# Efficient Combination of Confidence Measures for Machine Translation

*Sylvain Raybaud, David Langlois, Kamel Smaïli*

PAROLE team, LORIA
Campus Scientifique BP 239
54506 Vandoeuvre-lès-Nancy FRANCE
{sylvain.raybaud,david.langlois,kamel.smaili}@loria.fr

10.21437/Interspeech.2009-154

## Abstract

We present in this paper a twofold contribution to Confidence Measures for Machine Translation. First, in order to train and test confidence measures, we present a method to automatically build corpora containing realistic errors. Errors introduced into reference translation simulate classical machine translation errors (word deletion and word substitution), and are supervised by Wordnet. Second, we use SVM to combine original and classical confidence measures both at word- and sentence-level. We show that the obtained combination outperforms by 14% (absolute) our best single word-level confidence measure, and that combination of sentence-level confidence measures produces meaningful scores.

**Index Terms**: statistical machine translation systems, confidence measures, support vector machine, support vector regression

## 1. Introduction

Statistical methods for machine translation suffer from an intrinsic drawback: they only produce the most likely result given training and input data. It is easy to see that this will sometimes not be optimal with regard to human expectations. It is therefore important to be able to automatically evaluate the quality of the result, even when no reference translation is available: this can be handled by different *confidence measures (CMs)* which have been proposed for machine translation.

Confidence measures are designed to discriminate between correct and erroneous words and sentences in automatic translations. Beside manual correction of erroneous words we can imagine several applications of confidence estimation: pruning or re-ranking the n-best list, generating new hypothesis by recombining parts of different candidates having high scores, or discriminative training by tuning the parameters to optimise the separation between sentences with a high confidence score and those with a low one.

This paper extends and improves the work presented in [1, 2] in two directions: first we propose to automatically generate a training corpus for confidence measures, obtained by introducing realistic errors in a bilingual aligned corpus (section 3.2). Thanks to this technique we avoid the time and money consuming task of having a human translator classifying words and sentences as “correct” or “incorrect”; second, we use Support Vector Machines to combine different predictive parameters and generate sentence level confidence scores by using support vector regression.

## 2. Introduction to Confidence Measures

The role of a confidence measure in Machine Translation is to decide wether a word $t_j$ at position $j$ in the target sentence $\mathbf{t}_1^J = t_1,..,t_J$ is correct or not. Such a confidence measure should take into account the word, its position, the whole target sentence and the source $\mathbf{s}_1^I = s_1,..,s_I$. Let $correct_{j,t_j,\mathbf{t},\mathbf{s}}$ be a random variable whose value is 1 if the aforementioned word is indeed correct, 0 if it is not. For a word $t_j$, a theoretical confidence measure $C$ is:

$$C(j,t_j,\mathbf{t},\mathbf{s}) = P(correct_{j,t_j,\mathbf{t},\mathbf{s}} = 1|j,t_j,\mathbf{t},\mathbf{s}) \quad (1)$$

The decision can also be taken at sentence level: an ideal sentence-level estimator would be:

$$C(\mathbf{s},\mathbf{t}) = P(correct_{\mathbf{s},\mathbf{t}} = 1|\mathbf{t},\mathbf{s}) \quad (2)$$

It is difficult (if not impossible) to directly estimate such probabilities. Existing methods therefore either compute score which are supposed to monotonically depend on them, or - like we do - replace $(j,t_j,\mathbf{t},\mathbf{s})$ with so called *predictive parameters* (a vector $\mathbf{X}(\mathbf{t},\mathbf{s},j) \in \mathbb{R}^d$ of numerical features; a feature can be for example a word posterior probability) and approximate these probabilities by the more standard:

$$P(correct_{j,t_j,\mathbf{t},\mathbf{s}} = 1|\mathbf{X}(\mathbf{t},\mathbf{s},j))$$

which can be estimated by classical machine learning techniques. Classical predictive parameters are, for example, word posterior probabilities computed on the word lattice or given by a translation table [3, 4, 5], n-gram probabilities, and other well known features. In this article we use predictive parameters relying on mutual information [1, 2], and n-gram and backward n-gram language models [6].

**Sentence level confidence measures:** it is not always possible, even for a human translator, to discriminate between correct and incorrect translations, especially when dealing with whole sentences instead of words. It is generally preferred to assign a numerical score reflecting the quality of the sentence. Therefore we want to train confidence measures to estimate such score, hoping that it will correlate well with human judgement. To this end we compute sentence-level predictive parameters and perform support vector regression [7] to train our confidence measure to mimic the BLEU score [8], but without a reference translation (section 6).

## 3. Software and Material Description

Experiments are run using a French to English phrase-based translation system. A system is trained corresponding to the baseline described in the *ACL workshop on statistical machine translation* [9]. It uses an IBM-5 model [10] and has been trained on the EUROPARL corpus (proceedings of the European Parliament, [9]) using GIZA++ [11] and the SRILM toolkit [12]. The decoding process is handled by Moses [13].

### 3.1. Corpus for translation

The only difference between the previously mentioned baseline and our system is the size of the bilingual training corpus: instead of using the whole corpus (around 1M pairs of sentences) to train the system, we used only 500K pairs. We used 100K of the remaining pairs to train confidence measures (section 3.2), and the remaining 400K are kept for further work (see conclusion). We summarise in Table 1 the sizes of the different parts of the corpus.

| | Sentences pairs | Running words | |
|---|---|---|---|
| Set | | French | English |
| Training | 500K | 11M | 10M |
| Development | 2K | 55K | 50K |
| Test | 2K | 63K | 58K |

Table 1: Corpora sizes

The BLEU score achieved by this baseline is 31.9.

### 3.2. Generation of training and evaluation data for confidence measures

Confidence measures (CMs) must be trained to discriminate between correct and incorrect words and assess the quality of translated sentences. Therefore many measures need training examples. Ideally a human professional translator should read the output of a MT system and assign a label (*correct* or *incorrect*) to each word and a numeric evaluation to each sentence. This is a very tedious and time consuming task (classifying a thousand words takes around two hours, and hundreds of thousands are needed). The result depends also very much on the individual. Therefore a semi-automatic method for efficiently classifying words and evaluating sentences is needed. We discuss below two ways to achieve that:

**Comparing the candidate to references:** an intuitive idea would be to compare a generated translation to a reference translation, and classify as correct the candidate words that are levenshtein-aligned to a word in the reference translation [4]. However this is too harsh and many correct words would be incorrectly classified. This problem can be partly overcome by using multiple reference translations [14]. However multiple references are not always available.

**Automatic generation of incorrect examples:** starting from human-made reference translations, errors are automatically introduced in order to generate training examples for CMs. Given an English sentence **t** (which is a correct translation of source sentence **s**) we randomly introduce errors of four types: **swap**, **deletion**, **substitution** and **grammatical errors**. Swaps and deletions are straightforward: words are picked up at random and moved or deleted. Grammatical errors are generated by modifying the ending of randomly selected words ("preserving" may become "preserved", "environment" may become "environmental"). For substitution, we first use giza++ to align the words in source and target sentences. Then, a target word $t$ aligned to a source word $s$ can be replaced by a word $t'$ picked at random such that $t'$ is a possible translation of $s$ (we use a translation table also generated by giza++). WordNet [15] is used to check that $t'$ is not an exact synonym of $t$ (otherwise it would not be an incorrect word). Below is an example of such a degraded translation.

source sentence:
```
   Quant à eux, les instruments politiques doivent
   s'adapter à ces objectifs.
```

reference translation:
```
   Policy instruments, for their part, need to
   adapt to these goals.
```

degraded translation:
```
   Policy instruments, for the part, must to adapt
   to these goal.
```

A word in the degraded translation is labelled as correct if and only if it is levenshtein-aligned to a word in the reference translation. A degraded translation's reference score is its BLEU score, computed on the individual sentence.

We degraded 100,000 translations (section 3.1). In order to keep the computation time tractable, only 30,000 of these have been used to train the SVM (10K for training the SVM, 10K for optimising parameters and 10K for testing). The BLEU score of the degraded corpus is 47. This BLEU score is much higher that our baseline's (31.9), but the baseline's quality is underestimated because only one reference translation is available; moreover the degraded corpus comes directly from the reference sentences, while the baseline is the output of a MT system basing on the source sentences.

## 4. Evaluation of the confidence measures

**Word-level confidence measures**: we evaluate the usefulness of a confidence measure by its discriminative power. Given a confidence measure $C$ a classifier $\mathcal{C}_{C,\delta}$ is built such that:

$$\mathcal{C}_{C,\delta}(t) = \begin{cases} \text{accept} & \text{iff} \quad C(t) \geq \delta \\ \text{reject} & \text{iff} \quad C(t) < \delta \end{cases}$$

$\mathcal{C}_{C,\delta}(t)$ is then compared to the reference class of $t$ for each $t$ in the training corpus, and we compute three metrics:

- Correct Acceptance Rate ($CAR(\delta)$ or Sensitivity) is the ratio of correct words retrieved:
$$\frac{\text{number of correctly accepted words}}{\text{total number of correct words}}$$
- Correct Rejection Rate ($CRR(\delta)$ or Specificity) is the ratio of incorrect words retrieved:
$$\frac{\text{number of correctly rejected words}}{\text{total number of incorrect words}}$$
- F-measure is the harmonic mean of CAR and CRR:
$$F(\delta) = \frac{2 \times CAR(\delta) \times CRR(\delta)}{CAR(\delta) + CRR(\delta)}$$

These metrics are common in machine learning. Basically a relaxed classifier has a high CAR (most correct words are labelled as such) and low CRR (many incorrect words are not detected), while a harsh one has a high CRR (an erroneous word is often detected) and a low CAR (many correct words are rejected).

The plot of $CAR(\delta)$ against $CRR(\delta)$ is called the ***ROC curve*** (*Receiver Operating Characteristic*). As $\delta$ increases, CAR decreases and CRR increases. The ROC curve of a perfect classifier would go through the point (1,1), while that of the most naive classifier (based on random scores) is the segment joining (0,1) and (1,0). The ROC curve can therefore be used to quickly visualise the quality of the classifier: the higher above this segment a curve is, the better. For every classifier there exists an optimal threshold $\delta^*$ which maximises the F-measure: we always used this best F-measure $F(\delta^*)$ as the optimisation criterion for word-level confidence measures.

**Sentence-level confidence measures:** in this case sentences are not classified as correct or incorrect but are only assigned a numeric quality estimation. As said before, our training and test corpora contain source sentences, target sentences and their BLEU score:

$$\text{training sample: } \mathbf{s}^n, \mathbf{t}^n, BLEU(\mathbf{t}^n, \mathbf{reference}^n)$$

each sentence is replaced by a vector of predictive parameters:

$$\text{training sample: } \mathbf{X}(\mathbf{t}^n, \mathbf{s}^n), BLEU(\mathbf{t}^n, \mathbf{reference}^n)$$

Support vector machines are then trained to perform regression on the training samples, that is, to produce a score as close as possible to the given BLEU score, using only the vector **X**:

$$\text{output sample: } score(\mathbf{X}(\mathbf{t}^n, \mathbf{s}^n)), BLEU(\mathbf{t}^n, \mathbf{reference}^n)$$

We then use mean square error on the test corpus as an evaluation of the confidence measure:

$$MSE = \frac{1}{\sharp TEST} \sum_{(\mathbf{s},\mathbf{t},\mathbf{ref}) \in TEST} (score(\mathbf{X}(\mathbf{t},\mathbf{s})) - BLEU(\mathbf{t},\mathbf{ref}))^2$$

## 5. Fusion of Confidence Measures

In [1, 2], we developed several predictive parameters:

- **n-gram probability**: $C_1(t_j) = P(t_i|t_{i-1},..,t_{i-n+1})$
- **backward n-gram probability**:

$$C_2(t_j) = P(t_i|t_{i+1},..,t_{i+n-1})$$

- average **intra-language mutual information (MI)**:

$$C_3(t_j) = \frac{1}{J-1} \sum_{1 \leq i \neq j \leq J} MI(t_i, t_j)$$

- average **inter-languages mutual information**:

$$C_4(t_j) = \frac{1}{J} \sum_{1 \leq i \leq I} MI(s_i, t_j)$$

In addition a binary parameter $C_5(t_j) \in \{0,1\}$ is used to indicate whether a word is a function word (“the”, “to”, etc.) or not. N-gram based CMs measure the grammatical soundness and overall “well-formedness” of the sentence, while MI based ones measure the lexical consistency and translation accuracy.

LibSVM [7] is used for combining these measures. Instead of SVM as binary classifiers they are trained to produce a probability of correctness (the probability of belonging in the “positive” class). By doing so the acceptance threshold can be optimised. This gives some more flexibility to the classifier. The chosen kernel is a Radial Basis Function since it is simple and has been reported in [16] to give good results (near optimal):

$$K_\gamma(\mathbf{x},\mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}$$

| | F-measure | CAR | CRR |
|---|---|---|---|
| $C_1$ n-grams | 0.709 | 0.791 | 0.642 |
| $C_2$ backward n-grams | 0.653 | 0.652 | 0.654 |
| $C_3$ intra-language MI | 0.596 | 0.542 | 0.662 |
| $C_4$ inter-language MI | 0.597 | 0.516 | 0.708 |
| combination | 0.842 | 1.00 | 0.727 |

Table 2: Performances of word-level confidence measures

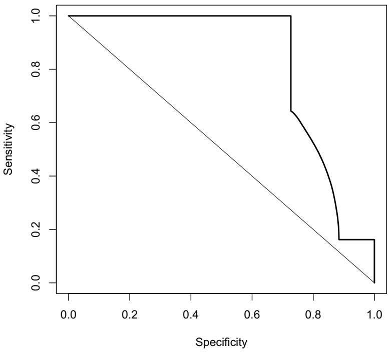

Figure 1: Roc curve of the classifier combining all word-level predictive parameters

In addition to γ, SVM also require that error cost $c$ be optimised. γ and $c$ were optimised by grid search with regard to the best F-measure (section 4) on a development corpus: ten thousand words are used to train the SVM given a couple $(\gamma, c)$, then the resulting model is evaluated on another set of ten thousand development words. When the grid search is finished, the model is trained with the best parameters found, and it is tested on a separate test corpus.

Table 2 shows the performances of the word-level confidence measures in terms of CAR, CRR and F-measure. It can be noticed that MI-based confidence measures perform poorly (especially intra-language MI), much worse than the results reported in [1, 2]. In this previous work CMs were evaluated on a natural corpus, which probably contained less grammatical and position errors than the automatically generated one. MI-based confidence measures are not good at detecting these types of errors, because they do not take word order into account, and are more focused on long range lexical relationships. However we show that combining all different features $C_1$ to $C_5$ permits to achieve highly accurate classification: the SVM combination outperforms the best confidence measure (n-grams) by 14% (absolute). This shows that the different CMs are complementary and that SVM combination is efficient. The combination correctly detects all correct words, and a reasonable share of incorrect words.

Figure 1 displays the ROC curve of this combination: the specific shape of the curve may be explained by the fact that input samples are well separated in the features space and that all correct words obtain high scores.

|  | MSE |
|---|---|
| combining ngrams and backward n-grams | 0.0433 |
| combining intra- and inter-language MI | 0.0835 |
| combining all features | 0.0429 |

Table 3: Performance of sentence-level confidence measures in terms of mean square error

## 6. Sentence-level Confidence Estimation

Sentence's reliability is estimated from the confidence scores of its words. We extend $C_1, C_2, C_3, C_4$ on sentences in the following fashion:

- $C_1(\mathbf{t}_1^J) = \sqrt[J]{\prod_{j=1}^{J} C_1(t_j)} \quad C_2(\mathbf{t}_1^J) = \sqrt[J]{\prod_{j=1}^{J} C_2(t_j)}$
- $C_3(\mathbf{t}_1^J) = \frac{1}{J}\sum_{j=1}^{J} C_3(t_j) \quad C_4(\mathbf{t}_1^J) = \frac{1}{J}\sum_{j=1}^{J} C_4(t_j)$

$C_1$ and $C_2$ are perplexities, $C_3$ and $C_4$ are average mutual information. Each of these scores is a component of the X features vector (section 4). During training SVM are used to perform regression of $\mathbf{X}(\mathbf{t},\mathbf{s})$ against the BLEU score $BLEU(\mathbf{t},\mathbf{s})$. During testing, SVM generate numerical scores $score(\mathbf{X}(\mathbf{t},\mathbf{s}))$ which are supposed to be close to "real" BLEU scores. During development stage the parameters $\gamma$, $c$ and accepted regression error are optimised by grid search with regard to mean square error on the development corpus. The performance is computed as the MSE on the test corpus (section 4).

Table 3 shows the performance of different combinations of sentence-level predictive parameters. The MSE obtained by a dummy measure giving random scores to sentences is around 0.20. This suggests that the combination of forward and backward perplexities is well correlated BLEU, which itself, although being suboptimal at sentence level, gives a good indication of quality. On the other hand MI based predictive parameters do not seem to significantly help in the combination.

## 7. Discussion and Conclusion

In this paper we presented a twofold contribution to machine translation: first we used SVM to combine original and classical word-level predictive parameters and showed that the combination outperforms the best single predictive parameters by 0.14 in F-measure; we also show that using support vector regression to mimic the BLEU score of a candidate translation without using a reference translation gives promising results; second, we present a method to automatically build a corpus with errors. This corpus is very useful for training and testing the confidence measures. Such tasks need large corpora indeed, which should ideally be actual machine translation systems output evaluated by a professional translator. But this is time and money consuming. Our generation method is flexible allow for realistic errors to be introduced in reference translations. Our goal now is to use sentence-level confidence measures to automatically detect good translation produced by a statistical machine translation system and feed them back in the training corpus, thus hopefully providing additional good translation training samples to the system.

## 8. References

[1] S. Raybaud, C. Lavecchia, D. Langlois, and K. Smaïli, "New confidence measures for statistical machine translation," in *Proceedings of the International Conference on Agents and Artificial Intelligence*, 2009, pp. 61–68.

[2] ——, "Word- and sentence-level confidence measures for machine translation," in *To be published in Proceedings of the European Association for Machine Translation Conference*, 2009.

[3] N. Ueffing and H. Ney, "Word-level confidence estimation for machine translation using phrase-based translation models," *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 763–770, 2005.

[4] ——, "Bayes decision rule and confidence measures for statistical machine translation." Springer, 2004, pp. 70–81.

[5] G. Guo, C. Huang, H. Jiang, and R. Wang, "A comparative study on various confidence measures in large vocabulary speech recognition," *2004 International Symposium on Chinese Spoken Language Processing*, pp. 9–12, 2004.

[6] J. Duchateau, K. Demuynck, and P. Wambacq, "Confidence scoring based on backward language models," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002.*, vol. 1, 2002.

[7] C. Chang and C. Lin, "Libsvm: a library for support vector machines," 2001.

[8] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation."

[9] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," *MT Summit*, vol. 5, 2005.

[10] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "The mathematic of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1994.

[11] F. Och and H. Ney, "Giza++: Training of statistical translation models," *available at http://www.fjoch.com/GIZA++.html*, 2000.

[12] A. Stolcke, "Srilm – an extensible language modeling toolkit," pp. 901–904, 2002.

[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, "Moses: Open source toolkit for statistical machine translation," *Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session*, 2007.

[14] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, and N. Ueffing, "Confidence estimation for machine translation. final report, jhu/clsp summer workshop," 2003.

[15] G. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[16] R. Zhang and A. Rudnicky, "Word level confidence annotation using combinations of features," in *Seventh European Conference on Speech Communication and Technology*, 2001, pp. 2105–2108.