LORIA System for the WMT15 Quality Estimation Shared Task

Langlois David
SMarT Group, LORIA
Inria, Villers-lès-Nancy, F-54600, France
Universit de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
david.langlois@loria.fr

Abstract

We describe our system for WMT2015 Shared Task on Quality Estimation, task 1, sentence-level prediction of post-edition effort. We use baseline features, Latent Semantic Indexing based features and features based on pseudo-references. SVM algorithm allows to estimate the linear regression between the features vectors and the HTER score. We use a selection algorithm in order to put aside needless features. Our best system leads to a performance in terms of Mean Absolute Error equal to 13.34 on official test while the official baseline system leads to a performance equal to 14.82.

1 Introduction

This paper describes the LORIA submission to the WMT'15 Shared Task on Quality Estimation. We participated to the Task 1. This task consists in predicting the edition effort needed to correct a translated sentence. The organizers provide English sentences automatically translated into Spanish, and the corresponding post-edited sentences. The edition effort is measured by edit-distance rate (HTER (Snover et al., 2006)) between the translated sentence and its post-edited version.

Classically, our system extracts numerical features from sentences and applies a machine learning approach between numeric vectors and HTER scores.

As last year, no information is given about the Machine Translation (MT) system used to build data. Therefore, it is only possible to use blackbox features, or to use other MT systems whom output is compared to the evaluated target sentence.

Our submission deals with the both kinds of features. First, we use a Latent Semantic Analysis approach to measure the lexical similarity between a source and a target sentence. To our knowledge, this approach has never been used in the scope of Quality Estimation. Second, we use the output of 3 online MT systems, and we extract information about the intersection between the evaluated target sentence and the 3 translated sentences by online systems. This intersection is measured in terms of shared 1,2,3,4-grams.

The paper is structured as follows. Section 2 give details about experimental protocol and used data. We describe the features we use in Section 3. Then, we give results (Section 4) and we conclude.

2 Experimental protocol and used corpus

In this section, we describe how we obtain results starting from training, development and test corpus. The training and development corpus are composed of a set of triplets. Each triplet is made up of a source sentence, its automatic translation, and a score representing the translation quality.

For our experiments, we use the corpora the organizers provide. The source language is English, the target language is Spanish. For each source sentence s, a machine translation system (unknown to the participants) gives a translation t (we keep notations s and t throughout this article for source and target sentences from the evaluation campaign data). t is manually post-edited into pe. The score of (s, t) is the HTER score between t and pe (noted $hter$).

We use the official training corpus tr composed of 11272 triplets $(s, t, hter)$, and the official development corpus dev composed of 1000 triplets.

For each triplet $(s_i, t_i, hter_i)$ in tr, we extract the features vector from (s_i, t_i) (see Section 3 for the list of the features we use), this leads to $v_{(s_i,t_i)}$. Then, we use the SVM algorithm in order to estimate the regression between the $v_{(s_i,t_i)}$ (i from 1 to 11272) and the $hter_i$. For this estimation, we use the LibSVM tool (Chang and Lin, 2011), with a Radial Basis Function (with default parameters:

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 323–329,
Lisboa, Portugal, 17-18 September 2015.

$C = 1, \lambda = \frac{1}{|v_{(s_i,t_i)}|}$).

Then, we use the obtained linear regression in order to predict the edit effort rate for each couple (s, t) from dev (or test corpus for final evaluation).

Filtering the features some features may not be useful because they provide more noise than information, or because training data is not sufficiently big to estimate the link between them and the scores. Therefore, it may be useful to apply an algorithm in order to select interesting features. For that, we use a backward algorithm (Guyon and Elisseeff, 2003) we yet described in (Langlois et al., 2012). This year, we did not use the initial step consisting in evaluating the correlations between features (see (Langlois et al., 2012)). The algorithm is applied on the dev corpus in order to minimise the MAE (Mean Absolute Error) score defined by $MAE(r, r') = \frac{\sum_{i=1}^{n} |r_i - r'_i|}{n}$ where r is the set of n predicted scores on dev, and r' is the set of HTER reference scores.

3 The features

We use three sources for our features. The first source is the baseline features. The second is based on information provided by Latent Semantic approach, and the third one is based on the information provided by 3 online MT systems.

3.1 The baseline features

These 17 features are provided by the organizers of the Quality Estimation Shared Task. They are extracted by the QuEst tool (Specia et al., 2013). We can find the list of these features in the QuEst website[1], (Specia et al., 2013) describe them precisely. Table 1 shows the list of these features. We can remark that no glassbox feature is used (no information about the translation process of the MT system is used). Moreover, there is not feature taking into account both the source and target sentences (basing on an external translation table for example). 13 features describe the source side, while only 4 describe the target side.

3.2 Latent Semantic Indexing Based Features

Latent Semantic Indexing (LSI) allows to measure the similarity between two documents. This measure is based on lexical contents of the both documents. To achieve this measure, the documents are projected into a Vector Space Model: one document is described by a numerical vector, two documents are compared by computing the distance between their corresponding vectors.

LSI has been applied to bilingual parallel corpora in the scope of Information Retrieval (Littman et al., 1998) and of measure of comparability of documents (Saad et al., 2014). Each document is composed of the pair $(source, target)$. The method describes the corpus by a $n \times m$ matrix M. n is the number of words in the union of source and target vocabularies. m is the number of parallel sentences (a 'document' can be simply a sentence). $M[i, j]$ is a numeric value representing the "presence" of word i in document j. This value can be the frequency of i in j, or the $tfidf$ value. This matrix is strongly sparse. Therefore, the LSI method applies a reduction of dimensions. Finally, it is possible to project a new document into the

id	S/T	description
1	S	number of tokens in s
2	T	number of tokens in t
3	S	average source token length
4	S	LM probability of source sentence
5	T	LM probability of target sentence
6	T	av. freq. of the target word in t
7	S	av. number of translations per word in s (as given by IBM 1 table thresholded such that $prob(t\|s) > 0.2$)
8	S	same as 7 but with $prob(t\|s) > 0.01$ and weighted by the inverse frequency of each word in the source corpus
9	S	% of unigrams in quartile 1 of frequency extracted from an external corpus
10	S	same as 9 for quartile 4
11	S	same as 9 for bigrams and quartile 1
12	S	same as 9 for bigrams and quartile 4
13	S	same as 9 for trigrams and quartile 1
14	S	same as 9 for trigrams and quartile 4
15	S	% of unigrams in s seen in an external corpus
16	S	number of punctuation marks in s
17	T	number of punctuation marks in t

Table 1: List of baseline features. id are given to refer later to a specific feature. S, T are for 'source' or 'target' feature.

[1] http://www.quest.dcs.shef.ac.uk/quest_files/features_blackbox_baseline_17

obtained low-dimension numeric space (called the LSI model).

The LSI method may be interesting for Quality Estimation because LSI allows to project a s sentence, and a t sentence into the same numeric space. In this space, each document is described by a numeric vector. We can compute the similarity between two vectors (two documents) by cosine distance. Two documents are similar if their lexical content is close. The interesting point for Quality Estimation is that similarity can model the 'proximity' between "dog" and "bark", "chien", "aboyer", (or "perro", "ladrar" in Spanish) for example because the input documents for building the LSI model are bilingual.

We propose to use this similarity as a feature for Quality Estimation. For that, we use a training set of $(source, target)$ sentences (actually, we use 2 different training corpus, see below). We build a corpus in which each document is made up of the concatenation of a $source$ sentence and its corresponding $target$ sentence. We build the matrix M of the $tfidf$ scores of the words in the $source$-$target$ sentences. This matrix has n lines (the number of different $source$ words + the number of different $target$ words occurring more than 1 in the training corpus) and m columns (the number of $source$-$target$ couples). Then, we have to choose the dimension of the reduced numeric space (this dimension is called the number of topics). We applied the LSI reduction to obtain a LSI model. In this LSI model, it is possible to project a $source$ sentence, or a $target$ sentence into the same numeric space. Then, the feature corresponding to a $(source, target)$ couple in development or test corpus is the cosine distance between the LSI vector corresponding to $source$, and the LSI vector corresponding to $target$.

We use two training corpus. the first one is tr the training corpus from the Quality Estimation Shared Task ($target$ is here pe because pe is a correct translation of s). This corpus is close to the experimental conditions, but it contains only 11272 sentences couples. This is quite low for the LSI approach. Therefore, we use also the English-Spanish part of the Europarl (Koehn, 2005) corpus composed of 2M sentences couples[2]. Each training corpus leads to one LSI model.

To synthesize, we extract a feature from a (s, t) couple in four steps:

1. **LSI** = buildLSI(training corpus,number of topics)
2. LSI_s = **LSI**(s)
3. LSI_t = **LSI**(t)
4. feature = cosine distance between LSI_s and LSI_t

LSI is a function which projects a sentence into the numeric LSI space. The number of topics is one crucial parameter of the LSI approach. In Section 4, we explore the performance of the LSI based features according to this parameter.

3.3 The Machine Translation systems based features

We propose here to use pseudo-references. The idea is to compare t with other translations of s, provided by other MT systems. We hypothesise that the more t and other target sentences from the same s share parts, the more correct t is.

Several online translation systems yet exist on the web, and a few of them provide API allowing to request translations. We used three online systems noted $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$[3]. We used each system $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$ to translate the sentences from tr and dev. Therefore, from each sentence s, we have four target sentences: t from the system we want to estimate the quality, $t_{\mathbb{A}}$ from system $\mathbb{A}$, $t_{\mathbb{B}}$ from system $\mathbb{B}$, and $t_{\mathbb{C}}$ from system $\mathbb{C}$.

For each online system, we define 9 features to describe how much t and t_X (X is $\mathbb{A}$, $\mathbb{B}$ or $\mathbb{C}$) share n-grams. Moreover, we define 4 features taking into account the three online systems together.

Pseudo-references has yet been used for Quality Estimation. (Luong et al., 2014) decide of the correctness of each word in t by checking its presence in two pseudo-references. The binary feature is based on the number of pseudo-references containing the evaluated word. (Wisniewski et al., 2014) define binary features for word-level Quality Estimation. These binary features indicate if the evaluated word occurs in a n-gram (n

[2]Release v7, http://www.statmt.org/europarl/

[3]We do not give the identity of these systems because one of them precises that its online service can not be used for evaluation purpose. Indeed, in the following experiments, we give results using or not each of the systems. These results do not allow to conclude that a system is better than another one (see Section 4), but a quick reading could lead to such a conclusion.

from 1 to 3) shared by t and the pseudo-reference sentence. (Wisniewski et al., 2014) do not precise the number of pseudo-references, but they use the lattice produced by their in-house system, this leads certainly to a high number of pseudo-references. (Luong et al., 2014; Wisniewski et al., 2014) works are applied to word-level Quality Estimation while we deal with sentence-level Quality Estimation. (Scarton and Specia, 2014) use features from pseudo-reference sentences for sentence-level quality estimation. The features they extract are classical measures of translation quality (BLEU, TER, METEOR, ROUGE) between t and pseudo-reference. (Scarton and Specia, 2014) cite different works (Soricut et al., 2012; Shah et al., 2013) using also these measures for Quality Estimation. Differently, in our work, we use n-grams statistics in order to measure the consensus between t and pseudo-references.

3.3.1 Amount of shared n-grams between t and t_X

We describe the intersection between t and each of $t_{\mathbb{A}}$, $t_{\mathbb{B}}$ and $t_{\mathbb{C}}$ by 9 features.

The first four ones are recall n-gram $R_{X,n}$:

$$R_{X,n}(t, t_X) = \frac{\sum_{ng \in t,\ |ng|=n} \delta(ng, t_X)}{|t_X|} \quad (1)$$

where X is $\mathbb{A}$, $\mathbb{B}$ or $\mathbb{C}$, ng is a n-gram of length n, $\delta(ng, t_X)$ is equal to 1 if ng is in t_X and equal to 0 otherwise, and $|t_X|$ is the number of n-gram in t_X. n takes its values between 1 and 4. Therefore, there are 4 features for each system.

The following four features are precision n-gram $P_{X,n}$, which are equivalent to $R_{X,n}(t, t_X)$, but the denominator is $|t|$. Here also, there are 4 features for each system.

For these 8 features, a n-gram in t_X is taken into account only one time. For example, if $t = a\ b\ a$, and $t_X = a\ b$, there is only one match for a when $n = 1$, even if there are two a in t.

The last feature is the maximum length words sequence from t that is also in t_X:

$$M(t, t_X) = \frac{\max[|ng|,\ s.t.\ ng \in t\ and\ ng \in t_X]}{|t|} \quad (2)$$

Each system leads to 9 features.

3.3.2 Taking into account the three online system together

We define 4 additional features which describe how many pseudo-references include a n-gram of t (n varies from 1 to 4). The idea is that if a n-gram from t occurs in 3 pseudo-references, it is likely a correct n-gram whereas if it occurs only in one pseudo-reference, it is more doubtful. These features are formalized by the following formula:

$$Inter(t, t_{\mathbb{A}}, t_{\mathbb{B}}, t_{\mathbb{C}}, n) = \frac{\sum_{i=1}^{i \leq |t|-n+1} \sum_{X \in \{\mathbb{A},\mathbb{B},\mathbb{C}\}} \delta(t_i^{i+n-1}, t_X)}{3 \times (|t|-n+1)} \quad (3)$$

where t_a^b is the words sequence from t starting at position a and ending at position b, and other notations are defined as previously. n takes values from 1 to 4. Therefore, this leads to 4 additional features. In the following, we use the acronym $Inter$ to refer to these 4 features.

Overall, our system deals with 50 features: 17 from baseline, 2 from LSI approach, 9 for each of the three online systems, and 4 from the combination of these three systems.

4 Results

4.1 Baseline features

Table 2 shows the results in terms of MAE on development corpus of each baseline feature used alone (only one feature is used to predict the HTER score). The feature ids refer to the line number in Table 1. Source/Target information indicates if the feature is a 'source' one (S) or a 'target' one (T). The last line of Table 2 shows the MAE performance when all the 17 baseline features are used ('whole' line). The baseline system leads to a performance of 14.59. Interestingly, a feature alone leads to performance between 14.76 and 14.99. Thus, using only one feature allows to obtain good performance compared with using the whole set of features.

4.2 LSI based features

We use the dev corpus in order to estimate the number of topics for each LSI model leading to the best performance. For that, we test several values for the number of topics. We build one LSI model according to each of these values. Then,

S/T	id	MAE ($\times$100)	S/T	id	MAE ($\times$100)
S	9	14.99	T	17	14.95
T	6	14.99	S	16	14.94
T	2	14.98	S	15	14.94
S	7	14.98	S	13	14.93
T	5	14.97	S	3	14.91
S	1	14.97	S	12	14.82
S	4	14.96	S	8	14.80
S	10	14.96	S	14	14.76
S	11	14.95	whole		14.59

Table 2: MAE score on dev of each baseline feature, and of the whole 17 baseline features

we compute the LSI score of each (s, t) in tr. We add this score as a new feature to the 17 baseline. We apply the protocol of Section 2 in order to obtain the MAE score on the dev corpus. We show in Table 3 the results.

Nb Topics	LSI Training Corpus	
	tr	Europarl
10	14.55	14.54
20	14.55	14.54
30	14.52	14.57
40	14.52	14.59
50	14.51	14.58
60	14.50	14.58
70	14.49	14.57
80	**14.48**	14.56
90	14.49	14.55
100	14.49	14.56
150	14.50	14.53
200	14.50	14.50
250	14.51	14.50
300	14.51	14.50
350	14.50	14.49
400	14.52	14.49
500	14.52	**14.48**

Table 3: Performance in terms of MAE on dev of LSI feature according to the number of topics. The LSI feature is associated with the 17 baseline features.

The best performance are obtained for a number of topics equal to 80 for the tr corpus, and equal to 500 for the Europarl corpus. This is not surprising because Europarl corpus is strongly bigger than tr. Compared to baseline MAE (14.59), the LSI features leads to an improvement of 0.11 points.

4.3 Online systems based features

Table 4 shows the performance when online systems based features are used with the 17 baseline features. For each line, a 'X' indicates that the used features set includes the 9 features corresponding to the system of the column ($\mathbb{A}$, $\mathbb{B}$ or $\mathbb{C}$). The 'X' in column 'Inter' indicates that the features taking into account the three systems (formula 3) are used. The table shows that $\mathbb{B}$ is the most useful system, and that $\mathbb{C}$ is the less useful for prediction. Be careful that this does not give indication about the relative translation performance of online systems, but this indicates how the output quality of each system is correlated to the quality of the unknown system used by the organizers. The lack of usefulness of $\mathbb{C}$ for prediction is confirmed when the features from $\mathbb{A}$, $\mathbb{B}$ and $\mathbb{C}$ are combined. We obtain a better performance (13.93) when $\mathbb{C}$ is not used. Finally, adding the 'Inter' features does not lead to improvement. This may be because these features are correlated with '$\mathbb{A}$', '$\mathbb{B}$' and '$\mathbb{C}$': if a sentence is easy to translate, then, all systems should propose the same translation, this leads to high values for '$\mathbb{A}$', '$\mathbb{B}$' and '$\mathbb{C}$', and also for 'Inter'.

Baseline	$\mathbb{A}$	$\mathbb{B}$	$\mathbb{C}$	Inter	MAE ($\times$100)
X			X		14.38
X	X				14.28
X		X			14.02
X	X	X	X		13.95
X	X	X	X	X	13.95
X	X	X			13.93

Table 4: MAE Score on dev corpus of online systems based features.

4.4 Whole set of features and filtering

In this section, we use the whole set of features: baseline, LSI based, and online system based. For the LSI features, we use the LSI models leading to best performance (see Section 4.2): with 80 topics for the tr corpus, with 500 topics for the Europarl corpus. Table 5 shows the performance in terms of MAE. In this table, we present results when filtering is applied, and when it is not applied. We present several combinations. If we do not use filtering we obtain best performance when we do not use '$\mathbb{C}$' features (13.87, line 6). But if we use fil-

Features set	Baseline	LSI		online system based features				MAE ($\times$100)	
		tr 80	Europarl 500	$\mathbb{A}$	$\mathbb{B}$	$\mathbb{C}$	Inter	without filtering	with filtering
1	X		X	X	X	X		13.92	
2	X	X		X	X	X		13.91	
3	X	X	X	X	X	X		13.90	
4	X	X	X	X	X	X	X	13.90	13.70
5	X	X	X	X	X		X	13.88	
6	X	X	X	X	X			13.87	13.72

Table 5: Performance in terms of MAE on dev of the whole set of features

tering, it is is better to use the 50 features (13.90, line 4) and let the algorithm to automatically select the useful features: this leads to a performance of 13.70, better than 13.72 obtained by filtering the features set 6.

When we filter features set 4, we obtain 29 final features. 11 baseline features are kept (8 'S' and 3 'T'). Therefore, 'T' features are not numerous, but they are essential (3 are kept among 4). The LSI feature from tr is kept, but not the one from Europarl, maybe because the Europarl corpus is external to the Quality Estimation task. The selection of online systems based features confirms the relative usefulness of online systems $\mathbb{A}$, $\mathbb{B}$, and $\mathbb{C}$: only 2 '$\mathbb{C}$' features are kept, 4 '$\mathbb{A}$' features are kept, and 8 '$\mathbb{B}$' features are kept. Last, 3 'Inter' features among 4 are selected.

Finally, the baseline system (17 features) obtained a MAE score equal to 14.82 on the official test corpus. We submitted two systems, corresponding to line 4 in Table 5 (without and with filtering). The system without filtering led to a performance equal to 13.42 on the test corpus, and the same one after filtering led to a better performance equal to 13.34. Therefore, the results on the development corpus are confirmed by the test corpus.

5 Conclusion and perspectives

In this paper, we present our submission to the WMT2015 Quality Estimation Shared Task. Our system estimates quality at sentence level. In addition to the 17 baseline features, we use Latent Semantic Indexing based features which allow to measure the similarity between source and target sentences. Moreover we use pseudo-references from online machine translation systems, we extract n-gram statistics measuring the consensus between the target sentence and pseudo-references.

The features based on pseudo-references are more helpful for prediction than LSI based features. But there is a bias here, because we use only 2 LSI based features. We have now to extend the LSI approach. One first possibility is to use other ways to describe the latent semantic space, such as Latent Dirichlet Allocation (Blei et al., 2003). Second, the main drawback of LSI approach is that only lexical information is taken into account. One promising way is to include words sequence into the LSI model because Machine Translation is phrase based. We have yet tested this direction, but words sequences should be integrated carefully to obtain a tractable model.

References

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

C.-C. Chang and C.-J. Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

I. Guyon and A. Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research (Special Issue on Variable and Feature Selection)*, pages 1157–1182.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

D. Langlois, S. Raybaud, and Kamel Smaïli. 2012. Loria system for the WMT12 quality estimation shared task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 114–119.

M. L. Littman, S. T. Dumais, and T. K. Landauer. 1998. Automatic cross-language information retrieval using latent semantic indexing. In *Cross-language information retrieval*, pages 51–62. Springer.

N. Q. Luong, L. Besacier, and B. Lecouteux. 2014. Lig system for word level qe task at wmt14. In *Proceedings of the Ninth Workshop on Statistical Machine*

Translation, pages 335–341. Association for Computational Linguistics.

M. Saad, D. Langlois, and K. Smaïli. 2014. Cross-lingual semantic similarity measure for comparable articles. In *Advances in Natural Language Processing*, pages 105–115. Springer.

C. Scarton and L. Specia. 2014. Exploring consensus in machine translation for quality estimation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 342–347. Association for Computational Linguistics.

K. Shah, T. Cohn, and L. Specia. 2013. An investigation on the effectiveness of features for translation quality estimation. In *Proceedings of the Machine Translation Summit*, volume 14, pages 167–174.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

R. Soricut, N. Bach, and Z. Wang. 2012. The sdl language weaver systems in the wmt12 quality estimation shared task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 145–151. Association for Computational Linguistics.

L. Specia, K. Shah, J. GC De Souza, and T. Cohn. 2013. QuEst A translation quality estimation framework. In *ACL (Conference System Demonstrations)*, pages 79–84.

G. Wisniewski, N. Pécheux, A. Allauzen, and F. Yvon. 2014. Limsi submission for wmt'14 qe task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 348–354. Association for Computational Linguistics.