Towards Accurate Predictors of Word Quality for Machine Translation: Lessons Learned on French - English and English - Spanish Systems

Ngoc-Quang Luong, Laurent Besacier, Benjamin Lecouteux

Towards Accurate Predictors of Word Quality for Machine Translation: Lessons Learned on French - English and English - Spanish Systems

Ngoc-Quang Luong, Laurent Besacier, Benjamin Lecouteux

Laboratoire d'Informatique de Grenoble, Campus de Grenoble
41, Rue des Mathématiques, BP53, F-38041 Grenoble Cedex 9, France
{Ngoc-Quang.Luong,Laurent.Besacier,Benjamin.Lecouteux}@imag.fr
http://www.liglab.fr

Abstract

This paper proposes some ideas to build effective estimators, which predict the quality of words in a Machine Translation (MT) output. We propose a number of novel features of various types (system-based, lexical, syntactic and semantic) and then integrate them into the conventional (previously used) feature set, for our baseline classifier training. The classifiers are built over two different bilingual corpora: French - English (**fr-en**) and English - Spanish (**en-es**). After the experiments with all features, we deploy a "Feature Selection" strategy to filter the best performing ones. Then, a method that combines multiple "weak" classifiers to constitute a strong "composite" classifier by taking advantage of their complementarity allows us to achieve a significant improvement in term of F-score, for both **fr-en** and **en-es** systems. Finally, we exploit word confidence scores for improving the quality estimation system at sentence level.

Keywords: Machine Translation, Confidence Measure, Confidence Estimation, Conditional Random Fields, Boosting

1. Introduction

Statistical Machine Translation (SMT) systems in recent years have marked impressive breakthroughs with numerous fruitful achievements, as they produced more and more user-acceptable outputs. Nevertheless the users still face with some open questions: are these translations ready to be published as they are? Are they worth to be corrected or do they require retranslation? It is undoubtedly that building a method which is capable of pointing out the correct parts as well as detecting the translation errors in each MT hypothesis is crucial to tackle these above issues. If we limit the concept "parts" to "words", the problem is called Word-level Confidence Estimation (WCE). The WCE's objective is to judge each word in the MT hypothesis as correct or incorrect by tagging it with an appropriate label. A classifier which has been trained

beforehand calculates the confidence score for the MT output word, and then compares it with a pre-defined threshold. All words with scores that exceed this threshold are categorized in the *Good* label set; the rest belongs to the *Bad* label set.

The contributions of WCE for the other aspects of MT are incontestable. First, it assists the post-editors to quickly identify the translation errors, determine whether to correct the sentence or retranslate it from scratch, hence improve their productivity. Second, the confidence score of words is a potential clue to re-rank the SMT N-best lists. Last but not least, WCE can also be used by the translators in an interactive scenario [2].

This article conveys ideas towards a better word quality prediction, including: novel features integration, feature selection and Boosting technique. It also investigates the usefulness of using WCE in a sentence-level confidence estimation (SCE) system. After reviewing some related researches in Section 2, we depict all the features used for the classifier construction in Section 3. The settings and results of our preliminary experiments are reported in Section 4. Section 5 explains our feature selection procedure. Section 6 describes the Boosting method to improve the system performance. The role of WCE in SCE is discussed in Section 7. The last section concludes the paper and points out some perspectives.

2. Related Work

To cope with WCE, various approaches have been proposed, and most of them concentrate on two major issues: which types of features are efficient? And which classifier is the most suitable? In this review, we refer mainly to two general types of features: internal and external features. "*Internal features*" (or "*system-based features*") are extracted from the components of MT system itself, generated before or during the translation process (N-best lists, word graph, alignment table, language model, etc.). "*External features*" are constructed thanks to external linguistic knowledge sources and tools, such as Part-Of-Speech (POS) Tagger, syntactic parser, WordNet, stop word list, etc.

In [1], the authors combine a considerable number of features, then train them by the Neural Network and naive Bayes learning algorithms. Among these features, Word Posterior Probability (henceforth WPP) proposed by [3] is shown to be the most effective system-based features. Moreover, its combination with IBM-Model 1 features is also shown to overwhelm all the other ones, including heuristic and semantic features [4].

A novel approach introduced in [5] explicitly explores the phrase-based translation model for detecting word errors. A phrase is considered as a contiguous sequence of words and is extracted from the word-aligned bilingual training corpus. The confidence value of each target word is then computed by summing over all phrase pairs in which the target part contains this word. Experimental results indicate that the method yields an impressive reduction of the classification error rate compared to the state-of-the-art on the same language pairs.

Xiong et al. [6] integrate the POS of the target word with another lexical feature named "Null Dependency Link" and train them by Maximum Entropy model. In their results, linguistic features sharply outperform WPP feature in terms of F-score and classification error rate. Similarly, 70 linguistic features guided by three main aspects of translation: accuracy, fluency and coherence are applied in [9]. Results reveal that these features are helpful, but need to be carefully integrated to reach better performance.

Unlike most of previous work, the authors in [7] apply solely external features with the hope that their classifier can deal with various MT approaches, from statistical-based to rule-based. Given a MT output, the BLEU score is predicted by their regression model. Results show that their system maintains consistent performance across various language pairs.

Nguyen et al. [8] study a method to calculate the confidence score for both words and sentences relied on a feature-rich classifier. The novel features employed include source side information, alignment context, and dependency structure. Their integration helps to augment marginally in F-score as well as the Pearson correlation with human judgment. Moreover, their CE scores assist MT system to re-rank the N-best lists which improves the translation quality.

The authors in [9] strongly emphasize the "linguistic" factor when applying 70 features of this type, guided by three main aspects of translation: accuracy, fluency and coherence to investigate their usefulness. Unfortunately these features were not yet able to outperform shallower features based on statistics from the input text, its translation and additional corpora. Results reveal that linguistic features are still helpful, but need to be carefully integrated to reach better performance.

In the submitted system to the WMT12 shared task on Quality Estimation, the authors in [19] add some new features to the baseline provided by the organizers, including averaged, intra-lingual, basic parser and out-of-vocabulary features. They are then trained by SVM model, then filtered by forward-backward feature selection algorithm. This algorithm discards features linearly correlated with others while keeping those relevant for prediction. It increases slightly the performance of all-feature system in terms of Root Mean Square Error (RMSE).

Two recent workshops on MT (WMT 2013, WMT 2014) also witness several attempts of participants, especially on WCE shared task. In WMT 2013, while [27] employ the Conditional Random Fields (CRF) model [15] to train features, [26] present the global learning model by dynamic training with adaptive weight updates in the perceptron training algorithm. Concerning features, [26] present the "common cover links" (the links that point from the leaf node containing this word to other leaf nodes in the same subtree of the syntactic tree); while [27] focus on various n-gram combinations of target words. In WMT 2014, [25] exploit random forest classifier and built only 16 dense and continuous features of two categories: association features between source sentence and each target word, and fluency features describing the quality of the translation hypotheses. Meanwhile, confusion network , word lexicon and POS tags are the main resources to form the feature set of the systems of [24].

Comparing to these above approaches, our work is distinctive at these main

points: firstly, we inherit and propose various types of features: system-based features extracted from the MT system, together with lexical, syntactic and semantic features to verify if this combination improves the baselines performance. Secondly, the usefulness of all features is investigated in detail using a greedy feature selection algorithm. Thirdly, we propose a solution which exploits Boosting algorithm as a learning method in order to enhance the contribution of dominant feature subsets for the system, thus improve of the system's performance. Remarkably, the robustness of the proposed methods is challenged over two different language pairs: **fr-en** (French-English, our corpus) and **en-es** (English - Spanish, provided by WMT13[1] Quality Estimation Task at word level). Lastly, we investigate the WCE's role in enhancing the entire sentence's goodness prediction. In the next section, we describe in details the feature set used.

3. Features

This section depicts 26 types of features exploited to train the classifiers. Some of them are already described in our previous paper [18]. They also helped us to get the first rank in the *Quality Estimation Shared Task (word level) 2013* [22], and the third rank in that task in 2014 [23]. We categorize them into two classes: the **conventional features**, which are proven to work efficiently in numerous CE works and are inherited in our systems, and the **LIG features** which are more specifically suggested by us.

3.1. Conventional Features

3.1.1. Target Side Features

We take into account the information of every word (at position i in the MT output), including:

- The word itself.
- The sequences formed between it and a word before ($i-1/i$) or after it ($i/i+1$).
- The trigram sequences formed by it and two previous and two following words (including: $i-2/i-1/i$; $i-1/i/i+1$; and $i/i+1/i+2$).
- The number of occurrences in the sentence.

3.1.2. Source Side Features

Using the alignment information, we can track the source words which the target word is aligned to. To facilitate the alignment representation, we apply the BIO[2] format: if multiple target words are aligned with one source word, the first word's alignment information will be prefixed with symbol "B-" (means "Begin"); meanwhile "I-" (means "Inside") will be added at the beginning of

[1] http://www.statmt.org/wmt13/quality-estimation-task.html
[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

the alignment information for each of the remaining ones. The target words which are not aligned with any source word will be represented as "O" (means "Outside"). Table 1 shows an example of this representation, in case of the

Target words	**Source aligned words**	**Target words**	**Source aligned words**
The	B-le	to	B-de
public	B-public	look	B-tourner
will	B-aura	again	B-à|nouveau
soon	B-bientôt	at	B-son
have	I-aura	its	I-son
the	B-l'	attention	B-attention
opportunity	B-occasion	.	B-.

Table 1: Example of using BIO format to represent the alignment information

hypothesis is *"The public will soon have the opportunity to look again at its attention."*, given its source: *"Le public aura bientôt l'occasion de tourner à nouveau son attention."*. Since two target words *"will"* and *"have"* are aligned to *"aura"* in the source sentence, the alignment information for them will be "B-aura" and "I-aura" respectively. In case a target word has multiple aligned source words (such as *"again"*), we separate these words by the symbol "|" after putting the prefix "B-" at the beginning.

3.1.3. Alignment Context Features

These features are proposed by [8] in regard with the intuition that collocation is a believable indicator for judging if a target word is generated by a particular source word. We also apply them in our experiments, containing:

- Source alignment context features: the combinations of the target word and one word before (left source context) or after (right source context) the source word aligned to it.
- Target alignment context features: the combinations of the source word and each word in the window ± 2 (two before, two after) of the target word.

For instance, in case of *"opportunity"* in Table 1, the source alignment context features are: *"opportunity/l' "* and *"opportunity/de"*; while the target alignment context features are: *"occasion/have"*, *"occasion/the"*, *"occasion/opportunity"*, *"occasion/to"* and *"occasion/look"*.

3.1.4. Word Posterior Probability

WPP [3] is the likelihood of the word occurring in the target sentence, given the source sentence. Numerous knowledge sources have been proposed to calculate it, such as word graphs, N-best lists, statistical word etc. To calculate it, the key point is to determine sentences in N-best lists that contain the word e under consideration in a fixed position i. Let $p(f_1^J, e_1^I)$ be the joint probability of source sentence f_1^J and target sentence e_1^I. The WPP of e occurring in position i is computed by aggregating probabilities of all sentences containing e in this position:

$$p_i(e|f_1^J) = \frac{p_i(e, f_1^J)}{\sum_{e'} p_i(e', f_1^J)} \quad (1)$$

where

$$p_i(e, f_1^J) = \sum_{I, e_1^I} \Theta(e_i, e) \cdot p(f_1^J, e_1^I) \quad (2)$$

Here $\Theta(.,.)$ is the Kronecker function. The normalization in equation (1) is:

$$\sum_{e'} p_i(e', f_1^J) = \sum_{I, e_1^I} p(f_1^J, e_1^I) = p(f_1^J) \quad (3)$$

In this work, we exploit the graph that represents MT hypotheses [10]. From this, the WPP of word e in position i (denoted by WPP *exact*) can be calculated by summing up the probabilities of all paths containing an edge annotated with e in position i of the target sentence. Another form is "WPP *any*" in case we sum up the probabilities of all paths containing an edge annotated with e in any position of the target sentence. In this paper, both forms are employed.

3.1.5. Lexical Features

A prominent lexical feature that has been widely explored in WCE researches is word's Part-Of-Speech (POS). We use TreeTagger[3] toolkit for both English and Spanish POS annotation tasks and obtain the following features for each target word:

- Its POS
- Sequence of POS of all source words aligned to it
- Bigram and trigram sequences between its POS and the POS of previous and following words. Bigrams are POS_{i-1}/POS_i , POS_i/POS_{i+1} and trigrams are: $POS_{i-2}/POS_{i-1}/POS_i$; $POS_{i-1}/POS_i/POS_{i+1}$ and $POS_i/POS_{i+1}/POS_{i+2}$

In addition, we also build four other binary features that indicate whether the word is a: *stop word* (based on the stop word list for target language), *punctuation* symbol, *proper name* or *numerical*.

3.2. LIG Features

3.2.1. Graph topology features

They are based on the N-best list graph merged into a confusion network. On this network, each word in the hypothesis is labelled with its WPP, and belongs to one *confusion set*. Every completed path passing through all nodes in the network represents one sentence in the N-best, and must contain exactly one link from each confusion set. Looking into a confusion set (which the hypothesis word belongs to), we find some information that can be the useful indicators, including: the *number of alternative paths* it contains (called *Nodes*), and the distribution of posterior probabilities tracked over all its words (most interesting are *maximum and minimum probabilities*, called *Max* and *Min*). We assign three above numbers as features for the hypothesis word.

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

3.2.2. Language Model Based Features

Applying SRILM toolkit [11] on the bilingual corpus, we build 4-gram language models for both target and source side, which permit to compute two features: the *"longest target n-gram length"* and *"longest source n-gram length"* (length of the longest sequence created by the current word and its previous ones in the language model). For example, with the target word w_i: if the sequence $w_{i-2}w_{i-1}w_i$ appears in the target language model but the sequence $w_{i-3}w_{i-2}w_{i-1}w_i$ does not, the n-gram value for w_i will be 3. The value set for each word hence ranges from 0 to 4. Similarly, we compute the same value for the word aligned to w_i in the source language model. Additionally, we consider also the *backoff behaviour* [17] of the target language model to the word w_i, according to how many times it has to back-off in order to assign a probability to the word sequence.

3.2.3. Syntactic Features

The syntactic information about a word is a potential hint for predicting its correctness. If a word has grammatical relations with the others, it will be more likely to be correct than those which has no relation. In case of **the target language is English**, we select the Link Grammar Parser[4] as our syntactic parser (only for English), allowing us to build a syntactic structure for each sentence in which each pair of grammar-related words is connected by a labeled link. Based on this structure, we get a binary feature called *"Null Link"*: 0 in case of word has at least one link with the others, and 1 if otherwise. Another benefit yielded by this parser is the "constituent" tree, representing the sentence's grammatical structure (showing noun phrases, verb phrases, etc.). This tree helps to produce more word syntactic features, including *its constituent label* and *its depth in the tree* (or the distance between it and the tree root).

Figure 3.2.3 represents the syntactic structure as well as the constituent tree for a MT output: *"The government in Serbia has been trying to convince the West to defer the decision until by mid 2007."*. It is intuitive to observe

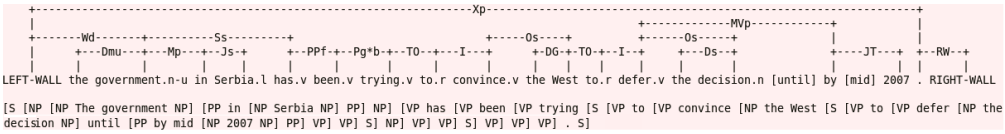

Figure 1: Example of parsing result generated by Link Grammar

that the words in brackets (including "*until*" and "*mid*") have no link with the others, meanwhile the remaining ones have. For instance, the word "*trying*" is connected with "*to*" by the link "TO" and with "*been*" by the link "Pg*b". Hence, the value of "Null Link" feature for "*mid*" is 1 and for "*trying*" is 0. The figure also brings us the constituent label and the distance to the root of

[4] http://www.link.cs.cmu.edu/link/

each word. In case of the word "*government*", these values are "NP" and "2", respectively.

For **Spanish** (as the target language), the Berkeley parser [20], which is trained over a Spanish treebank: AnCora[5], is employed, resulting in *"constituent label"* and *"depth in the tree"* values. Unfortunately, since the grammar link representation is not supported by this toolkit, we are not able to extract the *"Null Link"* feature for **en-es** WCE system.

3.2.4. Semantic Features

The word semantic characteristic that we study is its polysemy. We hope that the *number of senses* of each target word given its POS can be a reliable indicator for judging if it is the translation of a particular source word. For **English**, the feature *"Polysemy count"* is built by applying a Perl extension named Lingua::WordNet[6], which provides functions for manipulating the WordNet database. In case of **Spanish**, we employ BabelNet[7] - a multilingual semantic network that works similarly to WordNet but covers more European languages, including this language.

3.2.5. Pseudo Reference

This binary feature checks whether the target word appears in the output of another reliable MT system (which is called *pseudo reference*), given the same source sentence. Google Translate engine[8] was selected to generate this reference. The intuition behind is straightforward: if a target word occurs in multiple MT outputs, it would have higher possibility to become the correct translation. So far, this feature has been applied only for **en-es** WCE system.

It is worthy to mention again that the entire above feature set is used to train our **fr-en** (except *"Pseudo Reference"*) and **en-es** (except *"Null Link"*) WCE baselines. The crucial elements and settings for building them, along with their performance dealing with the test set are reported in the next section.

4. Baseline WCE Experiments

4.1. Experimental Settings

4.1.1. SMT System

Our **fr-en** SMT system is constructed using the Moses toolkit [12]. We keep the Moses's default setting: log-linear model with 14 weighted feature functions. The translation model is trained on the Europarl and News parallel corpora used for WMT10[9] evaluation campaign (1,638,440 sentences). Our target language model is a standard n-gram language model trained by the SRI language modeling toolkit [11] on the news monolingual corpus (48,653,884 sentences). Similarly, the **en-es** SMT system is also a phrase-based one (Moses),

[5] http://clic.ub.edu/corpus/en/ancora
[6] http://search.cpan.org/dist/Lingua-Wordnet/Wordnet.pm
[7] http://babelnet.org
[8] http://translate.google.fr
[9] http://www.statmt.org/wmt10/

trained on Europarl and News Commentaries corpora provided by WMT ([21]).

4.1.2. Corpus Preparation

Concerning the **fr-en** corpus, we used our SMT system to obtain the translation hypothesis for 10,881 source sentences taken from news corpora of the WMT evaluation campaign (from 2006 to 2010). Our post-editions were generated by using a crowdsourcing platform: Amazon Mechanical Turk [13]. We extract 10,000 triples (source, hypothesis and post edition) to form the training set, and keep the remaining 881 triples for the test set.

As stated in [21], the **en-es** corpus was formed by translating a dataset consisting of 22 English news articles into Spanish using WMT13 SMT system. Their post-editions were generated during CASMACAT[10] field trial. Of these, 15 documents have been processed by at least 2 out of 5 translators, therefore resulting a total of 1,087 triples (source, hypothesis, post-edition). Among them, the WMT13 organisers divided first 803 triples as training set, and keep the remaining 284 as test data.

4.1.3. Word Label Setting

For **fr-en** corpus, this task is performed by TERp-A toolkit [14]. Table 2 illustrates the labels generated by TERp-A for one hypothesis and reference pair. Each word or phrase in the hypothesis is aligned to a word or phrase in the reference with different types of edit: I (insertions), S (substitutions), T (stem matches), Y (synonym matches), and P (phrasal substitutions). The lack of a symbol indicates an exact match and will be replaced by E thereafter. We do not consider the words marked with D (deletions) since they appear only in the reference. Then, to train a binary classifier, we re-categorize the obtained 6-label set into binary set: The E, T and Y belong to the *Good* (G), whereas the S, P and I belong to the *Bad* (B) category. Finally, we observed that out of total words (train and test sets) are 85% labeled G, 15% labeled B.

The annotation for **en-es** corpus are derived automatically by computing WER between the MT hypothesis and its post-edited version. WMT13 supports two type of labels: multi-class (good, delete, substitute) and binary (good, bad). For the sake of consistency (with **fr-en**), in this work we employ only the binary version: *"G"* or *"B"*.

Reference	The	consequence	of	the	fundamentalist	movement		also	has	its importance	.
		S			S	Y	I		D	P	
Hyp. After Shift	The	result	of	the	hard-line	trend	is	also		important	.

Table 2: Example of training label obtained using TERp-A.

4.1.4. Classifier Selection

Motivated by the idea of addressing WCE as a sequence labeling task, we employ the *Conditional Random Fields* (CRF) model [15] as our Machine Learning (ML) method. The intuition behind this selection originates from the fact that the word quality cannot be correctly estimated if it is placed alone, without

[10] http://casmacat.eu

the relationship with others in the sentence. Whereas many other ML models predict quality for a single word without regard to "neighboring" ones, CRF can take context into account, which brings helpful information to the classifier and thus enable it perform more effectively. Among CRF based toolkits, we selected WAPITI [16] to train our classifier.

4.2. Preliminary Results and Analysis

In order to get an insight about the contribution of the *entire feature set*, as well as *our proposed features* (LIG features, when combined with conventional ones), we experiment with the following systems:

- **All features**: trained by the entire feature set.
- **Conventional features**: trained by all conventional features (without LIG features)
- **Baseline 1**: a *"naive"* classifier which always classifies words into *"G"* class.
- **Baseline 2**: assigns words randomly into G and into B label (random distribution in accordance with the percentage of these labels in the training corpus). For instance, in fr-en corpus, we observe 85% "G", 15% "B" labels, therefore this baseline will randomly classify 85% of words in the test set into "G" class, and the remaining part into "B" class.

We evaluate the performance of our classifiers by using three common evaluation metrics: Precision (Pr), Recall (Rc) and F-score (F). We perform the preliminary experiments by training a CRF classifier with the combination of all 25 features. The classification task is then conducted multiple times, corresponding to a threshold increase from 0.300 to 0.975 (step = 0.025). When threshold = α, all words in the test set which the probability of G class exceeds α will be labelled as "G", and otherwise, "B". The values of Pr and Rc of G and B label are tracked along this threshold variation, and then are averaged and shown in Table 3, for "all-feature" and baseline systems.

It can be seen from the results that *"G"* label is much better predicted than *"B"* label in all systems and naive baselines of both language pairs (for instance, 87.07 F-score vs 37.76 in **fr-en** and 84.16 vs 51.36 in **en-es** all-feature systems). In addition, the combination of all features helped to detect the translation errors significantly above the "naive" baselines. Using them, **fr-en** system obtains 37.76% instead of nothing (Baseline 1) or 16.26% (Baseline 2). The very significant improvement can be also seen in **en-es** system. More interestingly, the LIG features show clearly their contributions when combined with conventional ones: they help to gain more 0.6 and 0.95 F score for "G" and "B" label in **fr-en** system, respectively; while these gains in **en-es** system are 0.85 and 1.33 point. The usefulness of them will be further analyzed in the next section.

	System	Label	Pr(%)	Rc(%)	F(%)
fr-en	All features	Good	85.99	88.18	87.07
		Bad	40.48	35.39	37.76
	Conventional features	Good	85.82	87.14	86.47
		Bad	39.96	34.13	36.81
	Baseline 1	Good	81.78	100.00	89.98
		Bad	-	0	-
	Baseline 2	Good	81.77	85.20	83.45
		Bad	18.14	14.73	16.26
en-es	All features	Good	84.48	83.86	84.16
		Bad	50.02	52.78	51.36
	Conventional features	Good	84.01	82.63	83.31
		Bad	49.12	50.98	50.03
	Baseline 1	Good	74.24	100.00	85.21
		Bad	-	0	-
	Baseline 2	Good	76.21	73.97	75.07
		Bad	26.35	27.50	26.91

Table 3: Average Pr, Rc and F for labels of all-feature, basic-feature systems and two baselines.

5. Feature Selection for WCE

In Section 4, the all-feature **fr-en** and **en-es** systems yielded promising F scores for G label, but not very convincing F scores for B label. That can be originated from the risk that not all of features are really useful, or in other words, some are poor predictors and might be the obstacles weakening the other ones. In order to prevent this drawback, we propose a method to filter the best features based on the "Sequential Backward Selection" algorithm[11]. We start from the full set of N features, and in each step sequentially remove the most useless one. To do that, all subsets of (N-1) features are considered and the subset that leads to the best performance gives us the weakest feature (not included in the considered set). Obviously, the discarded feature is not considered in the following steps. We iterate the process until there is only one remaining feature in the set, and use the following score for comparing systems: $F_{avg}(all) = \beta \cdot F_{avg}(G) + \gamma \cdot F_{avg}(B)$, where $F_{avg}(G)$ and $F_{avg}(B)$ are the averaged F scores for G and B label, respectively, when threshold varies from 0.300 to 0.975. The coefficients β and γ are determined with respect to the balance between G and B labels in the corpus: the minor class will be put more weight than the major one, since it is harder to predict. In **fr-en** system, the value (β, γ) is $(0.3, 0.7)$, and that in **en-es** system is $(0.5, 0.5)$. This strategy enables us to sort the features in descending order of importance, as displayed in Table 4. Figure 2 shows the evolution of the WCE performance as more and more features are removed, and the details of 3 best feature subsets yielding the highest $F_{avg}(all)$.

Table 4 reveals informative features (eight bold ones) performing well in both

[11] http://research.cs.tamu.edu/prism/lectures/pr/pr_l11.pdf

Rank	fr-en	en-es
1	**Source POS**	**Source POS**
2	Source word	Occur in Google Translate
3	**Target word**	**Nodes**
4	Backoff behaviour	**Target POS**
5	**WPP** *any*	**WPP** *any*
6	**Target POS**	**Left source context**
7	Constituent label	**Right target context**
8	**Left source context**	Numeric
9	Null link	Polysemy count(target)
10	**Stop word**	Punctuation
11	Max	**Stop word**
12	**Right target context**	Right source context
13	**Nodes**	**Target word**
14	Punctuation	Distance to root
15	Polysemy count	Backoff behaviour
16	Longest source gram length	Constituent label
17	Number of occurrences	Proper name
18	Numeric	Number of occurrences
19	Proper name	Min
20	Left target context	Max
21	Min	Left target context
22	Longest target gram length	Polysemy count (source)
23	Right source context	Longest target gram length
24	Distance to Root	Longest source gram length
25	WPP *exact*	Source Word

Table 4: The rank of each feature (in term of usefulness) in **fr-en** and **en-es** systems. The bold ones perform well in both cases.

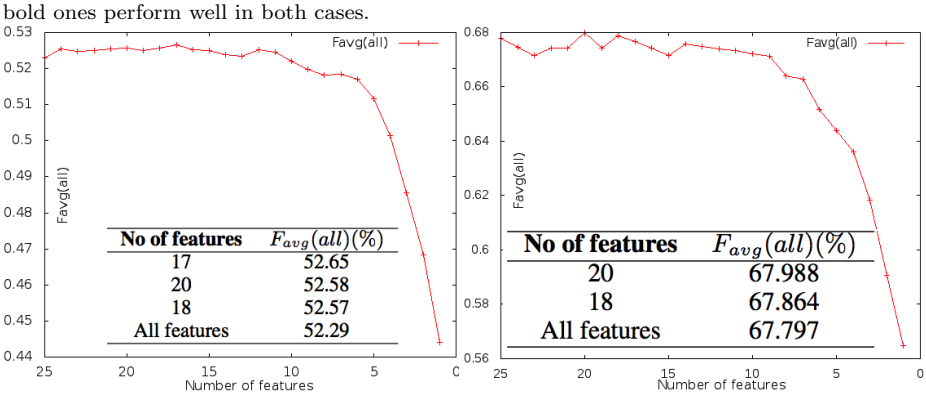

Figure 2: Evolution of system performance ($F_{avg}(all)$) during Feature Selection process on **fr-en** (left) and **en-es** (right) system

systems, as they constitute the "upper part" (top 13) of each one. Among them, the feature *"Source POS"* is the most outstanding when it holds the first rank in both cases, saying that the information from source side is valuable. On the contrary, three features including *Left target context*, *Min* and *Longest target n-gram length* are proven weak with their bottom-most positions in the two sets. Concerning **our proposed features**, we divide the further analysis into each particular case: in **en-es** system, the first-time-experimented feature "Occur in Google Translate" is the most prominent (rank 2), implying that such an online MT system can be a reliable reference channel for predicting word quality. Besides, *"Nodes"* and *"Polysemy count (target)"* are shown beneficial when dealing with Spanish data set. Meanwhile, the syntactic features (*"Constituent Label"*, *"Null Link"*) outperform the others in **fr-en** system, followed by *"Nodes"*

and *"Max"*. In addition, in Figure 2, when the size of feature set is small (from 1 to 8), we can observe sharply the growth of the system performance ($F_{avg}(all)$). Nevertheless the scores seem to saturate as the feature set increases from 9 up to 25, in both systems. This phenomenon raises a hypothesis about our classifier's learning capability when coping with a large number of features, hence drives us to an idea for improving the classification scores, which is detailed in the next section.

6. Classifier Performance Improvement Using Boosting

As stated before, the best performance did not come from the "all-feature" system, but from the system trained by a subset of features (17 in **fr-en** and 20 in **en-es** systems). Besides this, we could not find any considerable progression in F-score when the feature set is lengthened from 9 to 25. These observations lead to a question: if we build a number of "weak" (or "basic") classifiers using subsets of our features and combine predictions from them in an appropriate way, can we obtain a stronger classifier than each individual? In this work, the *"appropriate way"* that we propose is as follows: we consider the prediction coming from each "weak" CRF classifiers (mentioned above) as a new feature, and apply the Boosting method on the new feature set (called Boosting features to distinguish with the conventional and LIG features) in order to get a stronger classifier. When applying Boosting algorithm, our hope is that multiple models can complement each other as one model might be specialized in a part of the data where the others do not perform very well.

The process to build features for Boosting system is depicted as follows. First, we prepare 23 feature subsets ($F_1, F_2, ..., F_{23}$) to train 23 basic classifiers, in which: F_1 contains all features, F_2 is the best performing set after selection (top 17 in **fr-en** and and top 20 in **en-es** system). F_i ($i = \overline{3..23}$) contains 9 randomly chosen features. Next, the 10-fold cross validation is applied on the training set. We divide it into 10 equal subsets (S_1, S_2, ..., S_{10}). In the loop i ($i = \overline{1..10}$), S_i is used as the test set and the remaining data is trained with 23 feature subsets. After each loop, we obtain the results from 23 classifiers for each word in S_i. Finally, the concatenation of these results after 10 loops gives us the training data for Boosting. The detail of this algorithm is described as below:

Algorithm to build Boosting training data

for i :=1 **to** 10 **do**
begin
  TrainSet(i) := $\cup S_k$ ($k = \overline{1..10}, k \neq i$)
  TestSet(i) := S_i
  for j := 1 **to** 23 **do**
  begin
    Classifier C_j := Train *TrainSet(i)* with F_j
    Result R_j := Use C_j to test S_i
    Column P_j := Extract the *"probability of word to be G label"* in R_j
  end
  Subset D_i (23 columns) := $\{P_j\}$ ($j = \overline{1..23}$)

end
Boosting training set $D := \cup D_i$ $(i = \overline{1..10})$

Next, the Bonzaiboost toolkit[12] (which bases on decision trees and implements Boosting algorithm) is used for building Boosting model. In the training command, we invoked: algorithm = "AdaBoost", and number of iterations = 300. In both systems, the Boosting test set is prepared as follows: we train 23 feature sets with the training set to obtain 23 classifiers, then use them to test the CRF test set, finally extract the 23 probability columns (like in the above pseudo code). In the testing phase, similar to what we did in Section 5, the *averaged* Pr, Rc and F scores against threshold variation for G and B labels are tracked as seen in Table 5. The scores suggest that using Boosting algorithm on our

System		Pr(G)	Rc(G)	F(G)	Pr(B)	Rc(B)	F(B)
fr-en	Boosting	90.10	84.13	**87.02**	34.33	49.83	**40.65**
	all-feature	85.99	88.18	**87.07**	40.48	35.39	**37.76**
en-es	Boosting	85.35	83.15	**84.24**	50.99	55.98	**53.36**
	all-feature	84.48	83.86	**84.16**	50.02	52.78	**51.36**

Table 5: Comparison of the average Pr, Rc and F between CRF and Boosting systems

CRF classifiers' output is an efficient way to make them predict better. With **fr-en** system, on the one side, we maintain the already good achievement on G class (only 0.05% lost); on the other side we augment 2.89% the performance in B class. More remarkably, on **en-es** system, the improvements are obtained in both labels: 0.08% and 2.00% for G and B, respectively. It is likely that Boosting enables different models to better complement one another, in terms of the later model becomes experts for instances handled wrongly by the previous ones. Another advantage is that Boosting algorithm weights each model by its performance (rather than treating them equally), so the strong models (come from all features, best performing set after selection, etc.) can make more dominant impacts than the others.

It is worthy to mention that, our **en-es** optimized system participated in WMT 2013 shared task for word-level quality estimation, and got the first rank among submitted systems [21], when their performances were measured by the averaged F score of "G" and "B" label (macro F score). We obtained macro F score of 0.65, whereas that of the other participants are: 0.59 (CNGL GLM), 0.55 (UMAC NB), 0.55 (CNGL GLMd), 0.45 (UMAC CRF) and that of the baseline is 0.42. These commendable results show the effective contribution of our feature set and the proposed optimization methods.

7. Using WCE in Sentence Confidence Estimation (SCE)

WCE helps not only in detecting translation errors, but also in improving the sentence level prediction when combined with other sentence features. To

[12] http://bonzaiboost.gforge.inria.fr/x1-20001

verify this, firstly we build a SCE system (called **SYS1**) based on our WCE outputs (prediction labels). The seven features used to train **SYS1** are:

- The ratio of number of good words to total number of words. (1 feature)
- The ratio of number of good nouns to total number of nouns. The similar ones are also computed for other POS: verb, adjective and adverb. (4 features)
- The ratio of number of n consecutive good word sequences to total number of consecutive word sequences. Here, n=2 and n=3 are applied. (2 features)

Then, we inherit the script used in WMT12[13] for extracting 17 sentence features, to build an another SCE system (**SYS2**). In both **SYS1** and **SYS2**, each sentence training label is an integer score from 1 to 5, based on its TER score, as following:

$$score(s) = \begin{cases} 5 & if\ \text{TER(s)} \leq 0.1 \\ 4 & if\ 0.1 < \text{TER(s)} \leq 0.3 \\ 3 & if\ 0.3 < \text{TER(s)} \leq 0.5 \\ 2 & if\ 0.5 < \text{TER(s)} \leq 0.7 \\ 1 & if\ \text{TER(s)} > 0.7 \end{cases} \quad (4)$$

Two conventional metrics are used to measure the SCE system's performance: Mean Absolute Error (MAE) and Root of Mean Square Error (RMSE)[14]. Given a test set $S = s_1, s_2, ..., s_{|S|}$, let $R(s_i)$ and $H(s_i)$ be the reference score (determined by TERPA) and hypothesis score (by our SCE system) for sentence s_i respectively. Then, MAE and RMSE can be formally defined by:

$$MAE = \frac{\sum_{i=1}^{|S|} |R(s_i) - H(s_i)|}{|S|} \quad (5)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{|S|} (|R(s_i) - H(s_i)|)^2}{|S|}} \quad (6)$$

To observe the impact of WCE on SCE, we design a third system (called **SYS1+SYS2**), which takes the results yielded by **SYS1** and **SYS2**, post-processes them and makes the final decision. For each sentence, **SYS1** and **SYS2** generate five probabilities for five integer labels it can be assigned, we then select the label with highest probability as the official result. Meanwhile, **SYS1+SYS2** collects probabilities coming from both systems, then the probability for each label is the sum of two appropriate values in **SYS1** and **SYS2**. Similarly, the label with highest likelihood is assigned to this sentence. The experimental results are shown in Table 6.

Scores observed reveal that when WMT12 baseline features and those based

[13] https://github.com/lspecia/QualityEstimation/blob/master/baseline_system

[14] http://www.52nlp.com/mean-absolute-error-mae-and-mean-square-error-mse/

System		MAE	RMSE
fr-en	SYS1	0.5584	0.9065
	SYS2	0.5198	0.8707
	SYS1+SYS2	**0.4835**	**0.8415**
en-es	SYS1	0.7056	1.0339
	SYS2	0.6035	0.9121
	SYS1+SYS2	**0.5628**	**0.8876**

Table 6: Scores of 3 different SCE systems.

on our WCE are separately exploited, they yield acceptable performance. More interesting, the contribution of WCE is definitively proven when it is combined with a SCE system: the combination system **SYS1+SYS2** sharply reduces MAE and RMSE of both single systems. It demonstrates that in order to judge effectively a sentence, besides global and general indicators, the information synthesized from the quality of each word is also very useful.

8. Conclusions and Perspectives

We proposed some ideas to deal with WCE for MT, starting with the integration of our proposed features into the existing features to build the classifier. The first experiment's results show that precision and recall obtained in G label are very promising and much more convincing than those of B label. A feature selection strategy is then deployed to identify the valuable features, find out the best performing subset. One more contribution we made is the protocol of applying Boosting algorithm, training multiple "weak" classifiers, taking advantage of their complementarity to get a "stronger" one. Especially, the integration with SCE enlightens the WCE contribution in judging the sentence quality. These above propositions are shown robust as they deal well with two different languages pairs: **fr-en** and **en-es**.

In the future, we will take a deeper look into linguistic features of word, such as the grammar checker, dependency tree, semantic similarity, etc. Besides, we would like to investigate the segment-level confidence estimation, which exploits the context relation between surrounding words to make the prediction more accurate. Moreover, a methodology to conclude the sentence confidence relied on the word- and segment- level confidence will be also deeply considered.

References

[1] Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis A., Ueffing, N.: Confidence Estimation for Machine Translation. Technical report, JHU/CLSP Summer Workshop (2003)

[2] Gandrabur, S., Foster, G.: Confidence Estimation for Text Prediction. In: Conference on Natural Language Learning (CoNLL), pp. 315-321, Edmonton, May (2003)

[3] Ueffing, N., Macherey, K., Ney, H.: Confidence Measures for Statistical Machine Translation. In: MT Summit IX, pp. 394-401, New Orleans, LA, September (2003)

[4] Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., Ueffing, N.: Confidence Estimation for Machine Translation. In Proceedings of COLING 2004, pp. 315321, Geneva, April (2004)

[5] Ueffing, N., Ney, H.: Word-level Confidence Estimation for Machine Translation Using Phrased-based Translation Models. In: Human Language Technology Conference and Conference on Empirical Methods in NLP, pp. 763-770, Vancouver (2005)

[6] Xiong, D., Zhang, M., Li, H.: Error Detection for Statistical Machine Translation Using Linguistic Features. In: 48th ACL, pp. 604-611, Uppsala, Sweden, July (2010)

[7] Soricut, R., Echihabi, A.: Trustrank: Inducing Trust in Automatic Translations via Ranking. In: 48th ACL (Association for Computational Linguistics), pp. 612-621, Uppsala, Sweden, July (2010)

[8] Nguyen, B., Huang, F., Al-Onaizan, Y.: Goodness: A Method for Measuring Machine Translation Confidence. In: 49th ACL, pp. 211-219, Portland, Oregon, June (2011)

[9] Felice, M., Specia, L.: Linguistic Features for Quality Estimation. In: 7th Workshop on Statistical Machine Translation, pp. 96-103, Montreal, Canada, June 7-8 (2012)

[10] Ueffing, N., Och, F.J., Ney, H.: Generation of Word Graphs in Statistical Machine Translation. In: Conference on Empirical Methods for Natural Language Processing (EMNLP 02), pp. 156-163, Philadelphia, PA (2002)

[11] Stolcke, A.: Srilm - an Extensible Language Modeling Toolkit. In: 7th International Conference on Spoken Language Processing, pp. 901-904, Denver, USA (2002)

[12] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: 45th Annual Meeting of the Association for Computational Linguistics, pp. 177-180, Prague, Czech Re , June (2007)

[13] Potet, M., Rodier, E.E., Besacier, L., Blanchon, H.: Collection of a Large Database of French-English SMT Output Corrections. In: 8th International Conference on Language Resources and Evaluation, Istanbul (Turkey), 23-25 May (2012)

[14] Snover, M., Madnani, N., Dorr, B., Schwartz, R.: Terp System Description. In: MetricsMATR workshop at AMTA (2008)

[15] Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: CML-01, pp. 282-289 (2001)

[16] Lavergne, T., Cappé, O., Yvon, F.: Practical Very Large Scale CRFs. In: 48th Annual Meeting of the Association for Computational Linguistics, pp. 504-513 (2010)

[17] Raybaud, S., Langlois, D., Smaïli, K.: This sentence is wrong. Detecting errors in machine - translated sentences. Machine Translation, 25(1):1-34 (2011).

[18] Luong, N.Q.: Integrating Lexical, Syntactic and System-based Features to Improve Word Confidence Estimation in SMT. In: JEP-TALN-RECITAL, pp. 43-56, Grenoble, France, June 4-8 (2012).

[19] Langlois, D., Raybaud, S., Smaïli, K.: LORIA System for the WMT12 Quality Estimation Shared Task. In Proceedings of the 7th Workshop on Statistical Machine Translation, pages 114119, Montreal, Canada, June 7-8, 2012.

[20] Petrov, S., Klein, D.: Improved inference for unlexicalized parsing. In Proceedings of NAACL HLT 2007, pages 404411, Rochester, NY, April 2007

[21] Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., Specia, L.: Findings of the 2013 Workshop on Statistical Machine Translation. In Proceedings of the Eighth Workshop on Statistical Machine Translation, pp. 1-44, Sofia, Bulgaria (2013)

[22] Luong, N.Q., Lecouteux, B., Besacier, L.: LIG System for WMT13 QE Task: Investigating the Usefulness of Features inWord Confidence Estimation for MT. In Proceedings of the Eighth Workshop on Statistical Machine Translation, Sofia, Bulgaria (2013)

[23] Luong, N. Q., Besacier, L., and Lecouteux, B. (2014). LIG System for Word level WE Task at WMT14. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA. Association for Computational Linguistics.

[24] Camargo-de-Souza,J.G.,González-Rubio,J., Buck, C., Turchi, M., and Negri, M. FBK-UPV-UEdin Participation in the WMT14 Quality Estimation Shared-Task. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA. Association for Computational Linguistics. (2014).

[25] Wisniewski, G., Pécheux, N., Allauzen, A., and Yvon, F. (2014). LIMSI Submission for WMT14 QE Task. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, Maryland, USA. Association for Computational Linguistics.

[26] Ergun Bicici. Referential Translation Machines for Quality Estimation. In Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 343351, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb. org/anthology/W13-2242.

[27] Aaron Li-Feng Han, Yi Lu, Derek F. Wong, Lidia S. Chao, Liangye He, and Junwen Xing. Quality Estimation for Machine Translation Using the Joint Method of Evaluation Criteria and Statistical Modeling. In Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 365372, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb. org/anthology/W13-2245.