provided by the shared task organizer achieves a MAE of 42.641 ± 2.126 on the test set. When all the examples having a label higher than 300 are removed from the training set, the MAE drops to 41.843 ± 4.134. When outliers are removed from *both* the training and the test sets, the MAE further drops to 32.803 ± 1.673. These observations indicate that special care must be taken when collecting the data and that, maybe, post-edition times should be clipped to provide a more reliable estimation of the predictor performance.

In the following (and in our submission) only examples for which the post-edition time was less than 300 seconds were considered.

4 Results

4.1 Experimental Setup

We have tested different combinations of features and learning methods using a standard metric for regression: *Mean Absolute Error* (MAE) defined by:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

where n is the number of examples, y_i and $\hat{y}_i$ the true label and predicted label of the ith example. MAE can be understood as the averaged error made in predicting the quality of a translation.

Performance of both task 1-1 and task 1-3[4] was also evaluated by the Spearman rank correlation coefficient ρ that assesses how well the relationship between two variables can be described using a monotonic function. While the value of the correlation coefficient is harder to interpret as it not directly related to the value to predict, it can be used to compare the performance achieved when predicting different measures of the post-editing effort. Indeed, several sentence-level (or document level) annotation types can be used to reflect translation quality (Specia, 2011), such as the time needed to post-edit a translation hypothesis, the hTER, or qualitative judgments as it was the case for the shared task of WMT 2012. Comparing directly these different settings is complicated, since each of them requires to optimize a different loss, and even if the losses are the same, their actual values will depend on the actual annotation to be predicted (refer again to the discussion in (Specia, 2011, p5)). Using a metric that relies on the predicted rank of the example rather than the actual value predicted allows us to directly compare the performance achieved on the two tasks.

As the labels for the different tasks were not released before the evaluation, all the reported results are obtained on an 'internal' test set, made of 20% of the data released by the shared task organizers as 'training' data. The remaining data were used to train the regressor in a 10 folds cross-validation setting. In order to get reliable estimate of our methods performances, we used bootstrap resampling (Efron and Tibshirani, 1993) to compute confidence intervals of the different scores: 10 random splits of the data into a training and sets were generated; a regressor was then trained and tested for each of these splits and the resulting confidence intervals at 95% computed.

4.2 Results

Table 1and Table 2 contain the results achieved by our different conditions. We used, as a baseline, the set of 17 features released by the shared task organizers.

It appears that the differences in MAE between

---

[3]The two standard loss functions used to train and evaluate a regressor

[4]The Spearman ρ was an official metric only for task 1-1. For reasons explained in this paragraph, we also used it to evaluate our results for task 1-3.

the different configurations are always very small and hardly significant. However, the variation of the Spearman ρ are much larger and the difference observed are practically significant when the interpretation scale of (Landis and Koch, 1977) is used. We will therefore mainly consider ρ in our discussion.

For the two tasks 1-1 and 1-3, the features we have designed allow us to significantly improve prediction performance in comparison to the baseline. For instance, for task 1-1, the correlation is almost doubled when the features described in Section 3.1 are used. As expected, random forests are overfitting and did not manage to outperform a simple linear classifier. That is why we only used the elastic net method for our official submission. Including posterior probabilities in the feature set did not improve performance much (except when only the baseline features are considered) and sometimes even hurt performance. This might be caused by an overfitting problem, the training set becoming too small when new features are added. We are conducting further experiments to explain this paradoxical observation.

Another interesting observation that can be made looking at the results of Table 1 and Table 2 is that the prediction of the post-edition time seems to be easier than the prediction of the hTER: using the same classifiers and the same features, the performance for the former task is always far better than the performance for the latter.

5   Conclusion

In this paper, we described our submission to the WMT'13 shared task on quality estimation. We have explored the use of posteriors probability, hoping that information about the search space could help in predicting the quality of a translation. Even if features derived from posterior probabilities have shown to have only a very limited impact, we managed to significantly improve the baseline with a standard learning method and simple features. Further experiments are required to understand the reasons of this failure.

Our results also highlight the need to continue gathering high-quality resources to train and investigate quality estimation systems: even when considering few features, our systems were prone to overfitting. Developing more elaborated systems will therefore only be possible if more training resource is available. Our experiments also

stress that both the choice of the quality measure (i.e. the quantity to predict) and of the evaluation metrics for quality estimation are still open problems.

6   Acknowledgments

References

Leo Breiman. 2001. Random forests. *Mach. Learn.*, 45(1):5–32, October.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

B. Efron and R. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall.

Adrià Gispert, Graeme Blackwood, Gonzalo Iglesias, and William Byrne. 2013. N-gram posterior probability confidence measures for statistical machine translation: an empirical study. *Machine Translation*, 27(2):85–114.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging smt and tm with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630, Uppsala, Sweden, July. Association for Computational Linguistics.

R. J. Landis and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Hai Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured Output Layer Neural Network Language Model. In *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, pages 5524–5527, Prague, Czech Republic.

Hai-Son Le, Thomas Lavergne, Alexandre Allauzen, Marianna Apidianaki, Li Gong, Aurélien Max, Artem Sokolov, Guillaume Wisniewski, and François Yvon. 2012. Limsi @ wmt12. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 330–337, Montréal, Canada, June. Association for Computational Linguistics.

	MAE		ρ	
	train	test	train	test
Baseline Features				
RandomForest	0.109 ± 0.013	0.130 ± 0.004	0.405 ± 0.008	0.314 ± 0.016
Elastic	0.127 ± 0.001	0.129 ± 0.003	0.336 ± 0.004	0.319 ± 0.015
'Linguistic' Features				
RandomForest	0.082 ± 0.019	0.118 ± 0.003	0.689 ± 0.003	0.625 ± 0.009
Elastic	0.107 ± 0.004	0.115 ± 0.003	0.705 ± 0.009	0.660 ± 0.009
'Linguistic' Features + posteriors				
RandomForest	0.088 ± 0.017	0.116 ± 0.003	0.694 ± 0.003	0.615 ± 0.014
Elastic	0.105 ± 0.006	0.114 ± 0.002	0.699 ± 0.007	0.662 ± 0.011

Table 1: Results for the task 1-1

	MAE		ρ	
	train	test	train	test
Baseline Features				
RandomForest	25.145 ± 3.745	33.279 ± 1.687	0.669 ± 0.007	0.639 ± 0.017
Elastic	32.776 ± 0.795	33.702 ± 2.328	0.678 ± 0.006	0.657 ± 0.018
Baseline Features + Posteriors				
RandomForest	33.707 ± 0.309	35.646 ± 0.889	0.674 ± 0.004	0.637 ± 0.017
Elastic	31.487 ± 0.261	32.922 ± 0.789	0.698 ± 0.004	0.681 ± 0.016
'Linguistic' Features				
RandomForest	25.236 ± 4.400	33.017 ± 1.582	0.735 ± 0.007	0.666 ± 0.023
Elastic	28.706 ± 1.273	31.630 ± 1.612	0.760 ± 0.006	0.701 ± 0.017
'Linguistic' Features + Posteriors				
RandomForest	22.951 ± 3.903	33.013 ± 1.514	0.741 ± 0.003	0.695 ± 0.013
Elastic	28.911 ± 1.020	31.865 ± 1.636	0.761 ± 0.008	0.710 ± 0.017

Table 2: Results for the task 1-3

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2007. A note on platt's probabilistic outputs for support vector machines. *Mach. Learn.*, 68(3):267–276, October.

Franz Josef 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, Sapporo, Japan.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

John C. Platt, 2000. *Probabilities for SV Machines*, pages 61–74. MIT Press.

Radu Soricut and Abdessamad Echihabi. 2010. Trustrank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 612–621, Uppsala, Sweden, July. Association for Computational Linguistics.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50, March.

Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th conference of EAMT*, pages 73–80, Leuven, Belgium.

Guillaume Wisniewski, Anil Kumar Singh, and François Yvon. 2013. Quality estimation for machine translation: Some lessons learned. *Machine Translation*. accepted for publication.

Yong Zhuang, Guillaume Wisniewski, and François Yvon. 2012. Non-linear models for confidence estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 157–162, Montréal, Canada, June. Association for Computational Linguistics.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320.