LIMSI Submission for WMT'14 QE Task

Guillaume Wisniewski and **Nicolas Pécheux** and **Alexandre Allauzen** and **François Yvon**
Université Paris Sud and LIMSI-CNRS
91 403 ORSAY CEDEX, France
{wisniews, pecheux, allauzen, yvon}@limsi.fr

Abstract

This paper describes LIMSI participation to the WMT'14 Shared Task on Quality Estimation; we took part to the word-level quality estimation task for English to Spanish translations. Our system relies on a random forest classifier, an ensemble method that has been shown to be very competitive for this kind of task, when only a few dense and continuous features are used. Notably, only 16 features are used in our experiments. These features describe, on the one hand, the quality of the association between the source sentence and each target word and, on the other hand, the fluency of the hypothesis. Since the evaluation criterion is the f_1 measure, a specific tuning strategy is proposed to select the optimal values for the hyper-parameters. Overall, our system achieves a 0.67 f_1 score on a randomly extracted test set.

1 Introduction

This paper describes LIMSI submission to the WMT'14 Shared Task on Quality Estimation. We participated in the word-level quality estimation task (Task 2) for the English to Spanish direction. This task consists in predicting, for each word in a translation hypothesis, whether this word should be post-edited or should rather be kept unchanged.

Predicting translation quality *at the word level* raises several interesting challenges. First, this is a (relatively) new task and the best way to formulate and evaluate it has still to be established. Second, as most works on quality estimation have only considered prediction at the sentence level, it is not clear yet which features are really effective to predict quality at the word and a set of baseline features has still to be found. Finally, several characteristic of the task (the limited number of training examples, the unbalanced classes, etc.) makes the use of 'traditional' machine learning algorithms difficult. This papers describes how we addressed this different issues for our participation to the WMT'14 Shared Task.

The rest of this paper is organized as follows. Section 2 gives an overview of the shared task data that will justify some of the design decisions we made. Section 3 describes the different features we have considered and Section 4, the learning methods used to estimate the classifiers parameters. Finally the results of our models are presented and analyzed in Section 5.

2 World-Level Quality Estimation

WMT'14 shared task on quality estimation number 2 consists in predicting, for each word of a translation hypothesis, whether this word should be post-edited (denoted by the BAD label) or should be kept unchanged (denoted by the OK label). The shared task organizers provide a bilingual dataset from English to Spanish[1] made of translations produced by three different MT systems and by one human translator; these translations have then been annotated with word-level labels by professional translators. No additional information about the systems used, the derivation of the translation (such as the lattices or the alignment between the source and the best translation hypothesis) or the tokenization applied to identify words is provided.

The distributions of the two labels for the different systems is displayed in Table 1. As it could be expected, the class are, overall, unbalanced and the systems are of very different quality: the proportion of BAD and OK labels highly depends on the system used to produce the translation hypotheses. However, as our preliminary experiments have shown, the number of examples is

[1] We did not consider the other language pairs.

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 348–354,
Baltimore, Maryland USA, June 26–27, 2014.

too small to train a different confidence estimation system for each system.

The distribution of the number of BAD labels per sentence is very skewed: on average, one word out of three (precisely 35.04%) in a sentence is labeled as BAD but the median of the distribution of the ratio of word labeled BAD in a sentence is 20% and its standard deviation is pretty high (34.75%). Several sentences have all their words labeled as either OK or BAD, which is quite surprising as the sentences of the corpus for Task 2 have been selected because there were 'near miss translations' that is to say translations that should have contained no more that 2 or 3 errors.

Another interesting finding is that the proportion of word to post-edit is the same across the different parts-of-speech (see Table 2).[2]

Table 1: Number of examples and distribution of labels for the different systems on the training set

System	#sent.	#words	% OK	% BAD
1	791	19,456	75.48	24.52
2	621	14,620	59.11	40.89
3	454	11,012	59.76	40.24
4	90	2,296	36.85	63.15
Total	1,956	47,384	64.90	35.10

Table 2: Distribution of labels according to the POS on the training set

POS	% in train	% BAD
NOUN	23.81	35.02
ADP	15.06	35.48
DET	14.90	32.88
VERB	14.64	41.26
PUNCT	10.92	27.26
ADJ	6.61	35.68
CONJ	5.04	30.77
PRON	4.58	43.15
ADV	4.39	36.56

As the classes are unbalanced, prediction performance will be evaluated in terms of precision, recall and f_1 score computed on the BAD label. More precisely, if the number of true positive (i.e. BAD word predicted as BAD), false positive (OK word predicted as BAD) and false negative (BAD word predicted as OK) are denoted tp_{BAD}, fp_{BAD} and fn_{BAD}, respectively, the quality of a confidence estimation system is evaluated by the three following metrics:

$$p_{\text{BAD}} = \frac{\mathrm{tp}_{\text{BAD}}}{\mathrm{tp}_{\text{BAD}} + \mathrm{fp}_{\text{BAD}}} \quad (1)$$

$$r_{\text{BAD}} = \frac{\mathrm{tp}_{\text{BAD}}}{\mathrm{tp}_{\text{BAD}} + \mathrm{fn}_{\text{BAD}}} \quad (2)$$

$$f_1 = \frac{2 \cdot p_{\text{BAD}} \cdot r_{\text{BAD}}}{p_{\text{BAD}} + r_{\text{BAD}}} \quad (3)$$

3 Features

In our experiments, we used 16 features to describe a given target word t_i in a translation hypothesis $\mathbf{t} = (t_j)_{j=1}^{m}$. To avoid sparsity issues we decided not to include any lexicalized information such as the word or the previous word identities. As the translation hypotheses were generated by different MT systems, no white-box features (such as word alignment or model scores) are considered. Our features can be organized in two broad categories:

Association Features These features measure the quality of the 'association' between the source sentence and a target word: they characterize the probability for a target word to appear in a translation of the source sentence. Two kinds of association features can be distinguished.

The first one is derived from the lexicalized probabilities $p(t|s)$ that estimate the probability that a source word s is translated by the target word t_j. These probabilities are aggregated using an arithmetic mean:

$$p(t_j|\mathbf{s}) = \frac{1}{n}\sum_{i=1}^{n} p(t_j|s_i) \quad (4)$$

where $\mathbf{s} = (s_i)_{i=1}^{n}$ is the source sentence (with an extra `NULL` token). We assume that $p(t_j|s_i) = 0$ if the words t_j and s_i have never been aligned in the train set and also consider the geometric mean of the lexicalized probabilities, their maximum value (i.e. $\max_{s\in\mathbf{s}} p(t_j|s)$) as well as a binary feature that fires when the target word t_j is not in the lexicalized probabilities table.

The second kind of association features relies on pseudo-references, that is to say, translations of the source sentence produced by an independent MT system. Many works have considered

[2]We used FreeLing (`http:nlp.lsi.upc.edu/freeling/`) to predict the POS tags of the translation hypotheses and, for the sake of clarity, mapped the 71 tags used by FreeLing to the 11 universal POS tags of Petrov et al. (2012).

pseudo-references to design new MT metrics (Albrecht and Hwa, 2007; Albrecht and Hwa, 2008) or for confidence estimation (Soricut and Echihabi, 2010; Soricut and Narsale, 2012) but, to the best of our knowledge, this is the first time that they are used to predict confidence at the word level.

Pseudo-references are used to define 3 binary features which fire if the target word is in the pseudo-reference, in a 2-gram shared between the pseudo-reference and the translation hypothesis or in a common 3-gram, respectively. The lattices representing the search space considered to generate these pseudo-references also allow us to estimate the *posterior probability* of a target word that quantifies the probability that it is part of the system output (Gispert et al., 2013). Posteriors aggregate two pieces of information for each word in the final hypothesis: first, all the paths in the lattice (i.e. the number of translation hypotheses in the search space) where the word appears in are considered; second, the decoder scores of these paths are accumulated in order to derive a confidence measure at the word level. In our experiments, we considered pseudo-references and lattices produced by the n-gram based system developed by our team for last year WMT evaluation campaign (Allauzen et al., 2013), that has achieved very good performance.

Fluency Features These features measure the 'fluency' of the target sentence and are based on different language models: a 'traditional' 4-gram language model estimated on WMT monolingual and bilingual data (the language model used by our system to generate the pseudo-references); a continuous-space 10-gram language model estimated with SOUL (Le et al., 2011) (also used by our MT system) and a 4-gram language model based on Part-of-Speech sequences. The latter model was estimated on the Spanish side of the bilingual data provided in the translation shared task in 2013. These data were POS-tagged with FreeLing (Padró and Stanilovsky, 2012).

All these language models have been used to define two different features :

- the probability of the word of interest $p(t_j|h)$ where $h = t_{j-1}, ..., t_{j-n+1}$ is the history made of the $n-1$ previous words or POS
- the ratio between the probability of the sentence and the 'best' probability that can be achieved if the target word is replaced by any other word (i.e. $\max_{v \in \mathcal{V}} p(t_1, ..., t_{j-1}, v, t_{j+1}, ..., t_m)$ where the max runs over all the words of the vocabulary).

There is also a feature that describes the back-off behavior of the conventional language model: its value is the size of the largest n-gram of the translation hypothesis that can be estimated by the language model without relying on back-off probabilities.

Finally, there is a feature describing, for each word that appears more than once in the train set, the probability that this word is labeled BAD. This probability is simply estimated by the ratio between the number of times this word is labeled BAD and the number of occurrences of this word.

It must be noted that most of the features we consider rely on models that are part of a 'classic' MT system. However their use for predicting translation quality at the word-level is not straightforward, as they need to be applied to sentences with a given unknown tokenization. Matching the tokenization used to estimate the model to the one used for collecting the annotations is a tedious and error-prone process and some of the prediction errors most probably result from mismatches in tokenization.

4 Learning Methods

4.1 Classifiers

Predicting whether a word in a translation hypothesis should be post-edited or not can naturally be framed as a binary classification task. Based on our experiments in previous campaigns (Singh et al., 2013; Zhuang et al., 2012), we considered random forest in all our experiments.[3]

Random forest (Breiman, 2001) is an ensemble method that learns many classification trees and predicts an aggregation of their result (for instance by majority voting). In contrast with standard decision trees, in which each node is split using the best split among all features, in a random forest the split is chosen randomly. In spite of this simple and counter-intuitive learning strategy, random forests have proven to be very good 'out-of-the-box' learners. Random forests have achieved very good performance in many similar

[3] we have used the implementation provided by `scikit-learn` (Pedregosa et al., 2011).

tasks (Chapelle and Chang, 2011), in which only a few dense and continuous features are available, possibly because of their ability to take into account complex interactions between features and to automatically partition the continuous features value into a discrete set of intervals that achieves the best classification performance.

As a baseline, we consider logistic regression (Hastie et al., 2003), a simple linear model where the parameters are estimated by maximizing the likelihood of the training set.

These two classifiers do not produce only a class decision but yield an instance probability that represents the degree to which an instance is a member of a class. As detailed in the next section, thresholding this probability will allow us to directly optimize the f_1 score used to evaluate prediction performance.

4.2 Optimizing the f_1 Score

As explained in Section 2, quality prediction will be evaluated in terms of f_1 score. The learning methods we consider can not, as most learning method, directly optimize the f_1 measure during training, since this metric does not decompose over the examples. It is however possible to take advantage of the fact that they actually estimate a probability to find the largest f_1 score on the training set.

Indeed these probabilities are used with a threshold (usually 0.5) to produce a discrete (binary) decision: if the probability is above the threshold, the classifier produces a positive output, and otherwise, a negative one. Each threshold value produces a different trade-off between true positives and false positives and consequently between recall and precision: as the the threshold becomes lower and lower, more and more example are assigned to the positive class and recall increase at the expense of precision.

Based on these observations, we propose the following three-step method to optimize the f_1 score on the training set:

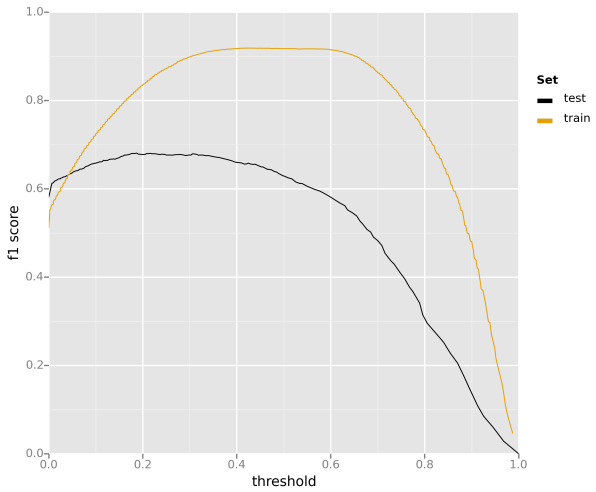
1. the classifier is first trained using the 'standard' learning procedure that optimizes either the 0/1 loss (for random forest) or the likelihood (for the logistic regression);

2. all the possible trade-offs between recall and precision are enumerated by varying the threshold; exploiting the monotonicity of thresholded classifications,[4] this enumeration can be efficiently done in $\mathcal{O}(n \cdot \log n)$ and results in at most n threshold values, where n is the size of the training set (Fawcett, 2003);

3. all the f_1 scores achieved for the different thresholds found in the previous step are evaluated; there are strong theoretical guarantees that the optimal f_1 score that can be achieved on the training set is one of these values (Boyd and Vandenberghe, 2004).

Figure 1 shows how f_1 score varies with the decision threshold and allows to assess the difference between the optimal value of the threshold and its default value (0.5).

Figure 1: Evolution of the f_1 score with respect to the threshold used to transform probabilities into binary decisions



5 Experiments

The features and learning strategies described in the two previous sections were evaluated on the English to Spanish datasets. As no official development set was provided by the shared task organizers, we randomly sampled 200 sentences from the training set and use them as a test set throughout the rest of this article. Preliminary experiments show that the choice of this test has a very low impact on the classification performance. The different hyper-parameters of the training algorithm

[4] Any instance that is classified positive with respect to a given threshold will be classified positive for all lower thresholds as well.

Table 3: Prediction performance for the two learning strategies considered

Classifier	thres.	r_{BAD}	p_{BAD}	f_1
Random forest	0.43	0.64	0.69	0.67
Logistic regression	0.27	0.51	0.72	0.59

were chosen by maximizing classification performance (as evaluated by the f_1 score) estimated on 150 sentences of the training set kept apart as a validation set.

Results for the different learning algorithms considered are presented in Table 3. Random forest clearly outperforms a simple logistic regression, which shows the importance of using non-linear decision functions, a conclusion at pair with our previous results (Zhuang et al., 2012; Singh et al., 2013).

The overall performance, with a f_1 measure of 0.67, is pretty low and in our opinion, not good enough to consider using such a quality estimation system in a computer-assisted post-edition context. However, as shown in Table 4, the prediction performance highly depends on the POS category of the words: it is quite good for 'plain' words (like verb and nouns) but much worse for other categories.

There are two possible explanations for this observation: predicting the correctness of some morpho-syntaxic categories may be intrinsically harder (e.g. for punctuation the choice of which can be highly controversial) or depend on information that is not currently available to our system. In particular, we do not consider any information about the structure of the sentence and about the labels of the context, which may explain why our system does not perform well in predicting the labels of determiners and conjunctions. In both cases, this result brings us to moderate our previous conclusions: as a wrong punctuation sign has not the same impact on translation quality as a wrong verb, our system might, regardless of its f_1 score, be able to provide useful information about the quality of a translation. This also suggests that we should look for a more 'task-oriented' metric.

Finally, Figure 2 displays the *importance* of the different features used in our system. Random forests deliver a quantification of the importance of a feature with respect to the predictability of the target variable. This quantification is derived from the position of a feature in a decision tree: features used in the top nodes of the trees, which contribute to the final prediction decision of a larger fraction of the input samples, play a more important role than features used near the leaves of the tree. It appears that, as for our previous experiments (Wisniewski et al., 2013), the most relevant feature for predicting translation quality is the feature derived from the SOUL language model, even if other fluency features seem to also play an important role. Surprisingly enough, features related to the pseudo-reference do not seem to be useful. Further experiments are needed to explain the reasons of this observation.

Table 4: Prediction performance for each POS tag

System	f_1
VERB	0.73
PRON	0.72
ADJ	0.70
NOUN	0.69
ADV	0.69
overall	0.67
DET	0.62
ADP	0.61
CONJ	0.57
PUNCT	0.56

6 Conclusion

In this paper we described the system submitted for Task 2 of WMT'14 Shared Task on Quality Estimation. Our system relies on a binary classifier and consider only a few dense and continuous features. While the overall performance is pretty low, a fine-grained analysis of the errors of our system shows that it can predict the quality of plain words pretty accurately which indicates that a more 'task-oriented' evaluation may be needed.

Acknowledgments

This work was partly supported by ANR project Transread (ANR-12-CORD-0015). Warm thanks to Quoc Khanh Do for his help for training a SOUL model for Spanish.

References

Joshua Albrecht and Rebecca Hwa. 2007. Regression for sentence-level mt evaluation with pseudo refer-

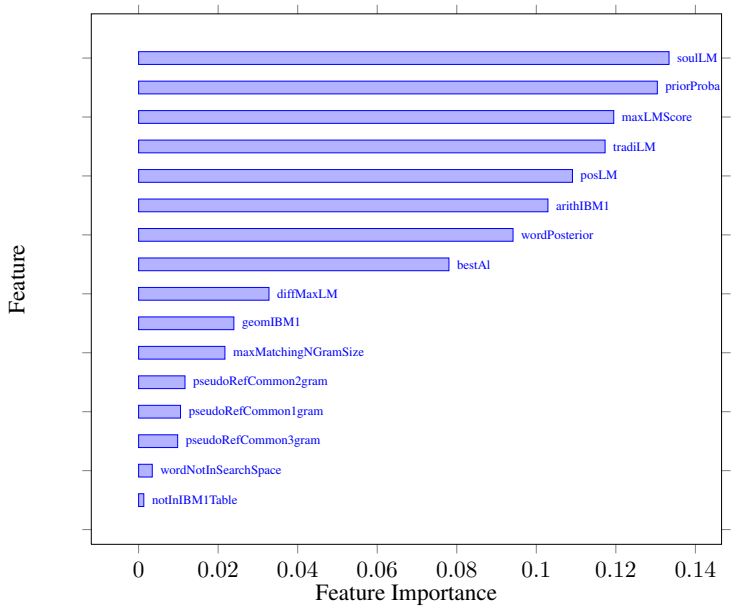


Figure 2: Features considered by our system sorted by their relevance for predicting translation errors

ences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 296–303, Prague, Czech Republic, June. ACL.

Joshua Albrecht and Rebecca Hwa. 2008. The role of pseudo references in MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 187–190, Columbus, Ohio, June. ACL.

Alexandre Allauzen, Nicolas Pécheux, Quoc Khanh Do, Marco Dinarelli, Thomas Lavergne, Aurélien Max, Hai-Son Le, and François Yvon. 2013. LIMSI @ WMT13. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 62–69, Sofia, Bulgaria, August. ACL.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.

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

Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In Olivier Chapelle, Yi Chang, and Tie-Yan Liu, editors, *Yahoo! Learning to Rank Challenge*, volume 14 of *JMLR Proceedings*, pages 1–24. JMLR.org.

Tom Fawcett. 2003. ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report HPL-2003-4, HP Laboratories, Palo Alto.

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.

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. 2003. *The Elements of Statistical Learning*. Springer, July.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5524–5527. IEEE.

Lluís Padró and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May. ELRA.

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.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Anil Kumar Singh, Guillaume Wisniewski, and François Yvon. 2013. LIMSI submission for the WMT'13 quality estimation task: an experiment with n-gram posteriors. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 398–404, Sofia, Bulgaria, August. ACL.

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. ACL.

Radu Soricut and Sushant Narsale. 2012. Combining quality prediction and system selection for improved automatic translation output. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 163–170, Montréal, Canada, June. ACL.

Guillaume Wisniewski, Anil Kumar Singh, and François Yvon. 2013. Quality estimation for machine translation: Some lessons learned. *Machine Translation*, 27(3).

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. ACL.