Word- and Sentence-level Confidence Measures for Machine Translation

Sylvain Raybaud, Caroline Lavecchia, David Langlois, Kamel Smaïli
PAROLE team, LORIA
Campus Scientifique BP 239
54506 Vandoeuvre-lès-Nancy FRANCE
{sylvain.raybaud,caroline.lavecchia,david.langlois,kamel.smaili}@loria.fr

Abstract

A machine translated sentence is seldom completely correct. Confidence measures are designed to detect incorrect words, phrases or sentences, or to provide an estimation of the probability of correctness. In this article we describe several word- and sentence-level confidence measures relying on different features: mutual information between words, n-gram and backward n-gram language models, and linguistic features. We also try different combination of these measures. Their accuracy is evaluated on a classification task. We achieve 17% error-rate (0.84 f-measure) on word-level and 31% error-rate (0.71 f-measure) on sentence-level.

1 Introduction

Statistical techniques have been widely used and remarkably successful in automatic speech recognition, machine translation and in natural language processing over the last two decades. This success is due to the fact that this approach is language independent and requires no prior knowledge, only large enough text corpora to estimate probability densities on. However statistical methods suffer from an intrinsic drawback: they only produce the result which is most likely 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: this can be handled by the different *confidence measures (CMs)* which have been proposed for machine translation.

This paper extends and improve the work presented in (Raybaud et al., 2009): we introduce new CMs to assess the reliability of translation results. The proposed CMs take advantage of the constituents of a translated sentence: n-grams, word triggers, and also word features. We also combine the scores given by the different measures in order to produce a new one, hopefully more powerful, and the scores given to the different words in order to estimate the whole sentence's reliability.

1.1 A brief overview of statistical machine translation

In this framework the translation process is essentially the search for the most probable sentence in the target language given a sentence in the source language; let $\mathbf{s} = s_1, .., s_I$ be the source sentence (to be translated) and $\hat{\mathbf{t}} = t_1, .., t_J$ be the sentence generated in the target language by the system:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{t}|\mathbf{s}) \quad (1)$$

which is equivalent (using the Bayes rule) to:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} P(\mathbf{t})P(\mathbf{s}|\mathbf{t}) \quad (2)$$

In Equation 2, $P(\mathbf{t})$ is estimated from a *language model* and is supposed to estimate the correctness of the sentence ("is it a good sentence in the target language ?"), and $P(\mathbf{s}|\mathbf{t})$ is computed from a *translation model* and is supposed to reflect the accuracy of the translation ("does the generated sentence carry exactly the same information than the source sentence ?"). The language model is itself estimated on a large text corpus written in the target language, while the translation model is computed on a bilingual aligned corpus (a text and its translation with line-wise correspondence).

The decoder then generates the best hypothesis by making a compromise between these two probabilities.
Of course there are three main drawbacks to this approach: first the search space is so huge that exact computation of the optimum is intractable; second, even if it was, statistical models have inherent limitations which prevent them from being completely sound linguistically; finally, the probability distribution P can only be estimated on finite corpora, and therefore suffers from imprecision and data sparsity. Because of that, any SMT system sometimes produces erroneous translations. It is an important task to detect and possibly correct these mistakes, and this could be handled by confidence measures.

2 An Introduction to Confidence Measures

2.1 Motivation and principle of confidence estimation

As said before, SMT systems make mistakes. A word's translation can be wrong, misplaced, or missing. Extra words can be inserted. A whole sentence can be wrong or only parts of it. In order to improve the overall quality of the system, it is important to detect these errors by assigning a so called confidence measure to each translated word, phrase or sentence. Ideally this measure would be the probability of correctness. An ideal word-level estimator would therefore be the probability that a given word appearing at a given position in a given sentence is correct; using the notations of Section 1.1 (t_j being the j-th word of sentence $\mathbf{t}$), this is expressed by the following formula:

$$\text{word confidence} = P(correct|j,t_j,\mathbf{s}) \quad (3)$$

and an ideal sentence-level estimator would be:

$$\text{sentence confidence} = P(correct|\mathbf{t},\mathbf{s}) \quad (4)$$

However these probabilities are difficult to estimate accurately; this is why existing approaches rely on approximating them or on computing scores which are supposed to monotonically depend on them.

2.2 State of the art

Confidence estimation is a common problem in artificial intelligence and information extraction in general (Culotta and McCallum, 2004; Gandrabur et al., 2006). When it comes to natural language processing, it has been intensively studied for automatic speech recognition (Mauclair, 2006; Razik, 2007; Guo et al., 2004). We find in literature (Blatz et al., 2003; Ueffing and Ney, 2004; Ueffing and Ney, 2005; Uhrik and Ward, 1997; Duchateau et al., 2002) different ways of approximating the probability of correctness or of calculating scores which are supposed to reflect this probability.

There exist three dominating approaches to estimation of word- and sentence-level confidence measures for machine translation:

- Estimate posterior probabilities (for example using a word-lattice or a translation table).
- Compute a predictive parameter (numerical score, for example a likelihood ratio) supposed to depend monotonically on the correctness probability.
- Combine predictive parameters through machine learning techniques in order to estimate the probability of correctness.

Many different confidence measures are investigated in (Blatz et al., 2003). They are based on source and target language models features, n-best lists, words-lattices, translation tables, and so on. The authors also present efficient ways of classifying words or sentences as "correct" or "incorrect" by using naïve Bayes, single- or multi-layer perceptron.

2.3 Our approach to confidence estimation

In the following we will first present three original word-level predictive parameters, based on:

- Intra-language mutual information (intra-MI) between words in the generated sentence.
- Inter-language mutual information (inter-MI) between source and target words.
- A target language model based on linguistic features.

We also implement two classical predictive parameters and combine them with our estimators:

- An n-gram model of the target language.
- A backward n-gram language model (Duchateau et al., 2002).

Mutual Information has been proved suitable for building translation tables (Lavecchia et al., 2007). We use intra-language MI to estimate the relevance of a word in the candidate translation given its context (it is supposed to reflect the lexical consistency). Inter-language MI based confidence estimation gives an indication of the relevance of a translation by checking that each word in the hypothesis can indeed be the translation of a word in the source sentence. N-gram, backward n-gram and linguistic features models estimate the lexical and grammatical correctness of the hypothesis. These different measures are then combined, either linearly with weights optimised with regard to error rate, or through logistic regression (Section 6). Each of these estimators produces a score for every word. This score is then compared to a threshold and the word is labelled as "correct" if its score is greater, or "incorrect" otherwise. This classification is then compared to a man made reference which gives an estimation of the efficiency of the measures, in terms of error rate, ROC curve and F-measure (Section 2.3.1). Finally we combine the word-level scores in order to compute sentence-level confidence measures. Each sentence is then classified as correct or incorrect by comparing its score to a threshold, and this decision is compared to a man-made decision in order to estimate the accuracy of the measure.

2.3.1 Evaluation of the confidence measures

As explained before, the CMs are evaluated on a classification task. We split the test corpus of our machine translation system into a development corpus (300 pairs of sentences) and a test corpus (200 sentences) for our confidence measures. We manually classified as correct or incorrect the words and sentences from these 500 French translation generated by Pharaoh (Koehn, 2004). Human were given few constraints; the first and most important one was "the first impression is the best"; the second one was "if a word makes no sense in the sentence or is really misplaced then it is wrong"; the third one was "a translation that does not contain essential information stated in the source sentence is wrong"; the last and most important one was "the first impression is the best".
We then ran our classifiers on the same sentences. A word was classified as correct if its score was above a given threshold. The results were then compared to the human-made references. We used the following metrics to estimate how well our classifier behaved; "item" refers either to "word" or "sentence":

Classification Error Rate (CER) is the proportion of errors in classification:

$$\frac{\text{number of incorrectly classified items}}{\text{total number of items}}$$

Correct Acceptance Rate (CAR or Sensitivity) is the proportion of correct items retrieved:

$$\frac{\text{number of correctly accepted items}}{\text{total number of correct items}}$$

Correct Rejection Rate (CRR or Specificity) is the proportion of incorrect items retrieved:

$$\frac{\text{number of correctly rejected items}}{\text{total number of incorrect items}}$$

F-measure is the harmonic mean of CAR and CRR:

$$F = \frac{2 \times CAR \times CRR}{CAR + CRR}$$

These metrics are fairly 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).
As the acceptance threshold increases, CAR decreases and CRR increases. The plot of *CRR vs. CAR* is called the ***ROC curve*** (*Receiver Operating Characteristic*). 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. We also plotted on the same diagrams F-measure and CER against CAR.

3 Software and Material Description

Experiments were run using an English to French phrase-based translation system. We trained a system corresponding to the baseline described in the *ACL workshop on statistical machine translation* (Koehn, 2005). It uses an IBM-5 model (Brown et al., 1994) and has been trained on the EUROPARL corpus (proceedings of the European Parliament, (Koehn, 2005)) using GIZA++ (Och and Ney,

2000) and the SRILM toolkit (Stolcke, 2002). The decoding process is handled by Pharaoh. The French vocabulary was composed of 63,508 words and the English one of 48,441 words. We summarise in Table 1 the sizes of the different parts of the corpus. This system achieves state of the art performances.

	sentences pairs	running words	
set		English	French
Learning	465,750	9,411,835	10,211,388
Development	3000	75,964	82,820
Test	500	4,945	4,899

Table 1: Corpora sizes

Human annotators reported 16.5% erroneous words and 32.6% erroneous sentences, according to the previously stated criteria.

4 Mutual Information based Confidence Measures

4.1 Mutual information in language modelling

In probability theory mutual information measures how mutually dependent are two random variables. It can be used to detect pairs of words which tend to appear together in sentences. Guo proposes in (Guo et al., 2004) a word-level confidence estimation for speech recognition based on mutual information. In this paper we will compute inter-word mutual information following the approach in (Lavecchia et al., 2007), which has been proved suitable for generating translation tables, rather than Guo's.

$$\begin{aligned} MI(x,y) &= p(x,y)log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (5)\\ p(x,y) &= \frac{N(x,y)}{N}\\ p(x) &= \frac{N(x)}{N} \end{aligned}$$

where N is the total number of sentences, $N(x)$ is the number of sentences in which x appears and $N(x,y)$ is the number of sentences in which x and y co-occur. We smooth the estimated probability distribution, as in Guo's paper, in order to avoid null probabilities:

$$\begin{aligned} N(x,y) &\leftarrow N(x,y)+C \quad (6)\\ p(x,y) &\leftarrow \frac{p(x,y)+\alpha p(x)p(y)}{1+\alpha} \quad (7) \end{aligned}$$

in which C is a non-negative integer and α a non-negative real number. For example, words like *"ask"* and *"question"* have a high mutual information, while words coming from distinct lexical fields (like *"poetry"* and *"economic"*) would have a very low one. Since it is not possible to store a full matrix in memory, only the most dependent word pairs are kept: we obtain a so called *triggers list*.

4.2 Confidence measure based on intra-language mutual information

By estimating which target words are likely to appear together in the same sentence, intra-language MI based confidence score is supposed to reflect the lexical consistency of the generated sentence. The source sentence is not taken into account. We computed mutual information between French words from the French part of the bilingual corpus. Table 2 shows an example of French intra-lingual triggers, sorted by decreasing mutual information.

word	→	*triggered word*
sécurité	→	*alimentaire*
sécurité	→	*étrangère*
sécurité	→	*politique*
	...	
politique	→	*commune*
politique	→	*économique*
politique	→	*étrangère*

Table 2: An example of French intra-lingual triggers

Let $\mathbf{t} = t_1..t_J$ be the generated sentence. The score assigned to t_j is the weighted average mutual information between t_j and the words in its context:

$$C(t_j) = \frac{\sum_{i=1..J, i\neq j} w(|j-i|)MI(t_i,t_j)}{\sum_{i=1..J, i\neq J} w(|j-i|)} \quad (8)$$

where $w()$ is a scaling function lowering the importance of long range dependencies. It can be constant if we do not want to take words' positions into account, exponentially decreasing if we want to give more importance to pairs of words close to each other, or a shifted Heaviside function if we want to allow triggering only within a given range (which we will refer to as *triggering window*). Function words (like *"the"*, *"of"*,...) generally have a very high mutual information with all other words thus polluting the trigger list; therefore they are not taken into account for computing mutual

information.

Presenting the performances of the confidence measure with all different settings (different triggering windows, size of trigger list,...) would be tedious. Therefore we only show the settings that yield the best performances. Note that while other settings often yield much worse performance, a few perform almost as well, therefore there are no definite "optimal settings". Figure 1 shows the ROC curve, CER and F-measure of a classifier based on intra-MI in which function words were ignored.

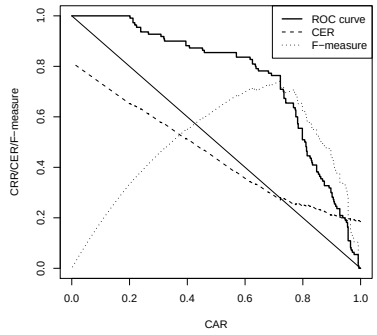

Figure 1: Intra-MI, no function words, no weighting nor triggering window.

Taking word positions into account yields lower performance: intra-language MI indeed reflects lexical consistency of the sentence, but two related words may not be next to each other in the sentence.

4.3 Confidence measure based on inter-language mutual information

The principle of intra-language MI was to detect which words trigger the appearance of another word in the same sentence. This principle can be extended to pairs of source and target sentences (Lavecchia et al., 2007): let $N_S(x)$ be the number of source sentences in which x appears, $N_T(y)$ the number of target sentences in which y appears, $N(x,y)$ the number of pairs *(source sentence, target sentence)* such that x appears in the source and y in the target, and N the total number of pairs of source and target sentences. Then let us define:

$$\begin{aligned} p_S(x) &= \frac{N_S(x)}{N} \\ p_T(y) &= \frac{N_T(y)}{N} \\ p(x,y) &= \frac{N(x,y)}{N} \\ MI(x,y) &= p(x,y)log_2\left(\frac{p(x,y)}{p_S(x)p_T(y)}\right) \end{aligned} \quad (9)$$

Guo's smoothing can be applied as in Section 4.2. One then keeps only the best triggers and obtain a so-called *inter-lingual triggers list*. Table 3 shows an example of such triggers between English and French words, sorted by decreasing mutual information.

English word	→	*triggered French word*
security	→	*sécurité*
security	→	*étrangère*
security	→	*politique*
	...	
policy	→	*politique*
policy	→	*commune*
policy	→	*étrangère*

Table 3: An Example of Inter-Lingual triggers

The confidence measure is then:

$$C(t_j) = \frac{\sum_{i=1}^{I} w(|j-i|)MI(s_i,t_j)}{\sum_{i=1}^{I} w(|j-i|)} \quad (10)$$

We show in Figure 2 the characteristics of such an inter-MI based classifiers. This time triggering was allowed within a window of width 9 centred on the word the confidence of which was being evaluated. Function words were excluded.

Unlike intra-MI based classifier, we found here that setting a triggering window yields the best performance. This is because inter-language MI indicates which target words are possible translations of a source word. This is much stronger than the lexical relationship indicated by intra-MI; therefore allowing triggering only within a given window or simply giving less weight to "distant" words pairs reflects the fact that words in the source sentence and their translations in the target sentence appear more or less in the same order (this is the same as limiting the distortion, which is the difference between the positions of a word and its translation).

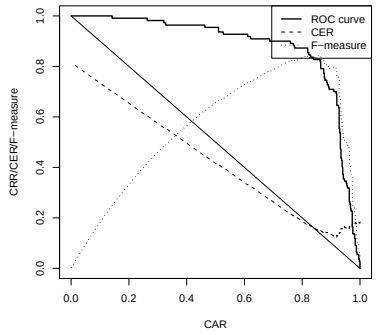

Figure 2: Inter-language MI based CM: function words excluded, no normalisation, triggering is allowed within a centred window of width 9.

5 Language-Model based Confidence Measures

We now present confidence measures based on different n-gram-like target-language models. We assume that if a sentence "looks wrong" in the target language then it is unlikely to be an accurate translation. We will not present their word-level performance, which are somewhat poor, however we will see in Section 7 that they are efficient for detecting incorrect sentences.

5.1 N-grams based confidence measure

Remember Equation 2: the decoder makes a compromise between $P(\mathbf{t})$ (which we will refer to as *language model score*) and $P(\mathbf{s}|\mathbf{t})$ (*translation score*). Because of that, if a candidate $\mathbf{t}$ has a high translation score and a low language model score, it might be accepted as the "best" translation. But a low LM score often means an incorrect sentence and therefore a bad translation. This consideration applies on sub-sentence level as well as on sentence level: if the n-gram probability of a word is low, it often means that it is wrong or at least misplaced. Therefore we want to use the language model alone in order to detect incorrect words. We decided to use the word probability derived from an n-gram model as a confidence measure:

$$C(t_j) \quad = \quad P(t_j|t_{j-1},..,t_{j-n+1}) \qquad (11)$$

While intra-language triggers are designed to estimate the lexical consistency of the sentence, this measure is supposed to estimate its well-formedness. We empirically found that 4-grams were best suited.

5.2 Backward n-gram language model

Because classical n-gram models only take into account the left context of a word, it is natural to extend the idea to consider the *right-context* (Duchateau et al., 2002). This should be efficient to detect, for example, incorrect determinants and other function words. A backward n-gram language model is simply trained on a corpus in which sentences have been "reverted": "*Hello world !*" becomes "*! world Hello*". We then use as a confidence measure:

$$C(t_j) \quad = \quad P(t_j|t_{j+1},..,t_{j+n-1}) \qquad (12)$$

We found that bigrams achieved the best performances, which backs our idea that this language model is useful for detecting wrong function words.

5.3 Linguistic features based confidence measure

We designed a confidence measure to specifically target grammatical errors. using BDLEX (De Calmès and Pérennou, 1998), each word $t$in the corpora was replaced by a vector $\tilde{t}$ of it's *syntactic class*, *tense* if relevant, and *number and gender* or *person*. We then built n-gram models on the modified training corpus, and used $P(\tilde{t}_j|\tilde{t}_{j-1},..,\tilde{t}_{j-n+1})$ as a confidence score. The performance were poor both at word- and sentence-level, therefore this measure won't be used in the rest of the paper. More information can be found in (Raybaud et al., 2009).

6 Fusion of Confidence Measures

We linearly combined the scores assigned to each word by different confidence measures to produce a new score. The weights are optimised with respect to error-rate on our development corpus. This method yields no significant improvement on the best measure used alone (inter-language mutual information 4.3). Therefore we used a more sophisticated logistic regression instead.

6.1 Logistic regression

An other option is to use logistic regression to estimate a probability of correctness given a vector of predictive parameters. If $X \in \mathbb{R}^k$ is a vector of predictive parameters, the idea of logistic regression is to find coefficients $\Theta \in \mathbb{R}^k, b \in \mathbb{R}$ such that:

$$P(correct|X) = \frac{1}{1+e^{\langle\Theta,X\rangle+b}} \qquad (13)$$

These coefficients are optimised with respect to the maximum likelihood criterion. Here again we could not improve word-level performances compared to inter-language mutual information; the latter is way better than any other measure we implemented, thus being difficult to improve on. however we will see that this performed well on sentence-level.

7 Sentence-level Confidence Estimation

We chose to estimate a sentence's reliability from the confidence score of its words. We empirically found that the best method was to combine LM and backward LM confidence measures through logistic regression, and then set the sentence's score as the normalised product of the correctness probabilities of words; let $X(t) \in \mathbb{R}^2$ be a vector whose components are LM and backward-LM probabilities of word t; let $\Theta \in \mathbb{R}^2$ and $b \in \mathbb{R}$ be the optimal logistic regression coefficients; then the score of sentence $\mathbf{t} = t_1, .., t_J$ is given by:

$$P(correct|j,\mathbf{t}) = \frac{1}{1+exp\big(\langle\Theta, X(t_j)\rangle + b\big)} \quad (14)$$

$$C(\mathbf{t}) = \sqrt[J]{\prod_{j=1}^{J} P(correct|j,\mathbf{t})} \quad (15)$$

Figure 3 shows the ROC curve, f-measure and error rate curves of a sentence classifier relying on the above combination of measures. The best f-measure is 0.71, corresponding to a 30.6% error rate.

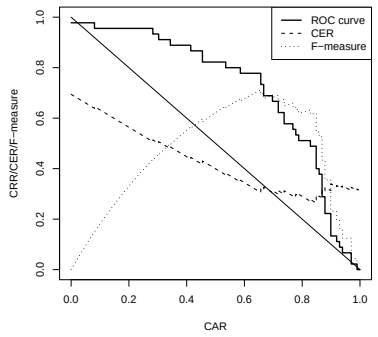

Figure 3: Detection of incorrect sentences based on n-gram and backward-n-gram language models.

8 Discussion and Conclusion

In this article, we present confidence scores that showed interesting discriminating power. We summarised the results obtained by the best different word-level estimators (in terms of F-measure) in Table 4. For comparison Blatz et al. obtain in (Blatz et al., 2003) a CER of 29.2% by combining two different word posterior probability estimates (with and without alignment) and the translation probabilities from IBM-1 model. The result obtained with sentence classifiers are presented in Table 5. For comparison Blatz et al. obtained an error rate around 28%.

	CER	CAR	CRR	F-measure
intra-MI	0.270	0.722	0.764	0.742
inter-MI	0.171	0.819	0.873	0.845

Table 4: Performances of the best word-classifiers.

	CER	CAR	CRR	F-measure
LM and LM-backward	0.306	0.657	0.778	0.712

Table 5: Performances of the best sentence-classifier.

It is interesting to remark that the confidence measures which perform well at sentence level are those who perform poorly at word level. It might be because sometimes while you can tell for sure that a sentence is wrong, it is difficult to pinpoint an erroneous word. Also an important cause of sentence incorrectness is wrong word order, about which MI based confidence measures are lenient, while LM based ones are not.

8.1 Application of Confidence Measures

Beside manual correction of erroneous words we can imagine several applications of confidence estimation: **pruning or re-ranking of the n-best list**, **generation of new hypothesis** by recombining parts of different candidates having high scores, or **discriminative training** by tuning the parameters to optimise the separation between sentences (or words, or phrases) having a high confidence score (hopefully they are correct translations) and sentences having a low one.

8.2 Prospects

We plan to go further in our investigation on confidence measures for SMT: first the measures we used do not directly take into account word deletion nor word order, neither do our reference corpus (missing words are not indicated). This serious drawback has to be addressed. Also many features used in speech recognition or automatic translation could be used for confidence estimation: distant models, word alignment, word spotting, etc...

We also plan to investigate SVM and neural network for combining predictive parameters (Zhang and Rudnicky, 2001). Finally we have to work on the corpora themselves: man-made classification is slow, tedious, and the results depend heavily on the operator. We will investigate semi-automatic creation of labelled training, development and test data for confidence measures.

References

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

Brown, P.F., S.D. Pietra, V.J.D. Pietra, and R.L. Mercer. 1994. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Culotta, A. and A. McCallum. 2004. Confidence estimation for information extraction. *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

De Calmès, M. and G. Pérennou. 1998. Bdlex: a lexicon for spoken and written french. In *Proceedings of 1st International Conference on Langage Resources & Evaluation*.

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

Gandrabur, S., G. Foster, and G. Lapalme. 2006. Confidence estimation for nlp applications. *ACM Transactions on Speech and Language Processing*, 3(3):1–29.

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

Koehn, P. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, pages 115–124.

Koehn, P. 2005. Europarl: A parallel corpus for statistical machine translation. *MT Summit*, 5.

Lavecchia, C., K. Smaïli, D. Langlois, and J.P. Haton. 2007. Using inter-lingual triggers for machine translation. *Eighth conference INTERSPEECH*, pages 2829–2832.

Mauclair, J. 2006. *Mesures de confiance en traitement automatique de la parole et applications*. Ph.D. thesis, LIUM, Le Mans, France.

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

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

Razik, Joseph. 2007. *Mesures de Confiance trame-synchrones et locales en reconnaissance automatique de la parole*. Ph.D. thesis, LORIA, Nancy, FRANCE.

Stolcke, A. 2002. Srilm – an extensible language modeling toolkit. pages 901–904.

Ueffing, N. and H. Ney. 2004. Bayes decision rule and confidence measures for statistical machine translation. pages 70–81. Springer.

Ueffing, N. and H. Ney. 2005. 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*, pages 763–770.

Uhrik, C. and W. Ward. 1997. Confidence metrics based on n-gram language model backoff behaviors. In *Fifth European Conference on Speech Communication and Technology*, pages 2771–2774.

Zhang, R. and A.I. Rudnicky. 2001. Word level confidence annotation using combinations of features. In *Seventh European Conference on Speech Communication and Technology*, pages 2105–2108.